UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO

LUCAS MOREIRA XAVIER

ADAPTIVE LARGE NEIGHBORHOOD SEARCH PARA O PROBLEMA DE LOCALIZAÇÃO DE SENSORES EM REDE COM PISTAS MONITORADAS

ALEGRE - ES
2022

LUCAS MOREIRA XAVIER

ADAPTIVE LARGE NEIGHBORHOOD SEARCH PARA O PROBLEMA DE LOCALIZAÇÃO DE SENSORES EM REDE COM PISTAS MONITORADAS

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.

Orientador: Prof. Dr. Geraldo Regis Mauri.

ALEGRE - ES

2022

LUCAS MOREIRA XAVIER

ADAPTIVE LARGE NEIGHBORHOOD SEARCH PARA O PROBLEMA DE LOCALIZAÇÃO DE SENSORES EM REDE COM PISTAS MONITORADAS

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 18 de agosto de 2022.

COMISSÃO EXAMINADORA

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

Prof. Dr. Antonio Almeida de Barros Junior
Universidade Federal do Espírito Santo

Prof. Dr. Dayan de Castro Bissoli
Universidade Federal do Espírito Santo

RESUMO

O processo de extrair informações de tráfego em rodovias se dá através de sensores de contagem de tráfego, sendo que cada um desses contadores cobre uma certa área. Se esses sensores forem situados em lugares que não sejam bem aproveitados, uma grande quantidade de sensores pode ser adquirida desnecessariamente. Dessa forma, fazendo com que seja gasto mais do que o necessário para cobrir uma certa área. Uma vez que o valor de aquisição desses sensores não é algo muito acessível, esse gasto pode muitas das vezes se tornar um enorme prejuízo. Encontrar os pontos ideais onde esses sensores devem ser instalados é conhecido como Problema de Localização de Sensores em Rede (PLSR). Este trabalho consiste em utilizar a meta-heurística *Adaptive Large Neighborhood Search* (ALNS) para resolução de problemas com base em dados reais da malha rodoviária brasileira, considerando restrições relacionadas ao número máximo de dispositivos e ao número máximo de pistas monitoradas. O ALNS apresentou bons resultados em outros problemas de otimização, e ainda não foi explorado como alternativa de solução para o problema em questão. Para avaliar a performance do ALNS aplicado ao PLSR, os resultados obtidos através dos testes computacionais são comparados com resultados de outros trabalhos encontrados na literatura.

Palavras-chave: *Adaptive Large Neighborhood Search*; Meta-heurística; Problema de localização de sensores de rede; Tráfego em rodovias.

ABSTRACT

The process of extracting traffic information on highways takes place through traffic counting sensors, each of which covers a certain area. If these sensors are located in places that are not well utilized, a large number of sensors can be acquired unnecessarily. That way, making it spend more than necessary to cover a certain area. Since the acquisition value of these sensors is not something very affordable, this expense can often become a huge loss. Finding the ideal points where these sensors should be installed is known as the Network Sensor Location Problem (NSLP). This paper consists of using the Adaptive Large Neighborhood Search (ALNS) meta-heuristic to solve problems based on real data from the Brazilian road network, considering restrictions related to the maximum number of devices and the maximum number of monitored lanes. ALNS showed good results in other optimization problems, and has not yet been explored as an alternative solution for the problem in question. To evaluate the performance of ALNS applied to PLSR, the results obtained via computational tests are compared with results from other works found in the literature.

Keywords: Adaptive Large Neighborhood Search; Meta-heuristic; Network sensor location problem; Highway traffic.

LISTA DE FIGURAS

Figura 1 - Sensores de contagem de tráfego posicionados para ser realizada a estimativa dos fluxos de tráfego 7
Figura 2 - Pseudocódigo da heurística construtiva. 18
Figura 3 - Pseudocódigo do ALNS utilizado. 19
Figura 4 – Pseudocódigo da heurística *Shaw Removal* implementada no ALNS. ... 21
Figura 5 – Pseudocódigo da heurística *Worst Removal* implementada no ALNS. ... 22

LISTA DE TABELAS

Tabela 1 - Resultados da busca na plataforma Science Direct utilizando as palavras-chaves em inglês 12
Tabela 2 - Resultados da busca na plataforma Science Direct utilizando as palavras-chaves em português 12
Tabela 3 - Resultados da busca na plataforma Google Scholar utilizando as palavras-chaves em inglês 13
Tabela 4 - Resultados da busca na plataforma Google Scholar utilizando as palavras-chaves em português 14
Tabela 5 - Intervalo de valores possíveis para cada parâmetro durante a calibração 26
Tabela 6 - Valores dos parâmetros do ALNS após a calibração 26
Tabela 7 - Dados de cada instância 27
Tabela 8 - Resumo dos resultados computacionais 29
Tabela 9 - Comparação dos resultados com o algoritmo Branch and Cut 30
Tabela 10 - Comparação dos resultados com o algoritmo Clustering Search..... 32

SUMÁRIO

1 INTRODUÇÃO 6
1.1 O PROBLEMA E SUA IMPORTÂNCIA 6
1.2 OBJETIVOS 10
1.2.1 Objetivo Geral 10
1.2.2 Objetivos Específicos 10
1.3 ESTRUTURA DO TRABALHO 10
2 REVISÃO DE LITERATURA 11
2.1 TRABALHOS CORRELATOS AO PLSR 14
2.2 CONSIDERAÇÕES 16
3 METODOLOGIA 17
3.1 FUNCIONAMENTO DO ALNS 17
3.2 HEURÍSTICAS DE DESTRUIÇÃO 21
3.2.1 *Shaw Removal* 21
3.2.2 *Random Removal* 22
3.2.3 *Worst Removal* 22
3.3 HEURÍSTICAS DE RECONSTRUÇÃO 23
3.3.1 *Basic Greedy Repair* 23
3.3.2 *Random Repair* 23
3.3.3 *Worst Element Repair* 23
4 TESTES COMPUTACIONAIS 25
4.1 Calibração dos parâmetros 25
5 RESULTADOS E DISCUSSÃO 27
5.1 Resultados computacionais 28
5.2 Comparação com a literatura 29
6 CONCLUSÃO 33
REFERÊNCIAS BIBLIOGRÁFICAS 34

1 INTRODUÇÃO

Conhecer o comportamento do tráfego em rodovias federais é um pré-requisito para que o Departamento Nacional de Infraestrutura de Transportes (DNIT) realize algumas atividades fim na administração da malha rodoviária (BRASIL, 2020). Além disso, segundo a Confederação Nacional do Transporte (2016), esses números são fundamentais para o planejamento de ampliação e manutenção de rodovias, podendo também auxiliar em intervenções prioritárias na malha e identificação dos principais eixos rodoviários.

Segundo a BBC (2018), o Brasil é o país com maior concentração de transporte rodoviário de cargas e passageiros, sendo que 58% do transporte no país é feito por rodovias, detendo também, uma malha rodoviária utilizada para o escoamento de 75% da produção no país.

Conforme Abreu *et al.* (2020), assim como os outros departamentos de transporte brasileiros, o DNIT possui um baixo orçamento para instalar esses sensores, além de possuir um quadro de funcionários limitado. Portanto, é necessário localizar pontos estratégicos para conseguir um número reduzido de sensores na malha rodoviária.

Com o intuito de definir a localização dos sensores, neste trabalho é proposta a aplicação do meta-heurística *Adaptive Large Neighborhood Search* – ALNS, que ainda não foi utilizada para resolver o Problema de Localização de Sensores em Rede – PLSR, apresentado em detalhes a seguir.

Além disso a meta-heurística ALNS proposta por Pisinger e Ropke (2007) apresentou bons resultados em outros problemas de otimização na literatura, como por exemplo, *Cumulative Capacitated Vehicle Routing Problem* (RIBEIRO; LAPORTE, 2012), *Pollution-Routing Problem* (DEMIR; BEKTAŞ; LAPORTE, 2012), *Discrete and Continuous Berth Allocation Problem* (MAURI *et al.*, 2016) e *Electric Vehicle Scheduling Problem* (WEN *et al.*, 2016).

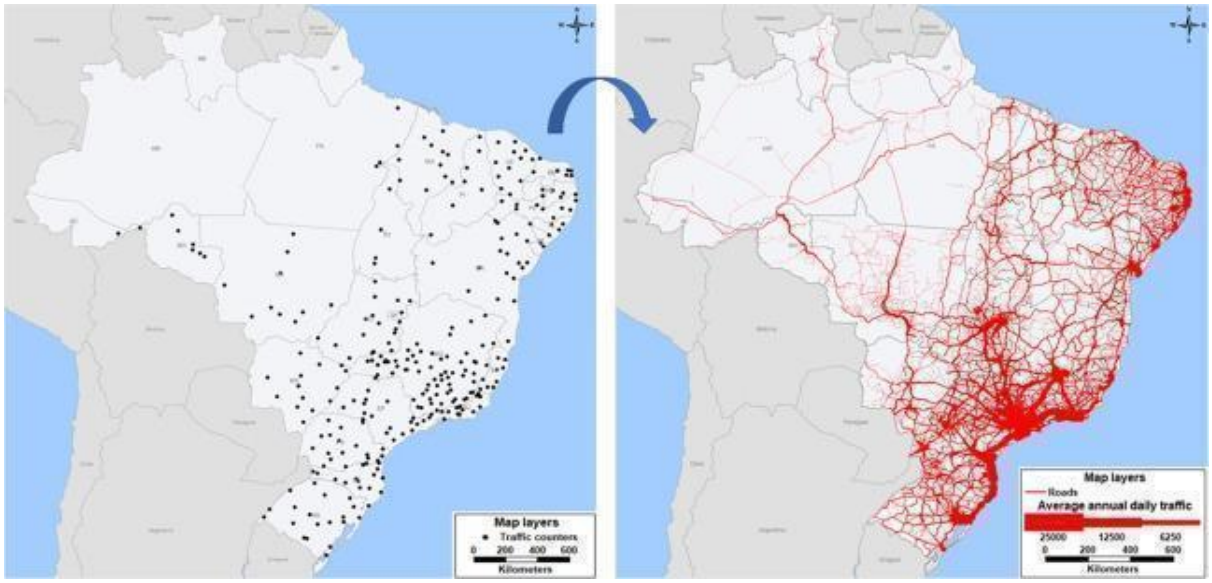
1.1 O PROBLEMA E SUA IMPORTÂNCIA

Fazer o monitoramento de uma rodovia não somente diz respeito à quantidade de veículos, mas também ao peso, direção, etc. Esses dados são interessantes pois, com eles é possível determinar por exemplo, a degradação dos pavimentos ou o nível de ruído atingido (ABREU *et al.*, 2020). Além disso, ainda conforme Abreu *et al.* (2020), após esses dados serem processados e tratados, é possível compreender os

fundamentos do tráfego rodoviário de uma determinada região. O que se mostra indispensável no que diz respeito a efetuar e manter uma política de tráfego eficiente, garantindo assim, a segurança e as necessidades dos usuários das vias daquela área.

Segundo Li *et al.* (2015), baseando-se no volume de tráfego é possível também desenvolver matrizes de origem e destino (O–D), que por sua vez, representam a distribuição das viagens entre as zonas de tráfego de uma certa área de interesse. De acordo com Fisk (1988), tais matrizes são uma importante fonte de informação para uma variedade de estudos de transporte, tais como realizar uma previsão de demanda futura de viagens e gestão de transporte e controle. A Figura 1 apresenta uma solução onde alguns sensores são colocados para estimar os fluxos de tráfego para a malha rodoviária brasileira.

Figura 1 - Sensores de contagem de tráfego posicionados para ser realizada a estimativa dos fluxos de tráfego.



Fonte: Abreu *et al.* (2020).

Entretanto, quando se trata de uma rede de grande dimensão, não é economicamente viável posicionar esses sensores em absolutamente todos os segmentos. Isso se dá pelo fato de que tanto a manutenção quanto a instalação desses equipamentos são custosas, principalmente por que existe a necessidade de uma mão de obra especializada (FRANCA *et al.*, 2020).

Além disso, devem ser consideradas restrições orçamentárias por parte das agências de controle e gestão de tráfego. Consequentemente, a opção encontrada para realizar o monitoramento das condições de tráfego se resume a implantar os sensores em pontos estratégicos da rede de transportes, tanto em condições normais quanto em dias atípicos (GONZÁLEZ *et al.*, 2017).

Nesse cenário, é importante estudar a localização onde esses equipamentos serão instalados, para que assim, os recursos disponíveis sejam bem aproveitados. Segundo Abreu *et al.* (2017), esse problema relacionado ao posicionamento desses equipamentos, considerando as restrições citadas anteriormente, é conhecido como Problema de Localização de Sensores em Rede (PLSR).

Uma revisão sobre o PLSR foi desenvolvida por Gentili e Mirchandani (2012) com o objetivo de dividir e classificar o problema em dois tipos principais: o problema de observação do fluxo e o problema de estimativa do fluxo. Onde, de acordo com Abreu *et al.* (2020), o problema de observação do fluxo trabalha com o intuito de encontrar a localização ideal para observar os fluxos parcialmente ou completamente, enquanto o problema de estimativa de fluxo identifica as posições ótimas para melhorar a qualidade das estimativas de fluxo.

De 60% a 70% dos segmentos em uma rede deveriam ser observados para que seja possível atingir uma observação completa do sistema (HU *et al.* 2009; CASTILLO *et al.* 2013). Isso é o suficiente para inviabilizar economicamente o problema de observação do fluxo em casos de uma rede cotidiana, o que resulta no problema de estimativa do fluxo. Segundo Gentili e Mirchandani (2012), esse modelo é condizente a duas situações: quando se busca minimizar a quantidade de sensores, considerando que eles são ilimitados e, no segundo caso, se busca maximizar uma regra de cobertura, considerando uma restrição orçamentária, o que acarreta em uma restrição de sensores.

Portanto, fornecer ótimas e/ou boas soluções para o PLSR é fundamental para que seja economicamente viável realizar esse monitoramento das rodovias. Além disso, quanto melhor forem essas soluções, maior será a economia para fazer o monitoramento dessas rodovias.

Neste trabalho, o problema foi modelado de acordo com a formulação apresentada por Abreu *et al.* (2020), que localiza sensores para maximizar o número de pares OD interceptados por eles, respeitando restrições orçamentárias

relacionadas ao número máximo de sensores disponíveis e ao número máximo de faixas de tráfego a serem monitoradas.

A Equação (1) representa a função objetivo do problema que visa maximizar o número de pares OD cobertos. Os itens restantes apresentam as restrições do modelo, onde a restrição (2) garante a cobertura do subconjunto de pares OD, as restrições (3) e (4) são relacionadas ao orçamento disponível, e o domínio das variáveis de decisão são definidos em (5) e (6).

$$Maximizar\ Z = \sum_{w \in W} y_w \tag{1}$$

Sujeito à:

$$\sum_{a \in E} p_{ra}^{w} x_a \geq y_w \ \forall r \in R_w, w \in W \tag{2}$$

$$\sum_{a \in E} c_a x_a \leq \alpha \tag{3}$$

$$\sum_{a \in E} f_a x_a \leq \beta \tag{4}$$

$$x_a \in \{0,1\} \ \forall a \in E \tag{5}$$

$$y_w \in \{0,1\} \ \forall w \in W \tag{6}$$

Considerando uma rede de transporte $G = (V, E)$, onde V representa os vértices e E as arestas, onde W seria o conjunto de todos os pares OD e R^w é o conjunto de caminhos que conectam o par OD w, onde $R = \cup_{w \in W} R^w$. Tem-se que, p_{ra}^{w} é igual a 1 se o segmento $a \in E$ no caminho $r \in R$ entre o par OD $w \in W$, e igual a 0 caso contrário.

Além disso, α é o número máximo de sensores a serem localizados e β o número máximo de faixas de tráfego a serem monitoradas, onde f_a representa o número de faixas no segmento $a \in E$ e c_a o número de contadores de tráfego que precisam ser instalados no segmento $a \in E$. Logo, se um determinado segmento $a_1 \in E$ tem duas faixas em cada sentido e cada um deles precisa de um contador de tráfego, então $f_{a1} = 4$ e $c_{a1} = 2$.

Com relação às variáveis de decisão, $x_a \in \{0,1\}, a \in E$ são variáveis de decisão binária. Caso um sensor seja colocado em um segmento a, $x_a = 1$, caso contrário, $x_a = 0$. Além disso, y_w também são variáveis de decisão binárias, onde, $y_w = 1$ indica que todos os caminhos que conectam o par $w \in W$ possuem ao menos 1 sensor, ou seja, este par é coberto, caso contrário $y_w = 0$.

1.2 OBJETIVOS

A seguir são apresentados os objetivos gerais e específicos relacionados a este trabalho.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho visa aplicar a meta-heurística *Adaptive Large Neighborhood Search* (ALNS) para resolver o Problema de Localização de Sensores em Rede – PLSR.

1.2.2 Objetivos Específicos

a) Resolver o PLSR a partir do ALNS;
b) Realizar experimentos computacionais utilizando as instâncias propostas por Abreu *et al.* (2020);
c) Comparar os resultados apresentados pelo ALNS com os já existentes na literatura.

1.3 ESTRUTURA DO TRABALHO

No Capítulo 2 uma revisão da literatura é apresentada com uma descrição dos trabalhos apresentados até então. Logo em seguida, no Capítulo 3, é apresentada a metodologia do trabalho. São descritos os testes computacionais no Capítulo 4, e os resultados obtidos no Capítulo 5. A conclusão é apresentada no Capítulo 6, seguida pelas referências utilizadas no desenvolvimento do trabalho.

2 REVISÃO DE LITERATURA

Foi realizada uma busca de trabalhos que sejam correlatos a este. Para tal, foram utilizadas as plataformas *Google Scholar*[1] e *Science Direct*[2]. As palavras-chave utilizadas foram: "*Sensor Location Problem*", "*Network Sensor Location Problem*", "*Network Sensor Location Metaheuristic*", "*Traffic Counter Location Problem*" e "*Network Traffic Sensor Location Problem*", em português: "Problema de localização do sensor de rede", "Metaheurística de localização do sensor de rede", "Problema de localização do sensor", "Problema de Localização de Contadores de Tráfego" e "Problema de Localização de Sensores de Tráfego em Rede".

Vale ressaltar que a busca foi realizada considerando apenas os resultados nos últimos 5 anos (2017 - 2021) e que para selecionar os trabalhos, foi feita uma análise a partir dos títulos e também da relevância do trabalho com relação à palavra-chave. Cada busca foi feita independente da outra, portanto o mesmo trabalho pode aparecer em mais de uma palavra-chave e plataforma. Após isso, foi feita a leitura dos trabalhos selecionados.

As Tabelas 1 e 2 apresentam os resultados obtidos na plataforma *Science Direct*. O número de resultados na Tabela 1 foi bem grande, porém a enorme maioria acabava não se referindo a área de otimização, nem mesmo se direcionava a área de contadores de tráfego como se pode observar. Na Tabela 2, pode-se observar que dos oito trabalhos selecionados, nenhum deles estava relacionado com o assunto tratado nesta pesquisa, apenas pela leitura dos títulos foi possível perceber que eles não estavam relacionados ao PLSR.

A Tabela 2 apresenta um número muito baixo de resultados, possivelmente devido à utilização de palavras-chave em português em um *site* onde os artigos são em grande maioria em inglês. No fim foram selecionados apenas três trabalhos, porém, dentre esses 3, todos eram repetidos, resultando então em apenas 1 trabalho.

---

[1] https://scholar.google.com.br/
[2] https://www.sciencedirect.com/

Tabela 1 - Resultados da busca na plataforma Science Direct utilizando as palavras-chaves em inglês.

Palavra-chave	**Resultados**	**Número de trabalhos selecionados**
Sensor Location Problem	75.818	1
Network Sensor Location Problem	41.697	1
Network Sensor Location Metaheuristic	911	0
Traffic Counter Location Problem	5.263	0
Network Traffic Sensor Location Problem	10.666	1
Total	134.355	3

Tabela 2 - Resultados da busca na plataforma Science Direct utilizando as palavras-chaves em português.

Palavra-chave	**Resultados**	**Número de trabalhos selecionados**
Problema de localização do sensor de rede	2	0
Metaheurística de localização do sensor de rede	0	0
Problema de localização do sensor	6	0
Problema de Localização de Contadores de Tráfego	0	0
Problema de Localização de Sensores de Tráfego em Rede	0	0
Total	8	0

As Tabelas 3 e 4 apresentam os resultados obtidos na plataforma *Google Scholar*. O *Google Scholar* limita a busca em mil resultados no máximo por pesquisa, portanto, independentemente do número de resultados, apenas os mil primeiros foram analisados devido a essa limitação da plataforma.

A Tabela 3 apresenta um número muito alto de resultados, porém apenas 2 trabalhos foram considerados relevantes. O primeiro trabalho já havia sido selecionado, portanto apenas 1 trabalho foi considerado relevante e correlato ao PLSR.

Tabela 3 - Resultados da busca na plataforma Google Scholar utilizando as palavras-chaves em inglês.

Palavra-chave	**Resultados**	**Número de trabalhos selecionados**
Sensor Location Problem	247.000	1
Network Sensor Location Problem	40.800	0
Network Sensor Location Metaheurístic	16.300	0
Traffic Counter Location Problem	20.800	1
Network Traffic Sensor Location Problem	17.100	0
Total	342.000	2

A Tabela 4 apresenta uma quantidade bem menor de resultados, porém, ainda é uma quantidade grande diante da limitação mencionada acima da plataforma. Essa foi a pesquisa com o maior número de trabalhos selecionados, porém, muitos se repetiram.

Na primeira pesquisa, 4 trabalhos foram selecionados, sendo que 1 deles é uma revisão da literatura e os demais ainda não haviam sido obtidos. Na segunda, apenas 1 trabalho foi desconsiderado pois já havia aparecido nas buscas anteriores.

Nas demais pesquisas, mais 1 trabalho foi selecionado, e o restante já havia aparecido nas buscas anteriores.

Tabela 4 - Resultados da busca na plataforma Google Scholar utilizando as palavras-chaves em português.

Palavra-chave	**Resultados**	**Número de trabalhos selecionados**
Problema de localização do sensor de rede	15.800	4
Metaheurística de localização do sensor de rede	185	4
Problema de localização do sensor	16.400	4
Problema de Localização de Contadores de Tráfego	1.460	8
Problema de Localização de Sensores de Tráfego em Rede	4.210	6
Total	38.055	26

Após a análise de todos os resultados, 10 trabalhos foram selecionados, sendo 1 deles uma revisão da literatura do PLSR. Os trabalhos selecionados são apresentados a seguir.

2.1 TRABALHOS CORRELATOS AO PLSR

González *et al.* (2017a) apresentaram um algoritmo *Branch-and-Cut* para um problema de localização de contagem de tráfego com a intenção de encontrar soluções ótimas, buscando as soluções com base nos Estados brasileiros com dados reais da malha rodoviária. No trabalho em questão, foram encontradas soluções ótimas para sete Estados e boas soluções para outros treze. Além disso, outro trabalho no mesmo ano foi apresentado por Gonzáles *et al.* (2017b), neste é

apresentado um algoritmo genético com chaves aleatórias viciadas para resolver a mesma instância do problema.

Mauri *et al.* (2017) apresentaram um algoritmo de *Clustering Search* para o problema de solução de contadores de tráfego. Neste caso, os autores utilizaram uma heurística denominada C3 para gerar as soluções iniciais e uma busca local aleatória para afunilar as soluções encontradas.

Abreu *et al.* (2018) utilizaram um modelo matemático de localização de sensores de contagem de veículos, para um problema de estimativa de fluxo de veículos. Neste modelo foram incorporadas restrições relativas ao número de sensores e também as faixas de tráfego observadas. Os autores propuseram um algoritmo *Branch-and-Cut*, para que, a partir de dados reais da malha rodoviária pudesse ser avaliado o algoritmo com outras três heurísticas, e também sem elas.

Vieira et al (2019) apresentam um método de solução guloso, exato e híbrido para solucionar o problema de localização de contadores de tráfego. Para os testes computacionais, foram utilizados dados reais da malha rodoviária brasileira.

González *et al.* (2019) propuseram uma nova abordagem para o tradicional problema de localização de contagem de tráfego, o problema tradicional consistia em determinar a localização das estações de contagem que poderiam cobrir uma rede da melhor forma possível, com o objetivo de obter fluxos de tráfego. Já na nova abordagem, eles utilizaram dois métodos, um *Branch-and-Cut* e um *Clustering Search* para definir as localizações desses contadores. Além disso, é feito uma comparação ao fim com um algoritmo genético apresentado na literatura a fim de comparar os resultados obtidos.

Monteiro (2019) propôs um algoritmo *Clustering Search* para solucionar o problema de solução de contadores de tráfego, porém, implementando as meta-heurísticas GRASP e ILS como geradoras de solução. Esse algoritmo ainda não havia sido proposto na literatura, porém, apresentou resultados iguais ou superiores aos melhores apresentados até então na literatura para todas as instâncias.

Abreu *et al.* (2020) propôs uma nova instância do PLSR considerando dados reais da malha rodoviária brasileira, junto de um algoritmo de *Branch-and-Cut* (B&C) e uma heurística *Clustering Search* (CS) para encontrar soluções para tal problema. O algoritmo de B&C apresentado por Abreu *et al.* (2020) considera um modelo matemático relaxado do problema, gerando cortes baseados em soluções inteiras que são encontradas durante a árvore ramificada. A heurística CS é responsável por dividir

o espaço de busca em *clusters* e aplicar a busca local onde for mais promissor. Foram criados diversos cenários variando o número máximo de sensores e o número de pistas monitoradas, conseguindo boas soluções em 92,31% dos cenários.

Camara *et al.* (2020) demonstra uma abordagem multiobjetiva para localização de contadores de tráfego em redes de transporte. Esse trabalho apresenta uma nova formulação matemática e sua aplicação a dados reais, e também o auxílio no planejamento estratégico do sistema rodoviário. Para os experimentos computacionais foram utilizados o modelo matemático em linguagem C++ e, a fins de comparação, o CPLEX.

Franca *et al.* (2020) apresenta uma revisão sistemática sobre o PLSR, com o objetivo de trazer suas principais características, formulação matemática, além do desenvolvimento de algoritmos para a solução do problema, dando ênfase ao algoritmo *Branch-and-Cut*. Foram analisados os principais resultados de 15 anos de publicações sobre o tema, tanto em bases de dados brasileiras quanto internacionais. Os autores também identificaram algumas limitações que o problema ainda apresenta, como a dificuldade de encontrar caminhos mínimos entre um par O–D com fluxos evasivos.

2.2 CONSIDERAÇÕES

Apesar de vários trabalhos, há um certo espaço na literatura para utilizar o ALNS para solucionar o PLSR, já que o ALNS ainda não foi utilizado para solucionar o PLSR.

Para realizar os testes computacionais, serão utilizadas as instâncias propostas por Abreu *et al.* (2020), para que assim, seja possível realizar a comparação entre os resultados obtidos com os que já estão na literatura.

3 METODOLOGIA

Neste trabalho, a metodologia consiste em resolver o Problema de Localização de Sensores em Rede (PLSR) utilizando a meta-heurística *Adaptive Large Neighborhood Search* (ALNS), que é, segundo Pisinger e Ropke (2007), uma extensão do *Large Neighborhood Search* (LNS) proposto por Shaw (1997).

Segundo Ribeiro *et al.* (2013), a meta-heurística LNS permite a destruição e reconstrução de uma solução, enquanto o ALNS utiliza uma camada adaptativa a mais, que permite que a meta-heurística se adapte conforme a instância sendo resolvida. Para isso, ela pontua as heurísticas, tanto de destruição quanto de reconstrução, que mais vem obtendo destaque.

3.1 FUNCIONAMENTO DO ALNS

Ainda segundo Ribeiro *et al*. (2013), a cada iteração o algoritmo destrói parte de uma solução *s* e a reconstrói gerando uma nova solução *s'*. A partir daí, seguindo alguns critérios, se essa solução for melhor do que a anterior, a busca contínua a partir de *s*'; caso essa solução não seja melhor que a anterior, ela ainda assim tem uma certa probabilidade de continuar a partir dela.

Para a construção da solução inicial, foi utilizada uma heurística gulosa, onde basicamente foram adicionados contadores em todas as arestas da solução. Em seguida, era verificado e removido o sensor da aresta que menos diminuía a área de cobertura ao remover o contador, e assim em diante até que todas as restrições mencionadas no modelo matemático fossem atendidas como mostrado na imagem abaixo.

A heurística é apresentada na Figura 2, na qual o parâmetro *flag* é utilizado apenas para percorrer a solução, *arestasmax* é o número máximo de arestas, *paresCobertos* é utilizado para guardar o número de pares cobertos da solução que menos descobre a malha até então, por fim *posRemover* é a posição desta aresta.

Figura 2 – Pseudocódigo da heurística construtiva.

```
PARA flag DE 0 ATÉ arestasmax FAÇA
    adicionarContador(flag)
FIM-PARA
ENQUANTO (Solução não atender (3) OU (4)) FAÇA
    paresCobertos ← -1
    posRemover ← -1
    PARA flag DE 0 ATÉ arestasmax FAÇA
        removerContador(flag)
        SE s.paresCobertos > paresCobertos ENTÃO
            posRemover ← flag
        FIM-SE
        adicionarContador(flag)
    FIM-PARA
    removeContador(posRemover)
FIM-ENQUANTO
```

Fonte: Autor.

A meta-heurística também apresenta os conceitos de vizinhança ampla que segundo Pisinger e Ropke (2007), podem acabar não sendo capazes de explorar bem a solução caso as heurísticas de destruição e construção restrinjam uma pequena parte da solução. Além disso, caso essas heurísticas destruam uma grande parte da solução, as soluções podem ser fracas e demandarem um grande gasto computacional.

Mas, tudo isso diz respeito à meta-heurística LNS, a parte adaptativa entra na seleção dessas heurísticas de construção e destruição. Para tal, é utilizada uma estratégia na qual cada heurística tem a sua pontuação, que é dada de acordo com as soluções que ela proporciona. Quanto maior a pontuação, maior a probabilidade de ela ser selecionada novamente. A Figura 3 apresenta o pseudocódigo do ALNS implementado neste trabalho e utilizado para realização dos testes computacionais.

Figura 3 – Pseudocódigo do ALNS utilizado.

```
1   s* ← s;   T ← T0;   Te ← 0;   Texec ← 0;   i ← 0;
2   ωH+ ← 1;   Para todo H+   ωH- ← 1; Para todo H-
3   πH+ ← 0;   Para todo H+   πH- ← 0; Para todo H-
4   ENQUANTO (Texec < Tmax ) FAÇA
5       i ← 0;
6       ENQUANTO ( i < ALNSmax ) FAÇA
7           i ← i + 1;   j ← j + 1;   s' ← s;
8           y ← número aleatório entre 1 e ( | ζ* ( n° de arestas ) | - 1 ) ;
9           SELECIONAR (uma heurística de destruição H- ) ;
10          APLICAR ( H- ( s' ) );
11          SELECIONAR (uma heurística de reconstrução H+);
12          APLICAR ( H+ ( s' ) );
13          lH+ ← lH+ + 1 ;   lH- ← lH- + 1 ;
14          SE ( f ( s' ) > f ( s ) )
15              s ← s' ;
16              SE ( f ( s ) > f ( s* ) )
17                  s* ← s ;   πH+ ← πH+ + σ1 ;   πH- ← πH- + σ1 ;
18              SENÃO
19                  πH+ ← πH+ + σ2 ;   πH- ← πH- + σ2 ;
20              FIM-SE
21          SENÃO
22              TOMAR ( x Є [ 0 , 1 ] )
23              SE ( x > e^-(f(s)-f(s'))/T )
24                  s ← s' ; πH+ ← πH+ + σ3 ; πH- ← πH- + σ3 ;
25              FIM-SE
26          FIM-SE
27          SE ( j = φ )
28              j = 0 ;
29              ATUALIZAR ( ωH+ , ωH- , oH+ , oH- , πH+ , πH- )
30          FIM-SE
31          T ← α * T ;
32      FIM-ENQUANTO
33      Texec ← tempo de execução ;
34  FIM-ENQUANTO
35  RETORNAR ( s* ) ;
```

Fonte: Adaptado de Ribeiro et al. (2013).

A partir de uma solução s, gerada pela heurística construtiva, o ALNS irá buscar os vizinhos de s. Para tal, nas linhas 9 e 11, com base em um algoritmo de

roleta, é escolhida uma heurística de destruição responsável por remover os sensores e uma heurística de reconstrução responsável por adicionar os sensores, criando assim, uma solução vizinha de s.

Esse algoritmo de roleta tem como objetivo premiar as heurísticas com maior eficiência através de uma probabilidade maior de ser executada. Essa eficiência é medida através das iterações anteriores, fazendo com que a busca por vizinhos se torne adaptada. É importante ressaltar que essa eficiência dos métodos é medida através da função objetivo mencionada no modelo matemático apresentado anteriormente.

Para cada heurística i é atribuído um peso ω_i que indica a performance da mesma nas últimas iterações. Sendo assim, j é uma heurística e h a quantidade de heurísticas que podem ser selecionadas, onde a probabilidade de j ser selecionada é dada por $w_j / \sum_{i=1}^{h} w_j$, onde as heurísticas de destruição são independentes das heurísticas de reconstrução.

Esses pesos são ajustados da linha 14 até a 26 de acordo com o tamanho dos segmentos, este tamanho define quantas iterações aconteceram sem atualizar os pesos, que é dado por φ. Ou seja, entre as linhas 27 e 30, após φ iterações, os pesos de cada heurística, tanto das destrutivas quanto as reconstrutivas, são atualizados de acordo com o *score* obtido durante aquele segmento, sendo que, na primeira iteração do método todas possuem o mesmo peso. Esses *scores* são dados pelos parâmetros, σ_1, σ_2 e σ_3 onde σ_1 é dado às heurísticas quando uma solução s' encontrada é melhor do que a solução $s*$, σ_2 quando s' é melhor que s e, por fim, σ_3 é dado quando s' for pior do que s, mas for aceito.

Para realizar a abstração de uma solução do problema para o algoritmo, foram utilizadas quatro variáveis para fazer o armazenamento de algumas informações, sendo elas, número de arestas com contador, número de arestas sem contador, número de faixas cobertas e número de pares cobertos.

Além disso, foram utilizados dois vetores para representar uma solução, sendo ambos proporcionais ao número de arestas no problema, um deles para representar as arestas que possuem os contadores e outro para o caso contrário. Π_i é o score da heurística i e o_{ij} o número de vezes em que i foi selecionada no último segmento j, por fim, o fator de reação, dado por η sendo o responsável por regular a velocidade de ajuste dos pesos em caso de mudança da eficiência da heurística.

3.2 HEURÍSTICAS DE DESTRUIÇÃO

A seguir são apresentadas as heurísticas de destruição que foram utilizadas no ALNS implementado.

3.2.1 *Shaw Removal*

Segundo Shaw (1997), essa heurística consiste em eliminar elementos semelhantes da solução. Portanto, é necessário calcular o grau de similaridade entre esses elementos. Esse cálculo é feito por meio de seus custos na solução atual e da diferença absoluta entre eles. No primeiro caso, onde nenhum elemento foi removido, um elemento é selecionado e o elemento com maior grau de similaridade é removido. O segundo caso é quando já se tem algum elemento removido da solução, então um desses elementos é comparado aos que ainda estão na solução, o mais similar é removido, e assim sucessivamente.

Figura 4 – Pseudocódigo da heurística *Shaw Removal* implementada no ALNS.

```
1   aresta : a = aresta aleatória de s removida;
2   conjunto de arestas removidas: D = {a};
3   ENQUANTO ( |D| < q ) FAÇA
4      a = aresta aleatória em D;
5      Vetor: L = vetor que contém todas as arestas que estão por remover;
6      ordenar L tal que: i < j => R ( a , L[i] ) < R ( a , L[j] );
7      escolher número aleatório n do intervalo [0,1);
8      a = { L [ nº |L| ] };
9      D = D ∪ { a };
10  FIM-ENQUANTO
11  remova as arestas em D de s;
```

Fonte: Adaptado de Shaw (1997).

Assim como mostrado na Figura 4, nas linhas 1 e 2 é escolhida uma aresta aleatória presente na solução s, que é colocada em um vetor *D* que irá conter o conjunto de arestas removidas da solução. Após isto, na linha 4 é escolhida uma aresta que está no vetor *D* para ser usada como pivô para ser comparada com as outras arestas da solução s que estão no vetor *L*. Após isto, o vetor *L* é ordenado na linha 6 de acordo com as arestas mais parecidas em termos de custo com o pivô, este

custo é calculado de acordo com a função objetivo, removendo e adicionando um contador nestas arestas.

É introduzida também, uma aleatoriedade segundo o valor de *p* na linha 7, isso se dá por que a posição do vetor *L* à qual se seleciona a próxima aresta a remover depende de um número aleatório *n* compreendido entre 0 e 1 elevado a *p*. Ou seja, se *p* for um valor alto, o resultado será um valor baixo, o que resultará em uma aleatoriedade baixa, sendo assim, escolhida uma aresta no início do vetor *L*, caso contrário, se *p* for um valor baixo, o resultado será um valor alto e, nesse caso, a seleção será de arestas mais próximas do fim do vetor *L*.

3.2.2 *Random Removal*

De acordo com Lutz (2014), essa é a heurística de destruição mais simples, pois ela simplesmente remove repetidamente os elementos de uma solução aleatoriamente, até que um certo nível de destruição seja atingido. Apesar de simples, essa é uma heurística extremamente importante para o ALNS, pois além de criar uma diversidade de soluções, ela permite que o algoritmo saia de certos ótimos locais, uma vez que ela pode remover os elementos independente de seus custos.

3.2.3 *Worst Removal*

Essa heurística basicamente remove o pior elemento da solução em termos de custo até que um certo nível de destruição seja atingido. Porém, a cada iteração é recalculado o custo desses elementos, garantindo assim que sempre o pior elemento seja removido.

Figura 5 – Pseudocódigo da heurística *Worst Removal* implementada no ALNS.

```
1   Vetor: L = vetor que contém todas as arestas que estão por remover;
2   ordenar L por ordem decrescente de custo;
3   qtdRemovidos = 0;
4   ENQUANTO ( qtdRemovidos < q ) FAÇA
5      escolher número aleatório n do intervalo [0,1);
6      a = { L [ n^p |L| ] };
7      remover a de s;
8      qtdRemovidos = qtdRemovidos  + 1;
9      ordenar L por ordem decrescente de custo;
10  FIM-ENQUANTO
```

Fonte: Autor.

A Figura 5 apresenta o pseudocódigo do *Worst Removal* implementado para o ALNS. Enquanto o *Shaw Removal* tem a intenção de remover as arestas que são semelhantes umas com as outras, o *Worst Removal* pretende remover os sensores das arestas que não estão contribuindo com a solução. Ou seja, a principal diferença dessas heurísticas está na ordenação do vetor L, em uma a ordenação é feita de acordo com a semelhança entre o custo das arestas e a outra é feita de acordo com o quão custosa aquela aresta é para a solução.

3.3 HEURÍSTICAS DE RECONSTRUÇÃO

A seguir são apresentadas as heurísticas de reconstrução que foram utilizadas na construção do ALNS.

3.3.1 *Basic Greedy Repair*

A primeira heurística de reconstrução exige um alto custo computacional, pois ela precisa calcular quanto cada um dos elementos removidos melhora em sua melhor posição. Após isso, o elemento que mais beneficia a solução é selecionado e realocado na solução. Então, o processo é repetido até que todos os elementos removidos sejam realocados.

Essa heurística é basicamente a mesma utilizada na heurística construtiva, com a diferença de que ao invés de ir removendo os elementos, ela vai alocando-os de acordo com os seus custos, quanto menos custoso para a solução, ou seja, quanto mais aquele elemento contribuir positivamente para a solução ele será realocado com prioridade.

3.3.2 *Random Repair*

Em contrapartida da heurística anterior, essa tem um custo computacional bastante leve. Basicamente, ela percorre o vetor de soluções sem contadores, ou seja, os que se encontram destruídos, e vai alocando-os até que alguma restrição seja violada.

3.3.3 *Worst Element Repair*

Essa heurística inicialmente adiciona o pior elemento que foi removido por uma das heurísticas de destruição, isso para sair de possíveis máximos locais. Feito isso, a heurística reconstrói o restante da solução baseando-se em uma das duas

heurísticas de reconstrução mencionadas acima, tendo ambas a mesma chance de serem selecionadas.

4 TESTES COMPUTACIONAIS

Foi realizada uma bateria de testes nas instâncias propostas por Abreu *et al.* (2020) para comprovar a eficácia do ALNS assim como mencionado nos objetivos do trabalho. Os testes foram executados em uma máquina equipada com um processador Intel Xeon ®E5–1650 v3 3.5HZ, 64GB de memória RAM e sistema operacional Ubuntu. O algoritmo foi desenvolvido em C/C++ e compilado pelo GNU GCC *Compiler* versão 7.0.3.

4.1 CALIBRAÇÃO DOS PARÂMETROS

Foi adotada uma estratégia de calibração dos parâmetros semelhante a de Ribeiro *et al.* (2013), onde basicamente na primeira etapa é feita a inicialização dos parâmetros e na segunda o aperfeiçoamento.

Para a etapa de inicialização dos parâmetros, os valores iniciais foram atribuídos de forma empírica, baseando principalmente em alguns dos outros trabalhos mencionados na revisão da literatura relacionados ao ALNS.

Já na etapa de aperfeiçoamento dos parâmetros, para melhorar a eficácia do algoritmo, foi escolhida uma instância aleatoriamente e, para cada um dos parâmetros, os demais eram inalterados e o escolhido era variado dentro de um intervalo específico. A melhora era considerada quando o tempo necessário para encontrar uma determinada solução era menor do que anteriormente. Isso foi feito 3 vezes para comprovar que não fora devido a alguma aleatoriedade. Caso não houvesse melhora após algumas tentativas, este valor era fixado e então um novo parâmetro era escolhido para ser alterado e essa etapa era novamente executada.

Tabela 5 - Intervalo de valores possíveis para cada parâmetro durante a calibração.

Parâmetro	Significado	Intervalo de valores
φ	Tamanho dos segmentos	{25; 50; 100}
η	Fator de reação	{0,01; 0,05; 0,1}
ψ	Percentual de remoção	{0,1; 0,3; 0,5}
T_0	Temperatura inicial	{10; 100; 500; 1000}
Tc	Temperatura de congelamento	{0,01; 0,1}
α	Taxa de resfriamento	{0,975; 0,985; 0,995}
$\sigma 1$	Pontuação se a melhor solução é encontrada	{20; 30; 50}
$\sigma 2$	Pontuação se a solução corrente é superada	{10; 15; 20}
$\sigma 3$	Pontuação se uma solução pior for aceita	{5; 7; 9}

Para realizar a calibração dos parâmetros, o conjunto de testes foi executado fixando a instância em Acre para uma instância um pouco menor e RJ para uma um pouco maior. Após a finalização da calibração, os parâmetros foram definidos conforme apresentado na Tabela 6.

Tabela 6 - Valores dos parâmetros do *ALNS* após a calibração.

Parâmetro	Significado	Valor
φ	Tamanho dos segmentos	50
η	Fator de reação	0,05
ψ	Percentual de remoção	0,1
T_0	Temperatura inicial	10
Tc	Temperatura de congelamento	0,01
α	Taxa de resfriamento	0,975
$\sigma 1$	Pontuação se a melhor solução é encontrada	50
$\sigma 2$	Pontuação se a solução corrente é superada	10
$\sigma 3$	Pontuação se uma solução pior for aceita	9

5 RESULTADOS OBTIDOS

Foram definidos alguns principais objetivos para apresentar os resultados obtidos pelo ALNS aplicado ao PLSR, sendo estes objetivos, analisar o quão robusto é o algoritmo, e também compará-lo a outros já existentes na literatura como, por exemplo, o de Abreu *et al.* (2020).

Para encontrar os caminhos mais curtos entre todos os pares OD $w \in W$, foi utilizado um procedimento chamado Breadth First Search - BFS (Ahuja et al., 2014). Basicamente, quando o caminho de um par OD w é maior que zero, ele pode ser considerado coberto, ou seja, ao menos um sensor está presente em um segmento naquele caminho. Este procedimento é aplicado para cada par OD $w \in W$, e é utilizado para calcular o número de pares cobertos na solução após criar uma solução inicial ou gerar uma solução vizinha.

Tabela 7 - Dados de cada instância.

Instância	**|*V*|**	**|*A*|**	**|*M*|**	**|*W*|**	**|*L*|**
Acre - AC	91	114	20	190	232
Alagoas - AL	192	242	97	4656	490
Amazonas - AM	207	210	37	666	432
Amapá - AP	75	98	13	78	196
Bahia - BA	970	1268	395	77815	2568
Ceará - CE	514	721	177	15576	1492
Espírito Santo - ES	352	462	75	2775	940
Goiás/Distrito Federal – GO/DF	929	1294	241	28920	2758
Maranhão - MA	308	405	163	13203	848
Minas Gerais - MG	2105	2543	803	322003	5342
Mato Grosso do Sul - MS	532	686	76	2850	1410
Mato Grosso- MT	1339	1696	140	9730	3450
Pará - PA	461	538	122	7381	1100
Paraíba - PB	431	526	213	22578	1112
Pernambuco - PE	426	536	172	14706	1168
Piauí - PI	486	631	212	22366	1280
Paraná - PR	1018	1316	381	72390	2882
Rio de Janeiro - RJ	850	1067	86	3655	2268
Rio Grande do Norte - RN	378	478	160	12720	1012
Rondônia - RO	251	322	50	1225	662
Roraima - RR	153	175	13	78	350
Rio Grande do Sul - RS	834	1019	389	75466	2216
Santa Catarina - SC	531	654	266	35245	1360
Sergipe - SE	219	282	74	2701	578
São Paulo - SP	1458	1861	606	183315	4798
Tocantins - TO	432	584	134	8911	1196

Fonte: Abreu *et al.* (2020).

A Tabela 7 apresenta os dados de cada instância, onde a coluna **|*V*|** representa o número de vértices ou nós, |*A*| a quantidade de arestas, |*M*| o número de municípios da instância, |*W*| a quantidade de pares OD e |*L*| o valor referente as faixas. Vale ressaltar que cada município é considerado uma origem para todos os demais municípios da rede, além disso, a quantidade de pares OD foi calculado como $|\boldsymbol{W}| = \frac{\boldsymbol{M}(\boldsymbol{M}-\mathbf{1})}{\mathbf{2}}$,

Os dados extraídos destes resultados para fins de comparação em geral foram, melhor FO, que representa o valor da função objetivo da solução obtida, onde neste problema, por se tratar de um problema de maximização, quanto maior, melhor a solução, uma média desses valores, uma vez que para cada instância o método foi executado mais de uma vez, uma variação que é obtida comparando a melhor FO obtida com a média e por fim o tempo médio.

5.1 RESULTADOS COMPUTACIONAIS

É apresentado um resumo dos resultados computacionais obtidos durante os testes na Tabela 7 logo abaixo. Esses dados foram obtidos após executar o método 5 vezes com a duração de 3600 segundos cada execução. A variação, ou desvio, é calculada por $Var = \frac{MelhorFO - Média}{MelhorFO} \times 100$.

É possível observar pela Tabela 7 que o método teve uma variação de no máximo 0,03% na instância de GO/DF, e uma média de 0,01%. Além disso, foram apresentadas soluções viáveis até mesmo nos piores casos onde o tempo médio chegou próximo ao limite de 3600 segundos. Portanto, pode-se concluir que o ALNS conseguiu retornar soluções viáveis e de uma forma bem robusta, com uma variação relativamente baixa.

Tabela 8 - Resumo dos resultados computacionais.

Inst.	*Adaptive Large Neighborhood Search*			
	Melhor FO	**Média**	**Var**	**Tempo médio (s)**
AC	186	186,00	0,00	0,70
AL	4.511	4.510,40	0,01	1422,60
AM	666	666,00	0,00	0,06
AP	76	76,00	0,00	1,30
BA	76.359	76.353,20	0,00	1626,80
CE	14.971	14.967,60	0,02	1456,10
ES	2.740	2.739,20	0,02	910,90
GO/DF	28.375	28.366,40	0,03	3590,20
MA	12.865	12.864,00	0,00	3513,48
MG	320.316	320.302,80	0,00	3598,00
MS	2.840	2.839,80	0,00	3059,40
MT	9.684	9.683,20	0,00	10,70
PA	7.315	7.315,00	0,00	3036,90
PB	22.059	22.057,20	0,00	3400,03
PE	14.469	14.465,60	0,02	3247,70
PI	22.041	22.037,40	0,01	3542,03
PR	70.274	70.262,60	0,01	3408,93
RJ	3.642	3.642,00	0,00	5,61
RN	12.510	12.507,80	0,01	55,85
RO	1.210	1.209,80	0,01	549,07
RR	78	78,00	0,00	0,10
RS	74.546	74.540,40	0,00	3501,53
SC	34.711	34.704,20	0,01	3350,82
SE	2.644	2.643,60	0,01	3001,92
SP	181.492	181.483,40	0,00	210,16
TO	8.787	8.787,00	0,00	2246,12
Médias	35.744,89	35.741,86	0,00	1874,88

5.2 COMPARAÇÃO COM A LITERATURA

Na Tabela 8 está uma comparação dos resultados obtidos pelo ALNS com uma outra abordagem da literatura, o algoritmo de *Branch and Cut* proposto por Abreu *et al.* (2020).

Os resultados que se encontram em negrito foram os melhores resultados encontrados para cada instância, enquanto os que estão em negrito e sublinhado representam a melhor solução encontrada até o momento. Quando o resultado está

em negrito em apenas um dos algoritmos, significa que ele encontrou um resultado melhor que o outro, se ambos estão em negrito é porque ambos encontraram o mesmo resultado e, portanto, aquele é o melhor.

Tabela 9 - Comparação dos resultados com o algoritmo *Branch and Cut*.

Inst.	***Branch and Cut*[3]**			***Adaptive Large Neighborhood Search***		
	Melhor FO	**Gap**	**Tempo médio (s)**	**Melhor FO**	**Var**	**Tempo médio (s)**
AC	**186**	0.00	0.21	**186**	0,00	0,70
AL	**4.511**	0.29	3600.00	**4.511**	0,01	1422,60
AM	**666**	0.00	0.04	**666**	0,00	0,06
AP	**76**	0.00	1.05	**76**	0,00	1,30
BA	47.757	62.27	3600.00	**76.359**	0,00	1626,80
CE	5.994	157.82	3600.00	**14.971**	0,02	1456,10
ES	**2.740**	0.02	910,9	**2.740**	0,02	910,90
GO/DF	13.562	112.77	3600.00	**28.375**	0,03	3590,20
MA	12.863	0.94	3600.00	**12.865**	0,00	3513,48
MG	300.756	7.06	3600.00	**320.316**	0,00	3598,00
MS	2.835	0.53	2388.57	**2.840**	0,00	3059,40
MT	7.989	21.79	3600.00	**9.684**	0,00	10,70
PA	**7.318**	0.29	3600.00	7.315	0,00	3036,90
PB	18.778	19.03	3600.00	**22.059**	0,00	3400,03
PE	14.367	1.48	3600.00	**14.469**	0,02	3247,70
PI	18.848	17.65	3600.00	**22.041**	0,01	3542,03
PR	57.502	25.54	3600.00	**70.274**	0,01	3408,93
RJ	**3.655**	0.00	29.15	3.642	0,00	5,61
RN	12.505	0.67	3600.00	**12.510**	0,01	55,85
RO	**1.210**	1.05	3600.00	**1.210**	0,01	549,07
RR	**78**	0.00	0.01	**78**	0,00	0,10
RS	58.303	28.85	3600.00	**74.546**	0,00	3501,53
SC	28.793	21.37	3600.00	**34.711**	0,01	3350,82
SE	**2.645**	0.11	3600.00	2.644	0,01	3001,92
SP	135.105	35.51	3600.00	**181.492**	0,00	210,16
TO	**8.787**	0.70	3600.00	**8.787**	0,00	2246,12
Médias	29531,88	19,83	2758,84	35744.89	0.00	1874.88

[3] Resultados do algoritmo *Branch and Cut* apresentado por Abreu *et al*. (2020).

Quanto a resultados é possível observar que o ALNS obteve uma solução pior que a apresentada pelo *Branch and Cut* em apenas 3 (11,5%) instâncias, encontrando uma solução com o mesmo resultado em 8 (30,7%) instâncias e apresentando uma solução melhor em 15 (57,8%) instâncias.

Levando em consideração que ambos os algoritmos foram executados na mesma máquina, é possível fazer também uma comparação do tempo, O *Branch and Cut* teve uma média de tempo de 2758,84 segundos, enquanto o ALNS 1874,88 segundos, sendo aproximadamente 68% mais rápido.

A Tabela 9 apresenta a comparação entre o *Clustering Search* apresentado por Abreu et al. (2020) com o ALNS apresentado neste trabalho. Assim como na comparação anterior, os testes foram executados no mesmo computador, portanto ainda é possível fazer não só a comparação das soluções como também do tempo médio.

Analisando os resultados apresentados, é possível perceber que, diferente da comparação anterior, o ALNS apresentou uma solução semelhante ao *Clustering Search* em 6 (23%) instâncias e uma solução pior no restante. Quanto ao tempo, o algoritmo apresentado por Abreu et al. (2020) obteve uma média de 1186,8 segundos, que quando comparado ao ALNS é 63% mais rápido.

Do ponto de vista computacional, quando se diz respeito a resultados e a velocidade, o ALNS apresentou bons resultados quando comparado ao *Branch and Cut*, porém mesmo sendo uma heurística bastante flexível com relação a seu desenvolvimento, sua performance deixou a desejar quando comparado ao *Clustering Search*. Apesar de não achar soluções melhores, as soluções obtidas estão próximas das melhores apresentadas na literatura.

Tabela 10 - Comparação dos resultados com o algoritmo *Clustering Search.*

Inst.	***Clustering Search***[4]			***Adaptive Large Neighborhood Search***		
	Melhor FO	**Var**	**Tempo médio (s)**	**Melhor FO**	**Var**	**Tempo médio (s)**
AC	**186**	0.00	0.47	**186**	0,00	0,70
AL	**4.511**	0.02	1453.87	**4.511**	0,01	1422,60
AM	**666**	0.00	0.06	**666**	0,00	0,06
AP	**76**	0.00	0.17	**76**	0,00	1,30
BA	**77.186**	0.04	1793.55	76.359	0,00	1626,80
CE	**15.305**	0.10	1118.05	14.971	0,02	1456,10
ES	**2.743**	0.01	1072.23	2.740	0,02	910,90
GO/DF	**28.723**	0.03	1372.74	28.375	0,03	3590,20
MA	**12.883**	0.05	2432.26	12.865	0,00	3513,48
MG	**321.144**	0.00	2067.32	320.316	0,00	3598,00
MS	**2.843**	0.02	1267.46	2.840	0,00	3059,40
MT	**9.726**	0.00	1744.06	9.684	0,00	10,70
PA	**7.323**	0.01	1173.65	7.315	0,00	3036,90
PB	**22.256**	0.01	1242.99	22.059	0,00	3400,03
PE	**14.501**	0.02	1585.32	14.469	0,02	3247,70
PI	**22.091**	0.02	1334.65	22.041	0,01	3542,03
PR	**71.957**	0.02	1993.48	70.274	0,01	3408,93
RJ	**3.655**	0.00	26.20	3.642	0,00	5,61
RN	**12.528**	0.02	1407.19	12.510	0,01	55,85
RO	**1.211**	0.02	18.89	1.210	0,01	549,07
RR	**78**	0.00	0.02	**78**	0,00	0,10
RS	**74.873**	0.01	2469.71	74.546	0,00	3501,53
SC	**34.806**	0.02	1578.01	34.711	0,01	3350,82
SE	**2.644**	0.02	197.75	**2.644**	0,01	3001,92
SP	**182.603**	0.01	1619.37	181.492	0,00	210,16
TO	**8.791**	0.05	1887.29	8.787	0,00	2246,12
Médias	35973.42	0.02	1186.79	35744.89	0.00	1874.88

[4] Resultados do algoritmo *Clustering Search* apresentado por Abreu et al. (2020).

6 CONCLUSÃO

Este trabalho apresentou uma nova forma de solucionar o Problema de Localização de Sensores em Rede (PLSR). O PLSR já é um problema conhecido e encontrar melhores formas para resolvê-lo é extremamente necessário para buscar poupar custos e ainda possibilitar o monitoramento das rodovias.

A meta-heuristica *Adaptive Large Neighborhood Search* (ALNS) foi proposta para solucionar o PLSR, pois além de ainda não ter sido explorada para resolver este problema em específico, a mesma apresentou bons resultados na literatura ao resolver problemas semelhantes.

Para comprovar o desempenho do ALNS para solucionar o PLSR, foram realizados diversos testes computacionais a partir de todos os Estados brasileiros a partir de dados reais da malha rodoviária. Com uma média de 1874,88 segundos, o ALNS apresentou bons resultados em todas as instâncias.

Além disso, o ALNS apresentou um resultado melhor em 15 instâncias quando comparado ao *Branch and Cut* apresentado por Abreu *et al.* (2020) com uma variação média de apenas 0,010%, porém, quando comparado ao *Clustering Search* o ALNS não apresentou nenhuma melhora nas soluções apresentadas no mesmo trabalho.

Os resultados obtidos nos mostram que a meta-heurística proposta é eficaz para solucionar o problema proposto, uma vez que soluções próximas às melhores conhecidas foram obtidas para todas as instâncias. Deste modo, é possível afirmar que o ALNS é uma alternativa para solucionar o PLSR, já que o algoritmo permite alterações em suas heurísticas de destruição e reconstrução.

Portanto, em trabalhos futuros, pode ser interessante verificar o desempenho do ALNS adicionando diferentes heurísticas de construção e reconstrução, ou até mesmo alterar as que foram implementadas neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, V. H. S. Revisão bibliográfica sistemática sobre o problema de localização de sensores de tráfego em redes. *In*: **Anais [...]** Congresso Nacional de Pesquisa em Transportes, 2017.

ABREU, V. H. S.; GONZÁLEZ, P. H.; MAURI, G. R.; RIBEIRO, G. M.; ORRICO, R. D.; CAMPOS JÚNIOR, N. F. R.; ABRAMIDES, C. A. Network sensor location problem with monitored lanes: branch-and-cut and clustering search solution techniques. **Computers & Industrial Engineering**, v. 150, 2020.

ABREU, V. H. S.; GONZÁLEZ, P. H.; RIBEIRO, G. M.; ORRICO, R. D.; GOMES, H. A.; PERIM, L. R. Técnicas Heurísticas e um Branch-and-Cut para o Problema de Localização de Sensores de Tráfego com Limitações quanto ao Número de Sensores e de Faixas Observadas. *In*: **Anais [...]** L Simpósio Brasileiro de Pesquisa Operacional, SBPO, 2018.

AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. Network flows. **Pearson Education**, 2014.

BBC. **Crise revela dependência de transporte rodoviário que é 'mais barato e dá voto'.** Disponível em: http://www.bbc.com/portuguese/brasil-44247460. Acesso em: 22 jun. de 2021.

BRASIL. **Plano Nacional de Contagem de Tráfego – PNCT: Dados coletados pelo DNIT trazem estimativa de Volume Médio Diário Anual para malha federal.** Disponível em: http://www.gov.br/dnit/pt-br/assuntos/noticias/plano-nacional-de-contagem-de-trafego-2013-pnct-dados-coletados-pelo-dnit-trazem-estimativa-de-volume-medio-diario-anual-para-toda-malha-federal. Acesso em: 22 jun. de 2021.

CASTILLO, E.; NOGAL, M.; RIVAS, A.; SÁNCHEZ-CAMBRONERO, S. Observability of traffic networks. Optimal location of counting and scanning devices. **Transportmetrica B: Transport Dynamics**, v. 1, n. 1, p. 68-102, 2013.

CAMARA, M. V. O.; FERRARI, T.; VIEIRA, B. S.; QUADROS, S. G. R.; ABRAMIDES, C. A.; NUNES, A. O.; CAMPOS JÚNIOR, N. F. R. Abordagem multiobjetivo para localização de contadores de tráfego em redes de transporte. *In*: **Anais [...]** 4º Congresso de Pesquisa e Ensino em Transporte da ANPET, 2020.

CONFEDERAÇÃO NACIONAL DO TRANSPORTE - CNT. **Dnit libera dados recentes da contagem de tráfego em rodovias federais.** Disponível em: http://www.cnt.org.br/agencia-cnt/dnit-libera-dados-da-contagem-de-trafego-em-rodovias-federais-cnt. Acesso em: 22 jun. de 2021.

DEMIR, E; BEKTAŞ, T; LAPORTE, G. An adaptive large neighborhood search heuristic for the pollution-routing problem. **European Journal of Operational Research**, v. 223, n. 2, p. 346-359, 2012.

FISK, C. On combining maximum entropy trip matrix estimation with user optimal assignment. **Transportation Research Part B: Methodological**, 22(1):69–73, 1988.

FRANCA, L. S.; GONZÁLEZ, P. H.; CLÍMACO, F. G. N.; ALVES, C. D.; CAMPOS JÚNIOR, N. F. R. Revisão sistemática sobre o problema de localização de sensores em redes visando a redução da evasão do tráfego rodoviário: abordagens, algoritmos e aplicações. *In*: **Anais [...]** 34º Congresso de Pesquisa e Ensino em Transporte da ANPET, 2020.

GENTILI, M.; MIRCHANDANI, P. B. Locating sensors on traffic networks: Models, challenges and research opportunities. **Transportation research part C: emerging technologies**, v. 24, p. 227-255, 2012.

GONZÁLEZ, P. H.; CLIMACO, G.; MAURI, G. R.; VIEIRA, B. S.; RIBEIRO, G. M.; ORRICO, R. D.; SIMONETTI, L.; PERIM, L. R.; HOFFMANN, I. C. S. New approaches for the traffic counting location problem. **Expert systems with applications**, v. 132, p. 189-198, 2019.

GONZÁLEZ, P. H.; MARACANÃ, A.; MARACANÃ, R. D. J. R.; CLIMACO, G.; SIMONETTI, L.; VIEIRA, B. S.; NUNES, A. O. BRKGA com população mista aplicado ao problema de localização de contadores de tráfego em redes de transporte. *In*: **Anais [...]** XLIX Simpósio Brasileiro de Pesquisa Operacional, 2017a.

GONZÁLEZ, P. H.; MARACANÃ, A.; MARACANÃ, R. D. J. R.; CLÍMACO, G.; SIMONETTI, L.; VIEIRA, B. S.; NUNES, A. O. Um algoritmo Branch-and-Cut para o problema de Localização Ótima de Contadores de Tráfego em Redes de Transporte. **Anais [...]** XLIX Simpósio Brasileiro de Pesquisa Operacional. 2017b.

HU, S.; PEETA, S.; CHU, C. Identification of vehicle sensor locations for link-based network traffic applications. **Transportation Research Part B: Methodological**, v. 43, n. 8-9, p. 873-894, 2009.

LI, H.; DONG, H.; JIA, L.; REN, M.; Li, S. Analysis of factors that influence the sensor location problem for freeway corridors. **Journal of Advanced Transportation**, v. 49, n. 1, p. 10-28, 2015.

LUTZ, R. Adaptive large neighborhood search, a heuristic for the rich pickup and delivery problem with time windows. **Bachelor thesis at Ulm University**, 2014.

MAURI, G. R.; CLIMACO, G.; VIEIRA, B. S.; RIBEIRO, G. M.; GONZÁLEZ, P. H.; CAMPOS JÚNIOR, N. F. R.; TEIXEIRA JÚNIOR, L. A. R.; NUNES, A. O. Meta-heurística cs para o problema de localização de contadores de tráfego em redes de transporte. **Anais [...]** XLIX SBPO-Simpósio Brasileiro de Pesquisa Operacional, p. 22.

MAURI, G. R.; RIBEIRO, G. M.; LORENA, L. A. N.; LAPORTE, G. An adaptive large neighborhood search for the discrete and continuous berth allocation problem. **Computers & Operations Research**, v. 70, p. 140-154, 2016.

MONTEIRO, R. C. **Clustering search multi-heurística paralelo para resolução do problema de localização de contadores de tráfego em redes de transporte**. 2007. 55 f. Dissertação (Mestrado em Informática). Universidade Federal do Espírito Santo, 2019.

PISINGER, D.; S. ROPKE. A general heuristic for the vehicle routing problem. **Computers & Operations Research**, v. 34, p. 2403–2435, 2007.

RIBEIRO, G. M.; LAPORTE, G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. **Computers & operations research**, v. 39, n. 3, p. 728-735, 2012.

RIBEIRO, G. M.; MAURI, G. R.; LORENA, L. A. N.; LAPORTE, G. Meta-heurística ALNS para resolução do problema de alocação de berços (ALNS metaheuristic to solve the berth allocation problem). **Anais [...]** XXVII ANPET - Congress on Transportation Research and Education, Belém (Brazil), November 04-08, 2013.

SHAW, P. A new local search algorithm providing high quality solutions to vehicle routing problems. **Technical report, University of Strathclyde Glasgow**, 1997.

VIEIRA, B. S.; FERRARI, T.; BAHIENSE, L.; RIBEIRO, G. M.; HOFFMANN, I. C. S.; CAMPOS JÚNIOR, N. F. R. Métodos de Solução Guloso, Exato e Híbrido para o Problema de Localização de Contadores de Tráfego em Redes de Transporte. In **Anais [...]** XXXIII Congresso de Pesquisa e Ensino em Transportes da ANPET, ANPET, Balneário Camboriú (pp. 3781-3792), 2019.

WEN, M.; LINDE, E.; ROPKE, S.; MIRCHANDANI, P.; LARSEN, A. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. **Computers & Operations Research**, v. 76, p. 73-83, 2016.