João Victor da Silva Feres de Souza

Algoritmo *Iterated Local Search* com *Simulated Annealing* para Problema de Rotulação Cartográfica de Pontos

ALEGRE - ES

2024

João Victor da Silva Feres de Souza

Algoritmo *Iterated Local Search* com *Simulated Annealing* para Problema de Rotulação Cartográfica de Pontos

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Espírito Santo

Centro de Ciências Exatas, Naturais e da Saúde

Departamento de Computação

Orientador: Prof. Dr. Edmar Hell Kampke

ALEGRE - ES

2024

João Victor da Silva Feres de Souza

Iterated Local Search com Simulated Annealing para Problema de Rotulação Cartográfica de Pontos

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 01 de outubro de 2024.

COMISSÃO EXAMINADORA

Prof. Dr. Edmar Hell Kampke
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Geraldo Regis Mauri
Universidade Federal do Espírito Santo

Prof. Dr. Rodrigo Freitas Silva
Universidade Federal do Espírito Santo

Agradecimentos

Primeiramente, agradeço a Deus por me guiar e amparar em todos os momentos vividos durante a graduação.

À minha família, especialmente à minha mãe Alyne, cujo apoio inicial foi fundamental para que eu pudesse seguir em frente. À minha querida avó, que, mesmo em sua ausência, continua sendo uma fonte inesgotável de ensinamentos e carinho, tendo cuidado de mim durante grande parte da minha vida.

Aos professores do DCOMP, pelos valiosos ensinamentos transmitidos ao longo desta jornada. Em especial, ao professor Edmar, pela orientação e apoio na realização deste trabalho de conclusão de curso, e ao professor Geraldo, pelo suporte com ideias e sugestões durante o desenvolvimento do trabalho.

Aos amigos que fiz durante a faculdade e aos meus companheiros de república, com quem compartilhei momentos inesquecíveis.

A todos, o meu sincero agradecimento.

Resumo

Este trabalho apresenta a combinação dos algoritmos *Iterated Local Search* (ILS) e *Simulated Annealing* (SA) para solucionar o Problema de Rotulação Cartográfica de Pontos (PRCP), que consiste em posicionar rótulos em mapas de forma a evitar sobreposições e garantir a legibilidade. Portanto, o objetivo é otimizar a disposição dos rótulos nos pontos de um mapa em um menor tempo, considerando restrições de sobreposição. O algoritmo híbrido proposto integra a meta-heurística ILS, responsável por introduzir perturbações a partir de uma solução ótima local, visando aumentar a diversificação no espaço de busca, o processo de busca é realizado utilizando a meta-heurística Simulated Annealing (SA). Os resultados obtidos apresentam equivalência e proximidade em relação à literatura, com um número reduzido de repetições.

Palavras-chave: Problema de Rotulação Cartográfica, *Simulated Annealing*, *Iterated Local Search*.

Abstract

This work presents the combination of the *Iterated Local Search* (ILS) and *Simulated Annealing* (SA) algorithms to solve the Point Feature Label Placement Problem (PFLP), which involves positioning labels on maps to avoid overlaps and ensure legibility. Therefore, the goal is to optimize the placement of labels on the points of a map in the shortest time, considering overlap constraints. The proposed hybrid algorithm integrates the ILS metaheuristic, responsible for introducing perturbations from a local optimal solution to increase diversification in the search space, while the search process is carried out using the Simulated Annealing (SA) metaheuristic. The results obtained show equivalence and proximity to the literature, with shorter computational times and a reduced number of repetitions.

Keywords: Point Feature Label Placement Problem, Simulated Annealing, Iterated Local Search

Lista de Figuras

Figura 1 – Mapa do Espírito Santo como exemplo de sobreposição de rótulos. . . 12

Figura 2 – Representação da padronização do ponto. 17

Lista de Tabelas

Tabela 1 – Instâncias para o problema cartográfico de pontos. 24
Tabela 2 – Parâmetros para o problema cartográfico de pontos. 25
Tabela 3 – Percentual (%) de Pontos Rotulados Sem Conflitos. 25
Tabela 4 – Resultados Obtidos para as instâncias de 13206 pontos. 26
Tabela 5 – Percentual (%) de 13206 Pontos Rotulados Sem Conflitos 27

Lista de Algoritmos

1 Algoritmo genérico ILS 19

2 Algoritmo *SA* Adaptado de Rocha (2013) 23

Lista de Abreviaturas e Siglas

PRCP	Problema de Rotulação Cartográfica de Pontos
SA	*Simulated Annealing*
ILS	*Iterated Local Search*
CS	*Clustering Search*

Sumário

1 Introdução . . . 11

1.1 O Problema e sua Importância . . . 11

1.2 Objetivos . . . 13

1.2.1 Objetivo Geral . . . 13

1.2.2 Objetivos Específicos . . . 13

2 Revisão de Literatura . . . 14

3 Metodologia . . . 17

3.1 Modelo Matemático do Problema . . . 17

3.2 Meta-Heurística *Iterated Local Search* . . . 19

3.2.1 Construção da Solução Inicial . . . 19

3.2.1.1 Algoritmo de Construção da Solução Inicial . . . 20

3.2.2 Pertubação . . . 20

3.2.2.1 Geração de Vizinhança com Troca de Uma Posição Aleatória 20

3.2.2.2 Geração de Vizinhança com Troca de Duas Posições Aleatórias 21

3.2.2.3 Geração de Vizinhança com Modificação de Múltiplos Pontos . . . 21

3.2.3 Busca local . . . 21

3.2.3.1 *Simulated Annealing* . . . 21

4 Resultados Computacionais . . . 24

4.1 Instâncias . . . 24

4.2 Experimentos Computacionais . . . 24

4.3 Resultados Obtidos . . . 25

5 Conclusão . . . 28

Referências . . . 29

1 Introdução

A cartografia é uma ciência dedicada à representação visual da Terra e de suas características geográficas, desempenhando um papel fundamental na nossa compreensão do mundo. De acordo com Oliveira (1988), a cartografia é a ciência que se dedica à elaboração de mapas, proporcionando uma representação visual e precisa do espaço geográfico, e abrange todas as fases dos trabalhos, desde os primeiros levantamentos até a impressão final dos mapas. Para elaboração de mapas existem técnicas que organizam as informações geográficas. Uma das formas de organizar é utilizando o sistema de rotulação, que trata como definir rótulos para representar as informações (ALVES, 2011). A rotulação em mapas permite que os usuários identifiquem e compreendam a localização de cidades, pontos de interesse, acidentes geográficos, recursos naturais e uma infinidade de outros elementos de um mapa. Além disso, a escolha das técnicas adequadas para a rotulação de pontos representa um desafio significativo, uma vez que envolve considerações de legibilidade, estética e eficiência. Para rotular manualmente um mapa, por exemplo, estima-se que leve 50% do tempo total de produção do mapa (WOLFF, 2002).

Nesse contexto, surgiu na área de otimização o desafio de encontrar a melhor forma de rotulação cartográfica de pontos. O Problema de Rotulação Cartográfica de Pontos (PRCP) é um campo de pesquisa interdisciplinar que busca soluções eficientes e eficazes para esse desafio. Segundo Christensen, Marks e Shieber (1995), o PRCP é classificado como NP-difícil. Isso significa que o problema é tão difícil de resolver que até o momento, não foi provada a existência de um algoritmo exato e eficiente capaz de solucioná-lo.

Portanto, para enfrentar o desafio do PRCP, é comum recorrer a métodos que incluem heurísticas e meta-heurísticas. Esses métodos são algoritmos aproximativos que não garantem a otimalidade, mas têm o objetivo de encontrar soluções aproximadas em um tempo de processamento curto.

1.1 O Problema e sua Importância

O PRCP é um problema que envolve a clara identificação de entidades cartográficas por meio de rótulos, que são colocados em regiões específicas de um mapa geográfico (OLIVEIRA, 2012). Segundo Cravo, Ribeiro e Lorena (2006) o problema consiste em posicionar os rótulos dos pontos de forma que não ocorram sobreposições. Os locais no mapa onde é possível posicionar os rótulos dos pontos cartográficos são chamados de posições candidatas. Exemplos de sobreposição em um mapa podem ser vistos na Figura 1, na qual as setas indicam locais onde os rótulos de dois ou mais pontos (cidades) estão sobrepostos.

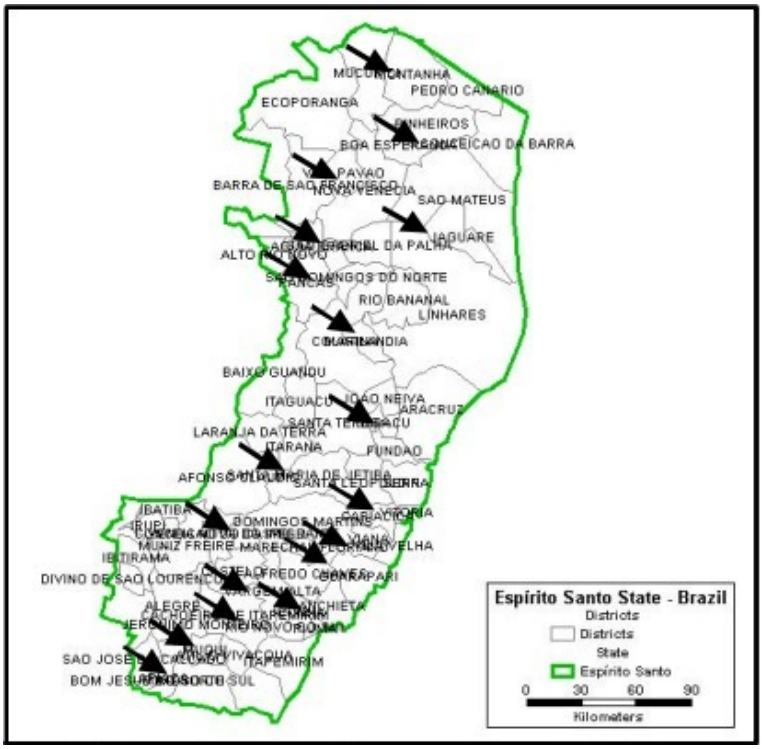

Figura 1 – Mapa do Espírito Santo como exemplo de sobreposição de rótulos.

Fonte: (RIBEIRO; LORENA, 2008).

O PRCP pode ser caracterizado como um problema de otimização combinatória, que envolve a disposição dos rótulos dos pontos em posições candidatas específicas (MAURI, 2008). As sobreposições que surgem ao posicionar os rótulos em posições candidatas podem ser representadas por um grafo de conflitos G = (V, A), onde V é o conjunto de vértices das posições candidatas, e A é o conjunto de arestas que representam conflitos entre as posições candidatas. O PRCP envolve a escolha de uma única posição viável em cada subconjunto de vértices adjacentes.

O PRCP desempenha um papel fundamental em diversos aspectos da cartografia e da geoinformação, algumas de suas contribuições são:

1. Navegação e GPS: Em sistemas de navegação e GPS, a rotulação de pontos é vital para orientar motoristas, viajantes e embarcações, auxiliando na identificação de destinos e pontos de interesse;

2. Comunicação Efetiva de Informações: Uma boa rotulação cartográfica ajuda na comunicação eficaz de informações geográficas, tornando os mapas mais acessíveis a um público amplo, desde geógrafos e urbanistas, até o público em geral;

3. Melhoria da Legibilidade de Mapas: O PRCP contribui para tornar os mapas mais legíveis, facilitando a interpretação e compreensão das informações geográficas representadas, o que é crucial para a navegação, planejamento urbano, e outras aplicações.

A solução do PRCP traz benefícios tangíveis em várias aplicações práticas e acadêmicas, e é fundamental para o desenvolvimento de soluções avançadas em sistemas de informação geográfica, planejamento urbano, logística, transporte e muito mais.

1.2 Objetivos

Nesta seção são apresentados o objetivo geral e os objetivos específicos deste trabalho.

1.2.1 Objetivo Geral

O propósito abrangente deste estudo foi empregar a meta-heurística *Iterated Local Search* (ILS) com a incorporação do método *Simulated Annealing* (SA) no processo de aprimoramento (busca local), para resolver o PRCP.

1.2.2 Objetivos Específicos

Os objetivos específicos desse trabalho são listados abaixo:

a) Analisar os dados essenciais para abordar a resolução do PRCP.

b) Desenvolver um algoritmo híbrido que promova a integração dos métodos ILS e SA.

c) Conduzir testes de validação e ajuste de parâmetros no algoritmo desenvolvido, utilizando instâncias específicas do problema.

d) Realizar eventuais correções no algoritmo.

e) Executar testes com o algoritmo desenvolvido, utilizando as instâcias propostas na literatura.

f) Avaliar o desempenho do algoritmo, comparando os resultados obtidos com os que existem na literatura.

2 Revisão de Literatura

Christensen, Marks e Shieber (1995) propuseram diversos algoritmos para a resolução do PRCP, reconhecendo que, dependendo da instância, o problema pode não ser resolvido em tempo hábil. Entre as abordagens incluídas está um algoritmo de busca exaustiva, que envolve a ordenação de variáveis e valores. Além disso, introduziram uma padronização cartográfica, que representa uma contribuição significativa tanto para a literatura quanto para a elaboração de futuros trabalhos. Nesta padronização consideram-se oito possíveis posições candidatas para cada ponto, atribuindo a cada uma delas uma preferência cartográfica, onde cada posição é identificada por um número.

A partir desse padrão, o PRCP pode ser definido como o problema de atribuir rótulos aos pontos em uma de suas posições candidatas disponíveis, sujeito a restrições de conflito e minimizar ou maximizar uma função objetivo (MAURI; RIBEIRO; LORENA, 2010).

A função objetivo está relacionada à atribuição de rótulos a uma das posições candidatas. A qualidade da rotulagem está associada à quantidade de sobreposição de rótulos de texto, recursos gráficos e preferências a priori, dentro de um conjunto canônico de possíveis posições de rótulo (CHRISTENSEN; MARKS; SHIEBER, 1995 apud YOELI, 1972).

O uso do algoritimo *Greedy Randomized Adaptive Search Procedure* (GRASP) foi proposto por Cravo, Ribeiro e Lorena (2008) como um caminho para a resolução do PRCP. Cravo, Ribeiro e Lorena (2008) empregaram uma busca local para encontrar um valor que permitisse iniciar uma solução viável, seguida pela realização de permutações entre as listas de posições candidatas até encontrar a melhor solução. Os resultados obtidos demonstraram melhor desempenho do que os apresentados por Christensen, Marks e Shieber (1995).

No trabalho de Oliveira, Urrutia e Noronha (2010), foi utilizada uma heurística de busca local combinada com a técnica de *backtracking*. A estratégia adotada incluiu o uso de um pré-processamento para reduzir o tamanho do grafo de conflitos antes da aplicação da heurística. Os resultados obtidos mostraram-se equivalentes aos melhores já reportados até então na literatura.

Outro estudo relevante na literatura para o PRCP foi apresentado por Mauri, Ribeiro e Lorena (2010), que propuseram novas abordagens para resolver o problema, com destaque para soluções de alta qualidade em instâncias com até 1000 pontos. Além disso, o trabalho introduziu novas estratégias baseadas na aplicação da Decomposição Lagrangiana, ampliando o leque de técnicas disponíveis para lidar com o PRCP.

Oliveira (2012) propôs o *Variable neighborhood Descent* (VND), que realiza permutações entre vizinhanças exploradas por um algoritmo de *backtracking*, para o PRCP. O procedimento é executado em três etapas: primeiramente, é aplicado um pré-processamento no grafo de conflitos para reduzir o número de vértices e arestas, o que consequentemente diminui o tamanho da busca. Em seguida, a solução inicial do problema é construída por meio de uma heurística construtiva gulosa, e por fim, o VND com o *backtracking* é aplicado. Os resultados obtidos mostraram que o método desenvolvido apresenta bom desempenho em instâncias em que as vizinhanças são grandes.

Rabello et al. (2013) implementaram estratégias visando a resolução do PRCP, as quais demonstraram uma significativa melhoria nos resultados em comparação aos apresentados por Mauri, Ribeiro e Lorena (2010). Além disso, Rabello et al. (2013) obteve resultados de qualidade para instâncias com 13206 pontos, no padrão de oito posições, utilizando a meta-heurística *Clustering Search* (CS) com o intuito de oferecer uma abordagem alternativa para a solução do PRCP.

Existem diversas estratégias que empregam diferentes meta-heurísticas para a resolução de problemas complexos. O *Iterated Local Search* (ILS) é uma meta-heurística que se destaca como uma técnica amplamente utilizada, que tem demonstrado consistentemente sua eficiência na resolução desses problemas.

Subramanian et al. (2013) utilizaram o ILS como base para desenvolver um algoritmo com o objetivo de resolver diversas variantes do Problema de Roteamento de Veículos (PRV). Subramanian et al. (2013) aplicaram duas estratégias de inserção para gerar soluções iniciais diversificadas: a Estratégia de Inserção Sequencial (EIS) e a Estratégia de Inserção Paralela (EIP). Também empregaram o algoritmo *Randomized Variable Neighborhood Descent* (RVND) na busca local, o qual utiliza a aleatoriedade para explorar o espaço de soluções nas vizinhanças. Os resultados demonstram que o algoritmo proporcionou soluções de alta qualidade e baixa variabilidade, além de alcançar mais de 80% dos melhores resultados encontrados na literatura.

Vieira et al. (2022) aplicaram o ILS para resolver o Problema de Roteamento de Veículos com Frota Heterogênea (PRVFH). Os autores aplicaram um procedimento de economias, criado por Clarke-Wright, para gerar a solução inicial, e também apresentaram métodos com os movimentos *two-opt* e *two-point-move*, os quais contribuem para o processo de busca local, resultando na identificação da melhor solução. Os resultados melhoraram 23% quando comparados com a solução inicial emitida pela heurística de Clarke-Wright, sendo que foi possível atingir tal resultado com tempo inferior a sete segundos.

Outra abordagem que tem demonstrado resultados promissores, assim como o ILS, é o *Simulated Annealing* (SA), que é eficaz na exploração do espaço de solução, permitindo escapar de ótimos locais. No estudo realizado por Carvalho et al. (2016), empregou-

se a meta-heurística SA para solucionar o Problema de Programação de Horários em Universidades (PPHU). Nesse contexto, os pesquisadores inicialmente utilizaram uma heurística gulosa que serviu como base para a aplicação do SA. Os resultados alcançados demonstraram notável eficiência. Quando comparados às soluções criadas manualmente, ficou evidente que a meta-heurística obteve um desempenho 32,9% superior, tudo isso em um tempo impressionantemente curto de 186 segundos.

Kampke et al. (2022) exploraram a aplicação da técnica SA para resolver o Problema de Transporte de Máquinas em Florestas de Produção (PTMFP), o qual é uma variante do Problema de Roteamento de Veículos com Coleta e Entrega (PRVCE). Os resultados obtidos revelaram-se promissores, uma vez que, em algumas instâncias, o desempenho alcançado superou em 2,58% o melhor resultado previamente conhecido na literatura.

Dado o notável desempenho das estratégias SA e ILS na resolução de problemas de otimização, Silva e Souza (2008) aplicaram essas meta-heurísticas para solucionar o desafiante Problema de Fluxo Multiproduto Inteiro Capacitado (PFMIC). Os resultados obtidos, em um tempo de execução aceitável, demonstraram a qualidade ótima das soluções. Vale ressaltar que um aspecto importante desse estudo é a incorporação da restrição de capacidade no espaço de busca da meta-heurística híbrida proposta, com uma penalização controlada por meio de uma relaxação, visando considerar a restrição de conservação de fluxo.

Monteiro et al. (2017) desenvolveram um algoritmo híbrido que combina o ILS com o SA como mecanismo de busca local, além de incorporar a Cadeia de Kempe como operação de permutação para resolver o Problema de Tabela-Horário de Universidades (PTHU). Os resultados obtidos foram comparados às soluções apresentadas na segunda edição do *International Timetabling Competition* (ITC-2007), bem como a outros resultados documentados na literatura. O estudo destacou a eficácia da combinação do ILS com o SA na exploração do espaço de soluções, evidenciando o potencial dessa abordagem para alcançar soluções de qualidade.

3 Metodologia

Neste estudo foi adotada a meta-heurística ILS, em conjunto com o SA, que é utilizado para aprimorar a busca local, para a resolução do PRCP. As instâncias empregadas nos testes são frequentemente encontradas em outras pesquisas da literatura e estão disponíveis para consulta (LORENA, 2024).

3.1 Modelo Matemático do Problema

Utilizando o método de padronização, proposto por Christensen, Marks e Shieber (1995), que consiste em separar as posições candidatas em modelos com 4 ou 8 posições, conforme apresentado na Figura 2, é possível formalizar as posições de rótulos usando variáveis binárias. As posições candidatas são representadas por essas variáveis, onde a variável correspondente a uma posição escolhida é definida como 1; caso contrário, é atribuída como 0. Isso cria um grafo de conflitos que modela sobreposições de rótulos, representando possíveis seleções simultâneas de posições candidatas. Vértices adjacentes indicam conflitos resultantes de sobreposições.

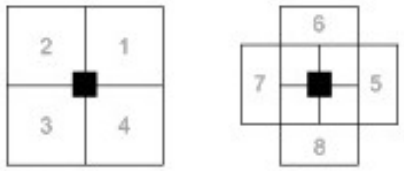

Figura 2 – Representação da padronização do ponto.

Fonte: (CHRISTENSEN; MARKS; SHIEBER, 1995).

O Problema do Máximo Conjunto Independente de Vértice (PMCIV) também pode ser modelado através de um grafo de conflitos. Em termos matemáticos, dado um grafo G=(V,A), onde V é o conjunto de vértices e A é o conjunto de arestas, um conjunto $I \subseteq V$ é um conjunto independente se nenhum par de vértices em I são adjacentes, ou seja, não há arestas conectando esses vértices. Desta forma, o objetivo é buscar a maximização da cardinalidade de I.

O modelo matemático utilizado neste trabalho é basedo na abordagem criada por Zoraster (1990), no qual propôs utilizar o PRCP como um PMCIV, onde a sua resolução está atrelada a maximizar a cardinalidade do conjunto I, que é equiparado a obter o maior número de pontos rotulados sem conflitos em um modelo de grafo de conflitos para o PRCP. O modelo citado inclui as seguintes variáveis:

- i expressa um ponto.
- j expressa uma posição candidata.
- m é o número de pontos.
- p é a quantidade de posições candidatas.
- $X_{i,j}$ é a váriavel de decisão que indica uma possível escolha de uma posição candidata j para o ponto i. Se $X_{i,j}$ é 1, indica que o ponto i foi rotulado na posição j; caso contrário, $X_{i,j}$ é 0.
- $S_{i,j}$ é o conjuto de posições candidatas adjacentes a posição candidata j do ponto i.

Desta forma, considerando a solução s sendo avaliada pela função objetivo f, o modelo mátematico pode ser descrito da seguinte maneira:

$$\text{Maximizar} \quad f(s) = \sum_{X_{i,j} \in V} X_{i,j} \tag{3.1}$$

Sujeito a:

$$\sum_{j=1}^{p} X_{i,j} \leq 1, \quad \forall i = \{1, ..., m\} \tag{3.2}$$

$$X_{i,j} + X_{k,t} \leq 1, \quad \forall i \in \{1, ..., m\}, \forall j \in \{1, ..., p\}, \forall (k,t) \in S_{i,j} \tag{3.3}$$

$$X_{i,j}, X_{k,t} \in \{0, 1\}, \forall X_{i,j}, X_{k,t} \in V \tag{3.4}$$

A função objetivo $f(s)$ (3.1) tem como finalidade maximizar a quantidade de pontos rotulados, buscando identificar o máximo de posições que podem ser escolhidas para rotulação sem sobreposição na solução s. A restrição (3.2) assegura que cada rótulo seja atribuido a no máximo uma posição candidata, possibilitando a existência de posições não rotuladas. A restrição (3.3) garante que, no máximo um vértice, dentre dois vértices adjacentes no grafo de conflitos, seja selecionado. A restrição (3.4) estabelece limites para as variáveis inteiras envolvidas no problema, proporcionando coerência ao modelo.

3.2 Meta-Heurística *Iterated Local Search*

A meta-heurística *Iterated Local Search* (ILS) foi introduzida por Lourenço, Martin e Stützle (2003). Ela é uma abordagem de busca local que visa explorar o espaço de soluções por meio de perturbações em soluções ótimas-locais identificadas durante a busca. Essa metodologia, essencialmente, busca melhorar a qualidade das soluções encontradas, gerando novas soluções através de pertubações nas soluções ótimas-locais previamente identificadas. O pseudocódigo genérico da meta-heurística ILS é exibido no Algoritmo 1.

Algoritmo 1: Algoritmo genérico ILS

Entrada: ILS_{max}
Saída: Solução S_{Melhor}

1 **início**
2 $S_{Melhor} \leftarrow geraSoluçãoInicial();$
3 $S_{Melhor} \leftarrow buscaLocal(S_{Melhor});$
4 **para** $i \leftarrow 1$ **até** ILS_{max} **faça**
5 $S_i \leftarrow pertubação(S_{Melhor});$
6 $S_i \leftarrow buscaLocal(S_i);$
7 **se** $f(S_i) > f(S_{Melhor})$ **então**
8 $S_{Melhor} \leftarrow S_i;$
9 **fim se**
10 **fim para**
11 **fim**

O algoritmo ILS adota uma estratégia que se inicia com a construção de uma solução inicial, conforme indicado na linha 2. Essa etapa pode ser realizada de maneira aleatória ou por meio de alguma heurística específica. Subsequentemente, após a obtenção da solução inicial, na linha 3 é aplicado um procedimento que envolve uma busca local, resultando em uma solução ótima-local. No intervalo de linhas 4 a 10, ocorre o procedimento iterativo da meta-heurística. Na linha 5, procede-se à geração de uma pertubação, na solução atual, para escapar de ótimos locais e explorar outras regiões do espaço de soluções. Posteriormente, na linha 6, é realizada uma iteração adicional de busca local com o intuito de explorar um novo espaço de soluções e aprimorar a solução atual. Por último, no intervalo das linhas 7 a 9, é aplicado o critério de aceitação, que avalia se a solução ótima-local é a melhor solução obtida até o momento.

3.2.1 Construção da Solução Inicial

Para a construção da solução inicial foi empregada uma heurística gulosa que busca encontrar uma solução aceitável para o problema, fazendo escolhas locais ótimas em cada passo do processo.

3.2.1.1 Algoritmo de Construção da Solução Inicial

Neste trabalho, para a construção de uma solução inicial, foi desenvolvida uma heurística gulosa que rotula os pontos individualmente. A cada iteração, a heurística seleciona a posição candidata j que apresenta o menor número de conflitos entre às possíveis posições do ponto i .

A estratégia tem como objetivo rotular os pontos de forma sequencial, de 1 a m. Em cada ponto, a heurística avalia todas as posições candidatas disponíveis e seleciona a posição que apresenta o menor número de conflitos.

Por exemplo, considere um ponto i que possui quatro posições candidatas ($p = 4$) e faz parte de uma instância com $m = 25$. A heurística gulosa desenvolvida irá percorrer as quatro posições de i, que por exemplo possuem número de conflitos iguais a 12, 9, 16 e 29, respectivamente. Nesse caso, a posição 2, que possui o menor número de conflitos, será selecionada para ser rotulada. Esse processo é repetido para garantir que todos os m pontos sejam rotulados, cada rótulo será atribuído a uma posição candidata de um ponto, porém, no cálculo final da função objetivo f, apenas os rótulos que não apresentarem sobreposição serão considerados.

3.2.2 Pertubação

A metodologia de perturbação utilizada neste estudo visa introduzir variações controladas na solução atual, a fim de promover diversidade e evitar que o algoritmo fique preso em ótimos locais. No procedimento de pertubação são realizadas γ iterações. O valor de γ é definido com base no valor de m. Em cada iteração um método de geração de vizinhança é selecionado aleatoriamente.

Neste trabalho foram implementados três diferentes métodos de geração de vizinhança, com o intuito de explorar o espaço de busca de maneira mais eficaz e aumentar as chances de encontrar soluções boas.

Essa abordagem de perturbação assegura que a solução não se estabilize prematuramente em ótimos locais. As variações controladas permitem que o algoritmo escape de máximos locais, melhorando sua capacidade de explorar o espaço de busca e aumentando a qualidade da solução final.

A seguir são apresentados os três métodos de geração de vizinhança implementados neste trabalho.

3.2.2.1 Geração de Vizinhança com Troca de Uma Posição Aleatória

O processo de geração de vizinhança recebe uma solução como entrada e realiza a modificação da seguinte forma: seleciona-se aleatoriamente um ponto i que, na solução original,

está rotulado na posição j. Em seguida esse ponto i é rotulado em uma nova posição j', que está dentro do intervalo de posições possíveis do ponto i, com a condição de que $j \neq j'$. Esse processo assegura que a nova posição j' é escolhida de forma aleatória, promovendo a diversificação da solução.

3.2.2.2 Geração de Vizinhança com Troca de Duas Posições Aleatórias

A estratégia descrita aqui para a geração de vizinhança compartilha semelhanças com o método abordado na seção 3.2.2.1, com algumas variações na aplicação. O procedimento começa com a seleção aleatória de um ponto i_1. Em seguida, o algoritmo escolhe aleatoriamente um segundo ponto i_2, garantindo que $i_1 \neq i_2$.

Após a seleção, o algoritmo avalia individualmente os dois pontos obtidos. Para cada ponto, a posição atual j é substituída por uma nova posição candidata j' ($j \neq j'$). Essa nova posição é escolhida aleatoriamente entre as posições candidatas do ponto. Essa atualização visa explorar diferentes configurações e potencialmente melhorar a solução existente.

3.2.2.3 Geração de Vizinhança com Modificação de Múltiplos Pontos

A metodologia utilizada para gerar a vizinhança nesse método inicia-se com a entrada de uma solução e um número de pontos ω.

O algoritmo realiza ω iterações, selecionando um ponto i em cada iteração. Para o ponto selecionado é escolhido uma nova posição candidata j' que substitui a posição original j, garantindo-se que $j \neq j'$.

Esse procedimento, diferentemente dos anteriores, promove uma maior diversificação da solução ao explorar diferentes configurações possíveis para múltiplos pontos simultaneamente, potencialmente melhorando a qualidade da solução final.

3.2.3 Busca local

A estratégia de busca local adotada neste trabalho fundamenta-se na aplicação da metaheurística conhecida como *Simulated Annealing* (SA). Esta abordagem representa um método sofisticado que busca otimizar soluções em espaços de busca complexos, e apresenta excelentes resultados obtidos em diversos trabalhos da literatura.

3.2.3.1 *Simulated Annealing*

O método *Simulated Annealing* (SA) foi proposto no estudo realizado por Kirkpatrick, Gelatt e Vecchi (1983) e é inspirado em um técnica da metalurgia conhecido como recozimento. Nesse processo, um metal é derretido e depois resfriado lentamente até a

solidificação. Em outras palavras, o metal é resfriado de forma controlada, para que sua estrutura se mantenha organizada e equilibrada (KAMPKE et al., 2022).

O SA requer a especificação de cinco parâmetros fundamentais para a sua execução, cada qual desempenhando um papel determinante no comportamento e eficácia do algoritmo. A seguir, são apresentados os elementos essenciais que compõem os parâmetros de entrada:

- Solução inicial ($S_{inicial}$).
- Temperatura inicial ($T_{inicial}$).
- Temperatura final (T_{final}).
- Taxa de resfriamento (β).
- Número de soluções vizinhas (N_v).

De acordo com o pseudocódigo do SA, que é apresentando no Algoritimo 2, uma iteração é iniciada com o valor $T_{inicial}$, o qual é progressivamente reduzido, seguindo a taxa estabelecida por β, até atingir o valor T_{final}. Em cada iteração, correspondente a um determinado valor de T, são geradas N_v soluções vizinhas. Subsequentemente, uma validação é realizada para atualizar a solução atual, caso a solução vizinha gerada seja superior à solução atual. A probabilidade $e^{-\Delta f/T}$ é empregada caso a solução vizinha não satisfaça os critérios de aceitação previamente estabelecidos. É relevante observar que Δf representa a diferença entre os valores de f das soluções vizinha e atual, enquanto T denota a temperatura atual no contexto da iteração em questão. À medida que a diferença Δf aumenta e a temperatura T diminui, a probabilidade de aceitar uma solução vizinha também diminui.

Inicialmente, em fases de temperatura elevada, o algoritmo demonstra uma propensão para aceitar uma ampla gama de soluções, promovendo uma exploração abrangente do espaço de busca. No entanto, à medida que a temperatura diminui, acontece uma transição para uma abordagem mais seletiva, reduzindo a aceitação de soluções de qualidade inferior. Esse decréscimo na aceitação de soluções menos favoráveis intensifica a busca em uma região específica, contribuindo para a convergência do algoritmo em direção a soluções melhores.

A geração de uma solução vizinha, abordada na linha 7 do Algoritmo 2, é realizada por meio de uma estratégia que seleciona aleatoriamente um dos métodos de geração de vizinhança, apresentados nas subseções 3.2.2.1 a 3.2.2.3. Essa abordagem assegura uma maior diversificação das soluções geradas, permitindo explorar diferentes regiões do espaço de busca.

Algoritmo 2: Algoritmo *SA* Adaptado de Rocha (2013)

Entrada: Solução $S_{Inicial}$, $T_{inicial}$, T_{final}, β, N_v

Saída: Solução S_{Melhor}

```
1  início
2      T ← T_inicial;
3      S_Melhor ← S_Inicial;
4      S_Atual ← S_Inicial;
5      enquanto T > T_final faça
6          para i ← 1 até N_v faça
7              S_Vizinho ← GeraVizinho(S_Atual);
8              Δf ← f(S_Vizinho) − f(S_Atual);
9              se Δf > 0 então
10                 S_Atual ← S_Vizinho;
11                 se f(S_Vizinho) > f(S_Melhor) então
12                     S_Melhor ← S_Vizinho;
13                 fim se
14             senão
15                 Gera um número aleatório ρ ∈ [0, 1];
16                 se ρ < e^(−Δf/T) então
17                     S_Atual ← S_Vizinho;
18                 fim se
19             fim se
20         fim para
21         T ← T ∗ β;
22     fim enquanto
23 fim
```

4 Resultados Computacionais

Nesse capítulo são expostos, examinados e discutidos os resultados derivados dos experimentos computacionais realizados. Primeiramente, na subseção 4.1, o conjunto de instâncias é descrito em detalhe. Em seguida, na subseção 4.2, é abordado o ambiente em que os experimentos foram conduzidos, além da definição dos parâmetros do método proposto. Na subseção 4.3, os resultados são apresentados, analisados e discutidos de forma detalhada.

4.1 Instâncias

Para a realização deste trabalho, cujo objetivo é abordar a resolução do problema PRCP, foram selecionados e utilizados os conjuntos de instâncias propostos por Lorena (2024). Esses conjuntos de instâncias estão divididos em duas partes. A primeira parte apresenta instâncias com número de pontos igual a 25, 100, 250, 500, 750 e 1000 pontos, todas compostas por quatro posições candidatas, seguindo o padrão estabelecido por Christensen, Marks e Shieber (1995). Para as instâncias com 25 pontos, foram geradas oito variações distintas, enquanto que as instâncias com 100, 250, 500, 750 e 1000 pontos, possuem 25 variações diferentes cada. A segunda parte do conjunto é composta por 40 instâncias, cada uma delas contendo 13206 pontos, sendo que 20 delas possuem quatro posições candidatas e as outras 20, oito posições candidatas, conforme apresentado na Tabela 1.

Tabela 1 – Instâncias para o problema cartográfico de pontos.

Instâncias	m	p	Nº de instâncias
PRCP-0025-8	25	4	8
PRCP-0100-25	100	4	25
PRCP-0250-25	250	4	25
PRCP-0500-25	500	4	25
PRCP-0750-25	750	4	25
PRCP-1000-25	1000	4	25
PRCP-13206-20-4	13206	4	20
PRCP-13206-20-8	13206	8	20

4.2 Experimentos Computacionais

Após a implementação do algoritmo híbrido proposto neste trabalho, foram realizados experimentos computacionais para obtenção dos resultados. Inicialmente, antes da coleta dos resultados, foi efetuada a calibração dos parâmetros. Para isso, foi selecionada aleatoriamente uma instância de cada tamanho do conjunto de instâncias, formando um

subconjunto com oito instâncias distintas. Cada instância usada no processo de calibração foi executada uma vez, durante cinco minutos, para cada combinação de valores apresentados na segunda coluna da Tabela 2. A terceira coluna da tabela 2 apresenta os valores escolhidos, após a calibração, para cada um dos parâmetros.

Tabela 2 – Parâmetros para o problema cartográfico de pontos.

Parâmetro	Conjuntos	Valor
$T_{inicial}$	{100, 300, 500 }	500
T_{final}	{0.001, 0.01, 0.1}	0.1
β	{0.995, 0.975, 0.95}	0.95
N_v	{100, 300, 500 }	100

Além da calibração dos parâmetros, o valor de γ foi definido como 75% do valor de m. Essa escolha, conforme discutido na Seção 3.2.2, visa garantir um equilíbrio entre a diversidade das soluções e a eficiência do processo de busca. Para a geração de vizinhança com modificação de múltiplos pontos, apresentado na subseção 3.2.2.3, o valor de ω foi estabelecido como 50% do valor de m.

O algoritmo implementado foi desenvolvido na linguagem de programação C++ e compilado com GNU g++ *compiler*. Os experimentos computacionais foram realizados em um ambiente com microprocessador Intel Core i3-7020U de 2.3 GHz, 4 GB de memória RAM, utilizando o sistema operacional Ubuntu 24.04 LTS.

4.3 Resultados Obtidos

Os resultados obtidos após a calibração dos parâmetros demonstraram a eficiência do algoritmo na busca por soluções de qualidade. A estratégia adotada consistiu em executar o algoritmo cinco vezes para cada instância, sendo que cada execução tem um tempo de cinco minutos, correspondente ao *loop* de iteração do ILS. Em outras palavras, o valor de ILS_{max} do Algoritmo 1 foi substituído pelo tempo de execução. O conjunto de instâncias com 25, 100 e 250 pontos apresentou soluções equivalentes às encontradas pelo método CS proposto por Rabello et al. (2013), que se destacou por gerar boas soluções quando comparadas às da literatura. Para os conjuntos com 500, 750 e 1000 pontos, foram encontrados valores próximos das soluções ótimas, conforme apresentado na Tabela 3.

Tabela 3 – Percentual (%) de Pontos Rotulados Sem Conflitos.

Métodos de solução	m		
	500	**750**	**1000**
ILS + SA	**98,04**	**93,45**	**84,53**
CS (RABELLO et al., 2013)	99,68	97,96	93,88

Para as instâncias compostas por 500 pontos, o algoritmo proposto apresentou uma variação de 1,65% em relação aos resultados de Rabello et al. (2013). Em contraste, no conjunto de instâncias com 1000 pontos, a variação foi de 9,9%. Essa comparação de resultados precisa levar em consideração dois fatores: o menor número de repetições do algoritmo proposto neste trabalho e as diferenças nos ambientes computacionais em que os experimentos foram conduzidos. Esses fatores reforçam a eficiência do algoritmo proposto, demonstrando seu potencial na busca de soluções ótimas.

A Tabela 4 apresenta os resultados obtidos para o conjunto de instâncias com 13206 pontos, considerando quatro e oito posições candidatas. Na Tabela 4, f_{best} refere-se à melhor solução encontrada dentre as cinco execuções do algoritmo, enquanto f_{avg} representa a média dos resultados obtidos. Além disso, o Coeficiente de Variação (CV) foi calculado para mensurar a dispersão dos resultados.

Tabela 4 – Resultados Obtidos para as instâncias de 13206 pontos.

Instâncias	PRCP-13206-020-4			PRCP-13206-020-8		
	f_{best}	f_{avg}	CV(%)	f_{best}	f_{avg}	CV(%)
1	9043	9038,2	0,05	9643	9642	0,01
2	8416	8411,8	0,04	8650	8645,6	0,05
3	8195	8193,4	0.01	8657	8654,4	0,03
4	7868	7865,2	0,03	8383	8381,4	0,01
5	7606	7601	0,06	8280	8278,6	0,01
6	6669	6661	0,11	7045	7037,4	0,10
7	6057	6055	0,03	6485	6481,2	0,05
8	6474	6469	0,07	6933	6927,8	0,07
9	6360	6357,4	0,04	6464	6462,4	0,02
10	6014	6012,4	0,02	6375	6372,8	0,03
11	6060	6051,4	0,14	5915	5913,4	0,02
12	5326	5324,6	0,02	5650	5646,6	0,06
13	6226	6221	0,08	6884	6879,6	0,06
14	5621	5614,2	0,12	5664	5661,6	0,04
15	5180	5176,6	0,06	5664	5560,8	1,8
16	4689	4686,8	0,04	5041	5035,6	0,10
17	5537	65530,2	0,12	6142	6138,2	0,06
18	4914	4905	0,18	4942	4940	0,04
19	4486	4482,2	0,08	4860	4855,6	0,09
20	4039	4036	0,07	4352	4344,4	0,17
Média	6239	6234,62	0,06	6601,45	6219,91	0,14

Com base nos resultados da Tabela 4, observa-se que tanto as instâncias com quatro posições, quanto as com oito posições, apresentaram baixa variação. As maiores variações ocorreram na instância 18 para o caso de quatro posições, com CV de 0,18%, e na instância

15 para o caso de oito posições, com CV de 1,18%. Embora o CV não tenha sido nulo, essas variações mínimas evidenciam a estabilidade e a robustez do algoritmo.

Os percentuais de rótulos livres para o conjunto de instâncias com 13206 pontos estão apresentados na Tabela 5.

Tabela 5 – Percentual (%) de 13206 Pontos Rotulados Sem Conflitos

Métodos de solução	p	
	4	**8**
ILS + SA	**30,5**	**49,9**
CS (RABELLO et al., 2013)	79,77	85,45

De acordo com os resultados, o algoritmo proposto apresentou uma variação de 61% para o caso com quatro posições e de 41% para o caso com oito posições, em comparação com os resultados de Rabello et al. (2013), que em ambos os casos foram inferiores. Uma possível causa para esse desempenho do algoritmo proposto é o fato que o tempo de execução foi menor, uma vez que o algoritmo de Rabello et al. (2013) teve um tempo de execução média de 21,46 minutos para a instâncias com quatro posições e 26,67 minutos para a instância com oito posições. Além disso, esse conjunto de instância possui um grande quantidade de pontos, o que dificulta encontrar soluções de qualidade.

Embora os resultados obtidos neste trabalho sejam menores em comparação aos resultados apresentados na literatura, é importante destacar que o algoritmo desenvolvido demonstrou potencial ao produzir soluções de qualidade com baixo tempo computacional e com um número menor de repetições. Essa eficiência no tempo de processamento pode ser uma vantagem significativa em aplicações práticas do PRCP, em que a rapidez é crucial.

5 Conclusão

Neste trabalho, foi abordado o Problema de Rotulação Cartográfica de Pontos (PRCP), propondo-se um método de solução baseado na meta-heurística ILS, combinada com o uso do *Simulated Annealing* na etapa de busca local. Além de apresentar uma fácil implementação, a combinação do ILS com o SA gerou resultados consistentes e eficientes em tempos de execução consideravelmente baixos.

Os resultados obtidos demonstraram a eficácia dessa abordagem para diferentes instâncias do problema, especialmente para aquelas com um número menor e médio de pontos. No entanto, para instâncias maiores, o método não conseguiu alcançar os melhores resultados já registrados na literatura. Isso sugere que o nível de exploração durante a busca por soluções pode ter sido insuficiente, uma vez que o tempo de processamento foi menor, assim como o número de repetições. Como resultado, o algoritmo pode ter convergido rapidamente para soluções subótimas, sem realizar uma exploração mais ampla das possíveis combinaçõe

Portanto, as diferenças nos resultados indicam que há espaço para melhorias. Ajustes finos nos parâmetros do algoritmo, bem como a implementação de diferentes estratégias de busca local, podem levar a melhorias significativas no desempenho. Trabalhos futuros podem se concentrar no refinamento do algoritmo, explorando novas estruturas de vizinhança, variando o número de repetições, e ajustando a parametrização com o objetivo de alcançar, ou até mesmo superar os resultados encontrados na literatura.

Referências

ALVES, L. F. *O uso de sistemas de organização e rotulação por arquitetos de informação web: estudo de caso*. Monografia (Trabalho de conclusão de graduação) — Universidade Federal do Rio Grande do Sul, 2011.

CARVALHO, A. S. et al. Simulated annealing aplicado ao problema de programação de horários do cca-ufes. In: *Anais do Simpósio de Pesquisa Operacional e Logística da Marinha*. Rio de Janeiro, RJ, Brasil: CASNAV, 2016. v. 2, n. 1, p. 341–352.

CHRISTENSEN, J.; MARKS, J.; SHIEBER, S. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics (TOG)*, v. 14, n. 3, p. 203–232, 1995.

CRAVO, G. L.; RIBEIRO, G. M.; LORENA, L. A. N. Heurística gulosa para o problema da rotulação cartográfica de pontos. *Revista Educação Tecnológica,,(1/2)*, v. 2, n. 1, p. 1–11, 2006.

CRAVO, G. L.; RIBEIRO, G. M.; LORENA, L. A. N. A greedy randomized adaptive search procedure for the point-feature cartographic label placement. *Computers Geosciences*, v. 34, n. 4, p. 373–386, 2008.

KAMPKE, E. H. et al. Meta-heurística simulated annealing aplicada ao problema de transporte de máquinas em florestas de produção. SOBRAPO, Rio de Janeiro, RJ, Brasil, p. 1–12, 2022.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983.

LORENA, L. A. *Instâncias para Testes de Algoritmos*. 2024. Disponível em: <`http://www.lac.inpe.br/~lorena/instancias.html`>.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. (Ed.). *Handbook of metaheuristics*. Boston, MA, USA: Springer, 2003. p. 320–353.

MAURI, G. R. Novas abordagens para representação e obtenção de limitantes e soluções para alguns problemas de otimização combinatória. *São José dos Campos–SP*, 2008.

MAURI, G. R.; RIBEIRO, G. M.; LORENA, L. A. A new mathematical model and a lagrangean decomposition for the point-feature cartographic label placement problem. *Computers & Operations Research*, Elsevier, v. 37, n. 12, p. 2164–2172, 2010.

MONTEIRO, R. C. et al. Algoritmo híbrido iterated local search e simulated annealing para o problema de tabela-horário de universidades. *Anais do XLIX Simpósio Brasileiro de Pesquisa Operacional*, p. 1867–1878, 2017.

OLIVEIRA, C.; URRUTIA, S.; NORONHA, T. F. Heurística de backtracking para o problema da rotulação cartográfica de pontos. *Anais do XLII SBPO-Simpósio Brasileiro de Pesquisa Operacional*, 2010.

OLIVEIRA, C. d. Curso de cartografia moderna. *Rio de janeiro: IBGE*, v. 1, p. 991, 1988.

OLIVEIRA, C. de. Heurísticas para o problema de rotulação cartográfica de pontos e coloração de vértices com pesos. Universidade Federal de Minas Gerais, 2012.

RABELLO, R. L. et al. Meta-heurística clustering search aplicada ao problema de rotulação cartográfica de pontos. *Anais do XLV SBPO-Simpósio Brasileiro de Pesquisa Operacional*, 2013.

RIBEIRO, G. M.; LORENA, L. A. N. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. *Computers & Operations Research*, Elsevier, v. 35, n. 7, p. 2129–2140, 2008.

ROCHA, W. S. *Algoritmo GRASP para o Problema de Tabela-horário de Universidades.* Dissertação (Mestrado em Informática) — Universidade Federal do Espírito Santo, 2013.

SILVA, C. A.; SOUZA, S. R. de. Uma aplicação da meta-heurística híbrida simulated annealing-iterated local search ao problema de fluxo multiproduto sob o espaço capacitado. *Trends in Computational and Applied Mathematics*, v. 9, n. 1, p. 165–174, 2008.

SUBRAMANIAN, A. et al. Um algoritmo heurístico baseado em iterated local search para problemas de roteamento de veículos. *Meta-Heurísticas em Pesquisa Operacional. Ed. Omnipax, Curitiba, Brazil*, 2013.

VIEIRA, N. H. et al. Uma heurística ils para a resolução do problema de roteamento de veículos com frota heterogênea. *Revista Latino-Americana de Inovação e Engenharia de Produção*, v. 10, n. 17, p. 63–77, 2022.

WOLFF, A. *Automated label placement in theory and practice.* Tese (Doutorado), 2002.

YOELI, P. The logic of automated map lettering. *The Cartographic Journal*, Taylor & Francis, v. 9, n. 2, p. 99–108, 1972.

ZORASTER, S. The solution of large 0–1 integer programming problems encountered in automated cartography. *Operations Research*, INFORMS, v. 38, n. 5, p. 752–759, 1990.