57º SBPO
SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
Gramado • RS
DE 5 A 9 DE OUTUBRO DE 2025

Uma Abordagem Heurística para a Precificação em Mercados Competitivos com Oferta Irrestrita

Heloisa Reis Leal
Universidade Federal Fluminense e Petrobras
Av. Gal. Milton Tavares de Souza, s/n, Praia Vermelha Boa Viagem Niterói - RJ
heloisareis@ic.uff.br

Yuri Abitbol de Menezes Frota
Universidade Federal Fluminense
Av. Gal. Milton Tavares de Souza, s/n, Praia Vermelha Boa Viagem Niterói - RJ
yuri@ic.uff.br

David Sotelo Pinheiro da Silva
Petrobras
R. General Canabarro, 500, Rio de Janeiro
david@petrobras.com.br

RESUMO

A precificação é uma das mais importantes atividades associadas à gestão financeira de empresas atuando em ambientes de livre concorrência de preços. Para a maioria delas, melhorias na política de definição de preços resultam em uma forma rápida e eficiente de aumento em seus lucros auferidos. Neste contexto, o problema de Precificação em Mercados Competitivos com Oferta Irrestrita (PMCOI) consiste no ajuste simultâneo de preços em diversos pontos de venda pertencentes a um mesmo agente comercializador com o objetivo de maximizar seu lucro. O presente trabalho propõe um método ILS (*Iterated Local Search*) para solucionar o PMCOI. Apresentamos no trabalho a descrição deste algoritmo, os resultados dos testes computacionais, e a sua comparação com um método exato.

PALAVRAS CHAVE. Precificação, Otimização Combinatória, Metaheurística.

ABSTRACT

Pricing represents a critical component of financial management for firms operating in environments characterized by free price competition. For many such firms, enhancements in pricing strategies offer a rapid and effective means to increase profitability. Within this context, the Pricing in Competitive Markets with Unrestricted Supply (PCMUS) problem involves the simultaneous adjustment of prices across multiple sales locations managed by a single commercial agent, with the objective of maximizing overall profit in a competitive market setting. This study proposes an Iterated Local Search (ILS) algorithm to address the PCMUS problem. We present the algorithm in detail, report the results of computational experiments, and compare its performance with that of an exact solution method.

KEYWORDS. Pricing, Combinatorial Optimization, Metaheuristic.

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

1. Introdução

O problema Precificação em Mercados Competitivos com Oferta Irrestrita (PMCOI) [Leal et al., 2023] é um problema de precificação de um produto em cada um dos pontos de venda pertencentes a uma mesma empresa (denominada *companhia*), a qual encontra-se inserida em um ambiente de livre competição. Este ambiente competitivo é composto por outras empresas (denominadas *concorrentes*), as quais, por sua vez, também possuem pontos de venda do produto em questão, para os quais considera-se que os preços se encontram pré-estabelecidos. O problema trata, portanto, do ajuste de preços nos pontos de venda pertencentes à companhia, uma vez considerados os preços dos pontos de venda concorrentes. Ressalta-se ainda que o preço p_s em um ponto de venda s da companhia deve pertencer a um conjunto finito discreto $\{p_s \in \mathbb{N} \mid \underline{p_s} \leq p_s \leq \overline{p_s}\}$, onde $\underline{p_s}$ e $\overline{p_s}$ são limites inferiores e superiores para o preço de p_s, respectivamente, fornecidos na instância do problema. No ambiente competitivo considerado, enquanto as empresas são responsáveis pela venda do produto em questão, existe um mercado consumidor, representado por um conjunto de consumidores, os quais possuem interesse em adquirir o produto comercializado ao menor custo unitário de aquisição. Mais especificamente, assume-se que cada consumidor possui uma demanda pré-estabelecida (inelástica à precificação) a qual será totalmente atendida pelo ponto de venda que lhe oferecer o menor custo unitário de aquisição. Em particular, o custo unitário de aquisição de um produto ofertado em um ponto de venda, sob a ótica de um consumidor, é definido como a soma entre o preço do produto no ponto de venda ofertante e o custo logístico unitário necessário para o atendimento do consumidor por parte do ponto de venda, sendo o custo logístico unitário pré-fornecido na instância do problema. Considera-se que a demanda de um consumidor é totalmente atendida pelo ponto de venda que lhe oferecer o menor custo unitário de aquisição pois assume-se, por simplificação, que as ofertas dos pontos de venda são ilimitadas. A função objetivo do problema analisado é a maximização do lucro total da companhia, o qual depende das demandas de cada um dos consumidores que decidem ser atendidos por algum dos pontos de venda da companhia, do preço estabelecido e do custo operacional associado em cada um de seus pontos de venda. Em particular, consideraremos que o custo operacional de um ponto de venda é uma função constante por partes, fornecida na instância do problema, a qual dependerá da quantidade total vendida pelo respectivo ponto de venda. O PMCOI é um problema de otimização combinatória, uma vez que cada preço a ser definido pertence a um conjunto finito e discreto.

Em Leal et al. [2023] temos a formalização do PMCOI e a apresentação da sua modelagem como problema de programação linear inteira mista. Também são apresentados a prova de que ele pertence à classe de problemas $\mathcal{NP}$-Difícil, a criação de instâncias de testes e os resultados dos experimentos computacionais para o método exato de solução do problema. Os resultados obtidos mostraram que soluções ótimas (ou próximas ao ótimo) foram obtidas para todas as instâncias testadas. Apesar de ser um resultado satisfatório, nosso objetivo maior é conseguir o melhor resultado possível para a instância representativa de abrangência nacional, na qual o conjunto de consumidores é composto por todos os municípios brasileiros. Nesta instância, a quantidade total demandada e vendida é muito alta, e, mesmo um aumento de um percentual pequeno neste resultado significa um grande aumento de lucro para a companhia em questão.

Propomos então no presente trabalho, uma abordagem heurística com o objetivo de obter resultados melhores para estas instâncias maiores. A metodologia proposta será a utilização de um método metaheurístico ILS (*Iterated Local Search*) para alcançar este objetivo. O artigo está dividido da seguinte forma: na Seção 2, fazemos uma breve revisão da literatura relacionada ao problema PMCOI e, na Seção 3, trazemos a sua formulação como problema de programação linear inteira mista. Em seguida, na Seção 4, a metaheurística ILS será descrita, e na Seção 5, analisaremos seus resultados. Por último, na Seção 7, apresentaremos a conclusão e os trabalhos futuros.

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

2. Trabalhos Relacionados

Nas últimas décadas, problemas de precificação têm sido amplamente modelados como problemas de otimização combinatória, sob diversas variações. Segundo Bui et al. [2025], tais problemas são denominados Problemas Combinatórios de Precificação (CPP). Dentro dessa classificação, destaca-se o Problema de Precificação em Redes (NPP), amplamente estudado nos setores de transporte e telecomunicações. Nesse problema, um operador define tarifas sobre um subconjunto de arestas sob sua responsabilidade, enquanto os clientes escolhem o caminho de menor custo que conecta dois vértices de interesse [van Hoesel, 2008]. Além disso, a classe CPP abrange variações do NPP nas quais a escolha do caminho de menor custo pelos clientes é substituída por outros problemas combinatórios, como o problema da árvore geradora mínima em Cardinal et al. [2009], o problema da mochila em Pferschy et al. [2021], o problema do conjunto estável em Böhnlein et al. [2023], e o problema da cobertura de vértices bipartidos em Briest et al. [2012].

Ressalta-se também o trabalho de [Ahmadi-Javid et al., 2018], cujo objetivo é maximizar o lucro através de precificação e determinação da localização de novos pontos de venda; trata-se de um problema característico de vários ramos da indústria, como serviços postais, sistemas de distribuição de frutas, couro, dentre outros produtos [Lin e Tian, 2023].

Até onde nosso conhecimento permite afirmar, a variação mais próxima ao problema PMCOI encontra-se em [Plyasunov e Panin, 2013]. Nele, os autores estudam a precificação com o objetivo de maximizar o lucro de um produtor e minimizar o custo de consumidores, mas com algumas características diferentes em relação ao problema apresentado aqui. Em particular: os consumidores possuem um orçamento máximo definido, mas não têm demanda; não é considerado o custo operacional dos pontos de venda e a companhia não tem concorrentes ou faixas de preços para limitar a definição de preços nos pontos de venda.

3. Precificação em Mercados Competitivos com Oferta Irrestrita

O problema PMCOI foi formulado em [Leal et al., 2023] como um problema de programação linear inteira mista. As Tabelas 1 e 2 apresentam a notação usada na formulação e as variáveis do problema, respectivamente. Usando esta notação, a função objetivo do problema é definida em (1) como a maximização do lucro da companhia. A Restrição (2) define que cada consumidor será atendido por apenas um ponto de venda.

Conjuntos e Parâmetros	Descrição
S	Conjunto de pontos de venda da companhia.
S^c	Conjunto de pontos de venda dos concorrentes.
$\mathbb{S} = (S \cup S^c)$	Conjunto de todos os pontos de venda.
B	Conjunto de consumidores.
L_s	Conjunto de intervalos $\{0, \ldots, \lvert L_s \rvert - 1\}$ do total de unidades vendidas pelo ponto de venda $s \in S$.
$\overline{u}_{sl}, \underline{u}_{sl}$	Máximo ($\overline{u}_{sl}$) e mínimo ($\underline{u}_{sl}$) de número de unidades vendidas no intervalo $l \in L_s$ pelo ponto de venda $s \in S$.
$\underline{p}_s, \overline{p}_s$	Limites inferiores ($\underline{p}_s$) e superiores ($\overline{p}_s$) para o preço do ponto de venda $s \in S$.
p^c_s	Preço do concorrente $s \in S^c$.
d_b	Demanda do consumidor $b \in B$.
c^{op}_{sl}	Custo operacional unitário do ponto de venda $s \in S$, no intervalo $l \in L_s$.
c^{lo}_{sb}	Custo logístico unitário incorrido no consumidor $b \in B$ quando compra uma unidade no ponto de venda $s \in \mathbb{S}$.

Tabela 1: Conjuntos e Parâmetros

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

Variáveis	Descrição
p_s	Variável contínua que indica o preço do ponto de venda $s \in \mathbb{S}$.
m_s	Variável contínua que indica a margem de lucro unitária do ponto de venda $s \in S$.
x_{sb}	Variável contínua que indica o lucro obtido pelo ponto de venda $s \in S$ com o consumidor $b \in B$.
w_{sl}	Variável binária que indica que o total de unidades vendidas pelo ponto de venda $s \in S$ está no intervalo $l \in L_s$.
y_{sb}	Variável binária que indica que o ponto de venda $s \in \mathbb{S}$ atende à demanda total de $b \in B$.

Tabela 2: Variáveis

A margem de lucro unitária de um ponto de venda da companhia é definida na Restrição (3), sendo dada pelo preço no ponto de venda decrescido de seu custo operacional unitário. A Restrição (4) define os preços mínimos e máximos para cada ponto de venda da companhia, e a Restrição (5) define os preços dos concorrentes. O lucro obtido pelo ponto de venda em relação a cada consumidor atendido encontra-se definido nas Restrições (6 e 7). A Restrição (8) garante que cada consumidor irá comprar do ponto de venda mais barato (considerando preço e custo logístico), onde M é um valor grande o suficiente. Nesta Restrição, se $s \in S$ e $q \in S^c$, temos que $\alpha = 1$, caso contrário $\alpha = 0$. Isso garante que um consumidor só compra de um ponto de venda da companhia se o seu preço acrescido do respectivo custo logístico (custo de aquisição) for estritamente menor do que o preço do concorrente acrescido do respectivo custo logístico. Caso dois pontos de venda da companhia empatem em menores custos de aquisição, consideramos que o consumidor comprará do ponto de venda que der mais lucro. Esta abordagem é conhecida como abordagem otimista. Por fim, as Restrições (9) e (10) identificam o intervalo correto da quantidade de unidades vendida por cada ponto de venda. Segue-se, portanto, a formulação do PMCOI:

$$\max \sum_{s \in S} \sum_{b \in B} x_{sb} \tag{1}$$

s.t.

$$\sum_{s \in \mathbb{S}} y_{sb} = 1, \qquad \forall b \in B \tag{2}$$

$$p_s = m_s + \sum_{l \in L_s} c^{op}_{sl} w_{sl} \qquad \forall s \in S \tag{3}$$

$$\underline{p}_s \leq p_s \leq \overline{p}_s \qquad \forall s \in S \tag{4}$$

$$p_s = p^c_s, \qquad \forall s \in S^c \tag{5}$$

$$x_{sb} \leq (m_s)\, d_b \qquad \forall s \in S, \forall b \in B \tag{6}$$

$$x_{sb} \leq (y_{sb})\, d_b \overline{p}_s \qquad \forall s \in S, \forall b \in B \tag{7}$$

$$p_s + c^{lo}_{sb} - p_q - c^{lo}_{qb} + \alpha \leq (1 - y_{sb})M, \qquad \forall s, q \in \mathbb{S}, \text{ tal que } s \neq q, \; \forall b \in B \tag{8}$$

$$\sum_{l \in L_s} w_{sl} = 1 \qquad \forall s \in S \tag{9}$$

$$w_{sl} \underline{u}_{sl} \leq \sum_{b \in B} y_{sb} d_b \leq (1 - w_{sl})M + \overline{u}_{sl} \qquad \forall s \in S, \forall l \in L_s \tag{10}$$

4. Metaheurística ILS

Nesta seção propomos um algoritmo metaheurístico baseado no método *Iterated Local Search* (ILS)[Lourenço et al., 2019], previamente aplicada com sucesso em uma ampla variedade

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

Algorithm 1: ILS

```
   Data: it_MAX, level_INI, level_MAX, step
 1 P* ← (p̄_0, p̄_1, ..., p̄_|S|) e P' ← P*;
 2 it ← 0 e level ← level_INI;
 3 while it < it_MAX do
 4     P'' ← Local_Search(P');
 5     if Profit(P'') > Profit(P*) then
 6         P* ← P'';
 7         it ← 0;
 8         level ← level_INI;
 9     else
10         if level ≤ level_MAX then
11             level++;
12         else
13             it++;
14             level ← level_INI;
15     P' ← Shake(P*, level, step);
16 return P*
```

de problemas de otimização combinatória. A idéia central do ILS é combinar a estratégia de busca local, que melhora a solução de entrada fazendo-a convergir para um ótimo local, com o mecanismo de perturbação, que modifica esta solução, com o objetivo de explorar novos ótimos locais. No ILS aqui proposto, os métodos de busca local ($Local_Search$) e de perturbação ($Shake$) são executados iterativamente com o objetivo explorar o espaço de solução do problema.

O Algoritmo 1 apresenta o método Iterated Local Search (ILS). Inicialmente, são definidos os parâmetros de entrada: it_{MAX}, que representa o número máximo de iterações consecutivas sem melhoria na solução; e os parâmetros $level_{INI}$, $level_{MAX}$ e $step$, utilizados no procedimento de perturbação, que indicam, respectivamente, os valores mínimo e máximo da quantidade de pontos de venda cujos preços serão modificados, bem como o incremento dessa quantidade a cada alteração. Cada solução de uma instância do problema PMCOI pode ser representada pelo conjunto de preços atribuídos a todos os pontos de venda da companhia. O algoritmo se inicia, na linha 1, com a construção de uma solução inicial, em que cada ponto de venda s da companhia recebe o preço máximo permitido, ou seja, $p_s = \overline{p}_s, \forall s \in S$. A condição de parada do algoritmo é verificada na linha 3, sendo alcançada quando o número máximo de iterações sem melhoria, definido por it_{MAX}, for atingido. O bloco de iterações do algoritmo está compreendido entre as linhas 4 e 15. Em cada iteração, realiza-se primeiramente uma busca local a partir da solução corrente, por meio do método $Local_Search$ (linha 4). Caso a solução resultante seja a melhor encontrada até o momento (definida pela função $Profit()$ equivalente a função objetivo 1), ela é armazenada e o contador de iterações sem melhora (it) e o nível de perturbação ($level$) são reinicializados (linhas 6-8). Caso a solução encontrada não atualize a melhor solução corrente, a intensidade da perturbação é aumentada por meio do incremento de $level$ (linha 11). Se $level$ já estiver no valor máximo permitido, ele é reiniciado com o valor mínimo, e contabiliza-se uma nova iteração sem melhoria (linhas 13-14). Na linha 15 o procedimento de perturbação ($Shake$) é aplicado com o objetivo de perturbar a solução corrente e explorar diferentes pontos do espaço de solução do problema. Por fim, na linha 16, o algoritmo é encerrado, retornando a melhor solução encontrada ao longo da execução.

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

4.1. Busca Local

Algorithm 2: Local_Search

```
   Data: P
1  continue = true;
2  while continue do
3  |   Profit_best ← Profit(P);
4  |   P ← Lower_Prices(P);
5  |   P ← Increase_Prices(P);
6  |   if Profit(P) = Profit_best then
7  |   |   continue = false;
8  return P
```

Um procedimento de busca local define, para uma solução dada do problema, uma vizinhança composta por um conjunto de soluções com características similares. A partir da solução corrente, realiza-se uma busca nessa vizinhança com o objetivo de identificar soluções com melhor desempenho. Caso uma solução superior seja encontrada, ela passa a ser a nova solução corrente, e o procedimento é repetido. Caso contrário, a solução corrente é considerada um ótimo local em relação à vizinhança adotada [Moura et al., 2021]. O Algoritmo 2 descreve o procedimento $Local_Search$ utilizado no método ILS. As iterações deste método ocorrem entre as linhas 3 e 7, sendo encerradas quando nenhuma melhoria adicional é obtida, o que é indicado pela atribuição da variável $continue = false$ na linha 7. Inicialmente, na linha 3, o melhor vizinho é definido como a própria solução corrente. A vizinhança de uma solução é construída por meio da alteração do preço em um dos pontos de venda. O procedimento $Lower_Prices$, executado na linha 4, gera o conjunto de soluções vizinhas da atual e seleciona, dentre elas, aquela que proporciona o maior valor de lucro. Em seguida, o método $Increase_Prices$, na linha 5, realiza ajustes pontuais na solução com o intuito de incrementar seu desempenho. Se o vizinho selecionado resultar em uma solução com lucro superior, conforme avaliado pela função $Profit()$, a variável de controle $continue$ é mantida como verdadeira, permitindo a continuidade da busca; esse vizinho, então, assume o papel de nova solução corrente. Caso contrário, considera-se que foi atingido um ótimo local, e o procedimento é encerrado.

A primeira vizinhança utilizada é definida pelo procedimento *Lower_Prices*, descrito no Algoritmo 3. Seu objetivo é conseguir "ganhar"compradores que na solução corrente estão comprando do concorrente, através do decremento do preço de venda de algum vendedor da companhia. O método inicia seu procedimento determinando na linha 1, o conjunto de todos os compradores atendidos pelo concorrente ($\overline{B}$), onde $s(b)$ é definido como o vendedor corrente do comprador $b \in B$. Enquanto o conjunto $\overline{B}$ não for vazio, o método iterativamente considera todos os compradores $b \in \overline{B}$, calcula seus custos de aquisição unitários $cost_b$ (linha 5), e verifica se existe algum vendedor da companhia $s \in S$ que possa ter seu preço de venda alterado para que o custo de aquisição para b fique uma unidade mais barata que o concorrente (linha 7). Desta forma, nesta solução vizinha, b (e, possivelmente outros consumidores) passarão a comprar deste ponto de venda da companhia. Em seguida, a função $Profit()$ é calculada (linha 8), e o melhor vizinho encontrado é armazenado (linhas 9-10). Note que quanto mais perto um vendedor está do comprador b, menor será o custo logístico e, portanto, menor o custo de aquisição para b. Portanto, consideramos apenas a diminuição de preços para vendedores que fiquem próximos do consumidor, definidos pela função $zone()$ (linha 6). A seguir, na linha 11, atualizamos a solução atual com o melhor vizinho encontrado (caso ele melhore a solução atual); e por fim, na linha 12, o conjunto $\overline{B}$ é atualizado, retirando

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

Algorithm 3: Lower_Prices

Data: P
1 $\overline{B} = \{b \in B \mid s(b) \in S^c\}$;
2 **while** $\overline{B} \neq \emptyset$ **do**
3 $\quad s_{best} \leftarrow 0;\ \ p_{best} \leftarrow 0;\ \ Profit_{best} \leftarrow 0;$
4 $\quad$ **for** $b \in \overline{B}$ **do**
5 $\quad\quad cost_b \leftarrow p_{s(b)} + c^{lo}_{b,s(b)};$
6 $\quad\quad$ **for** $s \in S$ *onde* $zone(s) = zone(b)$ **do**
7 $\quad\quad\quad new_p_s \leftarrow cost_b - c^{lo}_{bs}$ -1;
8 $\quad\quad\quad$ **if** $Profit(P, p_s \leftarrow new_p_s) > Profit_{best}$ **then**
9 $\quad\quad\quad\quad s_{best} \leftarrow s;\ \ p_{best} \leftarrow new_p_s;$
10 $\quad\quad\quad\quad Profit_{best} \leftarrow Profit(P, p_s \leftarrow new_p_s);$
11 $\quad Change(P, p_{s_{best}} \leftarrow p_{best});$
12 $\quad Update(\overline{B});$
13 **return** P

Algorithm 4: Increase_Prices

Data: P
1 **for** $s \in S$ **do**
2 $\quad Prices \leftarrow \emptyset;$
3 $\quad$ **for** $b \in b(s)$ **do**
4 $\quad\quad p_{s2}, c^{lo}_{bs2} \leftarrow Second_Cheapest_Seller(b);$
5 $\quad\quad Prices \leftarrow Prices \cup \{p_{s2} + c^{lo}_{bs2} - c^{lo}_{bs} - 1\};$
6 $\quad new_p_s \leftarrow Smallest(Prices);$
7 $\quad$ **if** $Profit(P, p_s \leftarrow new_p_s) > Profit(P)$ **then**
8 $\quad\quad Change(P, p_s \leftarrow new_p_s);$
9 **return** P

compradores que já foram considerados ou que passaram a comprar de vendedores da companhia.

O procedimento *Lower_Prices* gera vizinhos sempre baixando o preço de um ponto de venda. Observa-se que, em alguns casos, podemos obter vizinhos melhores com o aumento de preços. Com este objetivo, desenvolvemos nossa segunda vizinhança denominada $Increase_Prices$, descrita no Algoritmo 4. O método percorre cada ponto de venda $s \in S$ (linha 1) considerando todos os compradores b que compram de s (linha 3), onde $b(s)$ define os compradores que compram do ponto de venda s. Para cada comprador b, determinados o ponto de venda que oferece o segundo menor custo de aquisição para b (linha 4). Assim, calculamos o maior preço que s pode assumir sem perder b para este segundo ponto de venda e adicionamos este preço a um conjunto, $Prices$ (linhas 4-5). Dessa forma, para cada s, o menor de todos os preços do conjunto $Prices$ (guardado em new_{p_s}) é um preço que s pode assumir sem perder nenhum dos seus compradores na solução corrente. Logo, é verificado se alterando o preço de s para new_p_s, melhoramos o lucro da companhia (linha 7). Se for o caso, geramos uma nova solução com este novo preço, e armazenamos na solução corrente (linha 8). Após todos os pontos de venda serem verificados, o procedimento retorna a solução corrente.

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

4.2. Mecanismo de Perturbação

A fim de evitar que a heurística ILS fique estagnada em um ótimo local identificado pelo procedimento de busca local, e permitir a exploração de novas regiões do espaço de soluções, é incorporado um procedimento de perturbação responsável por modificar a solução atual. Tal procedimento visa gerar uma nova solução, potencialmente localizada em uma região distinta do espaço de busca, promovendo, assim, uma diversificação efetiva. O procedimento denominado $Shake$ é responsável por realizar essas perturbações, sendo parametrizado por dois elementos: $level$ e $step$. O parâmetro $level$ define a abrangência da perturbação, ou seja, a quantidade de pontos de venda cujos preços serão alterados — quanto maior o valor de $level$, maior será a extensão do espaço de soluções explorado. Por sua vez, o parâmetro $step$ controla a granularidade das modificações, determinando a magnitude das variações de preço. Valores menores de step induzem alterações mais sutis, enquanto valores maiores resultam em perturbações mais intensas.

Algorithm 5: Shake

Data: $P, level, step$
1 **for** $i = 1 \dots level$ **do**
2 $s \leftarrow Draw(S)$;
3 $inc \leftarrow (\overline{p}_s - \underline{p}_s)/step$;
4 **if** $Coin_Flip()$ **then**
5 $new_p_s \leftarrow \min(\overline{p}_s, p_s + inc)$;
6 **else**
7 $new_p_s \leftarrow \max(\underline{p}_s, p_s - inc)$;
8 $Change(P, p_s \leftarrow new_p_s)$
9 $P \leftarrow$ Refine_Prices(P);
10 **return** P

Algorithm 6: Refine_Prices

Data: P
1 **for** $s \in S$ **do**
2 $new_p_s \leftarrow +\infty$;
3 **for** $b \in b(s)$ **do**
4 $p_{s2}, c^{lo}_{bs2} \leftarrow Second_Cheapest_Seller(b)$;
5 **if** $new_p_s > (p_{s2} + c^{lo}_{bs2} - c^{lo}_{bs} - 1)$ **then**
6 $new_p_s \leftarrow p_{s2} + c^{lo}_{bs2} - c^{lo}_{bs} - 1$
7 **if** $new_p_s > p_s$ **then**
8 $Change(P, p_s \leftarrow new_p_s)$;
9 **return** P

O procedimento $Shake$ é apresentado no Algoritmo 5. No trecho compreendido entre as linhas 1 e 8, tem-se um laço que, a partir da solução atual P, percorre um subconjunto de pontos de venda, modificando seus respectivos preços com o intuito de gerar uma nova solução, a qual poderá conduzir a um novo ótimo local. Em cada iteração do laço, um ponto de venda é selecionado aleatoriamente (linha 2). Na sequência (linha 3), define-se um valor de incremento ou decremento para o preço desse ponto de venda, com base no intervalo de preços viável, o qual é particionado de acordo com o parâmetro $step$. Posteriormente (linha 4), realiza-se uma nova escolha aleatória para

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

determinar, com iguais probabilidades, se o preço do ponto de venda será aumentado ou reduzido (linhas 5 e 7). Após essa decisão, o novo preço new_p_s é atribuído ao ponto de venda s, conforme descrito na linha 8. Após a aplicação do procedimento de perturbação, os preços dos pontos de venda são modificados, resultando em uma nova solução. De maneira análoga ao que ocorre no método $Lower_Prices$, essa solução pode apresentar margem para reajustes de preços que não impliquem na perda de consumidores. Dessa forma, é possível elevar determinados preços com o intuito de aumentar o lucro da solução perturbada. Esse ajuste é realizado por meio do procedimento $Refine_Prices$ (linha 9).

O procedimento $Refine_Prices$, definido no Algoritmo 6, é conceitualmente igual ao método $Increase_Prices$, com a diferença de que, neste caso, não é necessário garantir um aumento imediato no lucro da solução resultante (linha 7). Isso se deve ao fato de que o principal objetivo do procedimento é promover a diversificação da busca, permitindo escapar de ótimos locais e explorar novas vizinhanças no espaço de soluções, com vistas à identificação de novos ótimos locais.

5. Resultados

Com o objetivo de testarmos a metaheurística proposta no presente trabalho, foram geradas novas instâncias que foram usadas tanto no método ILS proposto, quanto no método exato baseado no modelo matemático, com o objetivo de fazer uma comparação entre os resultados destes dois métodos.

Inicialmente, foi construída uma instância representativa de abrangência nacional (denominada $Nacional$), na qual o conjunto de consumidores é composto por todos os municípios brasileiros. Nessa instância, os conjuntos de pontos de venda da companhia e dos concorrentes são formados por 34 e 16 municípios, respectivamente. A partir dessa instância base, foram geradas outras 8 instâncias de forma aleatória (denominada $Aleatorias$), em que os consumidores são definidos como subconjuntos compostos por 1000, 1500, 2000, 2500, 3000, 3500, 4000 e 4500 municípios brasileiros. Além disso, foram construídas 10 instâncias representativas de 10 Unidades Federativas brasileiras (estados), escolhidas como os estados com maior número de municípios no país. Estas instancias são denominadas pelas respectivas siglas de cada estado (AM, MT, PA, PE, GO, PR, BA, RS, SP, MG). Complementarmente, foram geradas mais 5 instâncias regionais, cada uma representando uma das cinco grandes regiões geográficas do Brasil (denominadas pelas siglas de cada uma das respectivas regiões). Nas instâncias representativas de estados ou regiões, o conjunto de consumidores é composto exclusivamente pelos municípios pertencentes à respectiva unidade geográfica, enquanto os conjuntos de pontos de venda mantêm todos os municípios definidos na instância nacional. Ao todo, foram consideradas 24 instâncias de teste, que foram geradas e, consequentemente, preprocessadas utilizando o método de pré-processamento apresentado em Leal et al. [2023]. Para maiores detalhes de construção das instancias e pré-processamento, veja [Leal et al., 2023].

Os experimentos foram realizados em um ambiente computacional com processador Intel Core i5 com 1,60 GHz e 8 GB de RAM, e com sistema operacional Linux Debian. Os métodos foram implementados na linguagem de programação C++, utilizando o CPLEX v12.10 como solver, utilizando uma única *thread*, com tempo máximo de processamento de 1 hora. Para encontrar a melhor configuração dos parâmetros do ILS, foi utilizada a ferramenta Irace (López-Ibánez [2011]), desenvolvida para a calibração automática de parâmetros de algoritmos de otimização. Seu principal objetivo é encontrar a melhor configuração dos parâmetros, dado um conjunto de instâncias do problema. A melhor configuração dos parâmetros de entrada, obtida usando instancias não consideradas nos experimentos, foi: $it_{MAX} = 30$, $level_{INI} = 2$, $level_{MAX} = 16$, $step = 5$.

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

Os resultados do método exato e o método ILS aqui proposto são mostrados na Tabela 3. As linhas da tabela (instâncias) estão ordenadas pelo número de consumidores. As três primeiras colunas contêm a identificação da instância ($|S|$ é a quantidade de pontos de venda da companhia, $|S^c|$ é a quantidade de competidores e $|B|$ é a quantidade de consumidores da instância). As duas colunas seguintes mostram o resultado do pré-processamento, ou seja, as novas quantidades de pontos de venda da companhia e dos competidores (a quantidade de consumidores não se altera). Em seguida, temos uma coluna com o tipo da Instância. As 2 colunas seguintes trazem o resultado do método exato: o valor da função objetivo encontrada (FO_{EX}), e o tempo de processamento gasto em segundos (T_{EX}), onde o símbolo '-' indica que o limite de tempo de uma hora foi ultrapassado. Caso a solução ótima for encontrada, os valores de FO_{EX} estarão em negrito. As colunas seguintes trazem os resultados do método ILS: a coluna FO_{ILS} traz o valor da função objetivo, a coluna T_{ILS}, o tempo de processamento em segundos, e a coluna $gap\%$, o valor do gap entre a solução heurística e a exata. Este gap é calculado pela fórmula: $gap\% = ((FO_{EX} - FO_{ILS})/FO_{EX}) * 100$.

$\|S\|$	$\|S^c\|$	$\|B\|$	$\|S\|_P$	$\|S^c\|_P$	$Tipo$	FO_{EX}	T_{EX}	FO_{ILS}	T_{ILS}	$gap\%$
34	16	62	2	1	AM	**13.528.907**	0.00	13.528.755	0.00	0.00
34	16	141	24	2	MT	**11.418.150**	4.00	11.190.242	0.00	2.00
34	16	144	5	0	PA	**29.020.219**	0.00	28.846.292	0.00	0.60
34	16	185	7	2	PE	**28.458.489**	4.00	28.456.818	0.00	0.01
34	16	246	24	3	GO	28.469.090	-	28.414.499	3.00	0.19
34	16	399	11	3	PR	**56.536.333**	7.00	56.535.640	0.00	0.00
34	16	417	7	1	BA	**32.082.994**	0.00	31.934.464	0.00	0.46
34	16	450	18	2	Norte	**70.519.058**	23.00	70.459.564	2.80	0.08
34	16	467	24	4	Centro-Oeste	79.139.746	-	79.086.858	8.20	0.07
34	16	497	6	2	RS	**63.155.577**	0.00	63.154.805	0.00	0.00
34	16	645	17	3	SP	165.792.108	-	164.715.902	8.00	0.65
34	16	853	16	4	MG	**76.772.178**	0.00	76.768.145	1.00	0.01
34	16	1000	34	14	Aleatorio	**145.572.766**	6.00	145.424.897	6.60	0.10
34	16	1191	11	4	Sul	**179.313.056**	962.00	179.310.005	4.20	0.00
34	16	1500	33	10	Aleatorio	216.346.640	-	214.883.425	108.20	0.68
34	16	1668	22	5	Sudeste	308.091.554	-	307.081.012	40.00	0.33
34	16	1794	13	2	Nordeste	161.931.763	-	162.352.666	7.40	-0.26
34	16	2000	33	10	Aleatorio	327.399.817	-	322.657.972	30.80	1.45
34	16	2500	33	10	Aleatorio	336.236.803	-	344.447.569	55.20	-2.44
34	16	3000	33	10	Aleatorio	465.689.552	-	469.898.405	142.40	-0.90
34	16	3500	33	10	Aleatorio	508.268.174	-	510.992.101	115.80	-0.54
34	16	4000	33	10	Aleatorio	569.280.323	-	575.981.054	152.20	-1.18
34	16	4500	33	10	Aleatorio	617.313.594	-	644.551.870	257.60	-4.41
34	16	5570	33	10	Nacional	740.770.530	-	786.678.166	240.00	-6.20

Tabela 3: Resultado do método Meta-Heurístico ILS.

A Tabela 3 evidencia que a dificuldade de resolução do problema PMCOI está fortemente correlacionada ao número de consumidores presentes na instância. À medida que o número de consumidores aumenta, observa-se um crescimento no tempo necessário tanto para a obtenção de soluções exatas quanto heurísticas. Especificamente, a partir de 1500 consumidores, o método exato não é mais capaz de atingir a otimalidade dentro do limite de tempo de uma hora. A partir de 2500 consumidores, o método heurístico ILS passou a apresentar resultados superiores aos obtidos pelo método exato, sendo que essa diferença se acentua progressivamente conforme o tamanho da instância aumenta, alcançando um gap de -6,2 na maior instância testada (denominada Nacional). Considera-se esse desempenho bastante satisfatório, uma vez que o ILS apresentou um gap médio

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

de apenas 0,39 em relação ao método exato, mantendo, ao mesmo tempo, tempos de processamento significativamente inferiores — com um tempo máximo de 10 segundos para instâncias com até 1200 consumidores e até 4 minutos para instâncias de maior porte.

Tais resultados demonstram tanto a robustez quanto a eficiência do método ILS. Ademais, considerando o foco principal deste estudo — a instância Nacional —, o método heurístico foi capaz de encontrar uma solução significativamente superior àquela obtida pelo método exato.

6. Conclusão e Trabalhos Futuros

Os resultados obtidos demonstram que a heurística ILS foi capaz de alcançar, no mínimo, valores muito próximos ao ótimo fornecido pelo método exato em todas as instâncias avaliadas. No caso da maior instância — denominada Nacional e considerada o foco principal deste trabalho —, a heurística obteve um resultado significativamente superior ao do método exato. Dado que o problema PMCOI envolve a definição de preços por parte de uma companhia com elevado volume de vendas e margens expressivas de lucro, mesmo um aumento percentual relativamente pequeno, como 6,20%, que foi o resultado obtido, representa um incremento financeiro de grande magnitude.

Dessa forma, conclui-se que o objetivo principal deste estudo foi plenamente alcançado: a heurística proposta apresentou melhorias expressivas no lucro obtido para a maior instância do problema, evidenciando o potencial do método metaheurístico ILS em aplicações práticas. Assim, sua utilização mostra-se vantajosa não apenas para a companhia em questão, mas também para outras empresas com características operacionais semelhantes.

Como direções para trabalhos futuros, propõe-se a extensão do problema PMCOI para contemplar a presença de restrições de capacidade, isto é, considerando a limitação nas quantidades disponíveis para venda. Outra possível evolução consiste em adaptar o modelo para lidar com a precificação de múltiplos produtos simultaneamente. Além disso, dado que os preços praticados pelos concorrentes são fornecidos como dados de entrada nas instâncias atuais, uma alternativa relevante seria incorporar ao modelo a dinâmica de equilíbrio de preços entre os pontos de venda da companhia e os de seus concorrentes.

Referências

Ahmadi-Javid, A., Amiri, E., e Meskar, M. (2018). A profit-maximization location-routing-pricing problem: A branch-and-price algorithm. *European Journal of Operational Research*, 271 (3):866–881. ISSN 0377-2217. URL https://www.sciencedirect.com/science/article/pii/S0377221718301358.

Böhnlein, T., Schaudt, O., e Schauer, J. (2023). Stackelberg packing games. *Theor. Comput. Sci.*, 943(C):16–35. ISSN 0304-3975. URL https://doi.org/10.1016/j.tcs.2022.12.006.

Briest, P., Hoefer, M., e Krysta, P. (2012). Stackelberg network pricing games. *Algorithmica*, 62(3): 733–753. ISSN 1432-0541. URL https://doi.org/10.1007/s00453-010-9480-3.

Bui, Q. M., Carvalho, M., e Neto, J. (2025). Solving combinatorial pricing problems using embedded dynamic programming models. *INFORMS Journal on Computing*. ISSN 1526-5528. URL http://dx.doi.org/10.1287/ijoc.2024.0686.

Cardinal, J., Demaine, E. D., Fiorini, S., Joret, G., Langerman, S., Newman, I., e Weimann, O. (2009). The stackelberg minimum spanning tree game. *Algorithmica*, 59(2):129–144. ISSN 1432-0541. URL http://dx.doi.org/10.1007/s00453-009-9299-y.

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br

Leal, H., Sotelo, D., e Abitbol, Y. (2023). Precificação em mercados competitivos com oferta irrestrita. In *Anais do XXVIII SBPO*, https://proceedings.science/sbpo/sbpo-2023/trabalhos/precificacao-em-mercados-competitivos-com-oferta-irrestrita?lang=pt-br. SOBRAPO. Acessado: 2024-06-01.

Lin, Y. H. e Tian, Q. (2023). Facility location and pricing problem: Discretized mill price and exact algorithms. *European Journal of Operational Research*, 308(2):568–580. ISSN 0377-2217. URL https://www.sciencedirect.com/science/article/pii/S0377221722009250.

Lourenço, H. R., Martin, O. C., e Stützle, T. (2019). *Iterated Local Search: Framework and Applications*, p. 129–168. Springer International Publishing, Cham. ISBN 978-3-319-91086-4. URL https://doi.org/10.1007/978-3-319-91086-4_5.

López-Ibánez, M. (2011). The irace package, iterated race for automatic algorithm configuration. Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles.

Moura, A., Noronha, T., e Bogue, E. (2021). Uma heurística ils para o problema da mochila com penalidades. In *ANAIS DO SIMPóSIO BRASILEIRO DE PESQUISA OPERACIONAL, 2021, João Pessoa.*, https://proceedings.science/sbpo/sbpo-2021/trabalhos/uma-heuristica-ils-para-o-problema-da-mochila-com-penalidades?lang=pt-br. Acessado: 2024-06-01.

Pferschy, U., Nicosia, G., Pacifici, A., e Schauer, J. (2021). On the stackelberg knapsack game. *European Journal of Operational Research*, 291(1):18–31. ISSN 0377-2217. URL https://www.sciencedirect.com/science/article/pii/S0377221720307931.

Plyasunov, A. V. e Panin, A. A. (2013). The pricing problem part i: Exact and approximate algorithms. *J. Appl. Indust. Math.*, 7:241–251.

van Hoesel, S. (2008). An overview of stackelberg pricing in networks. *European Journal of Operational Research*, 189(3):1393–1402. ISSN 0377-2217. URL https://www.sciencedirect.com/science/article/pii/S037722170700625X.

DOI: 10.59254/sbpo-2025-212581
https://proceedings.science/p/212581?lang=pt-br