

OTIMIZAÇÃO DE ROTAS DE FISCALIZAÇÃO URBANA EM ARACATI-CE USANDO O PROBLEMA DO CAIXEIRO VIAJANTE E *OR-TOOLS*

Francisca Jéssica Oliveira Mota *

Diego Rocha Lima **

RESUMO

O Instituto de Qualidade do Meio Ambiente do Aracati (IQUAMA) é a autarquia municipal responsável por assegurar o uso sustentável dos recursos naturais e preservar o equilíbrio dos ecossistemas no município de Aracati-CE, entre suas principais atividades, destaca-se a fiscalização, que envolve o deslocamento de equipes e veículos para atender às demandas e realizar monitoramentos. Atualmente, a elaboração das rotas de fiscalização é feita de forma empírica, baseada no conhecimento prévio e na experiência dos motoristas em relação aos endereços a serem visitados. Essa abordagem, no entanto, carece de fundamentação científica e eficiência operacional. Diante desse cenário, o presente estudo propõe a otimização das rotas de fiscalização urbana, com o objetivo de minimizar ineficiências e aprimorar o desempenho do setor. Para isso, utiliza-se o Problema do Caixeiro Viajante (PCV) como modelo para a gestão de rotas, implementado por meio da ferramenta *Google OR-Tools* Solver de Roteamento. As rotas atualmente utilizadas são analisadas e comparadas com aquelas geradas pela ferramenta, considerando diferentes parâmetros e estratégias de solução. A avaliação é realizada com base em métricas como redução da distância percorrida e tempo de deslocamento além da análise da atratividade visual das rotas propostas. Os resultados obtidos evidenciam uma maior eficiência operacional com a implementação de rotas otimizadas, com consequente redução significativa nas distâncias percorridas e no tempo de deslocamento, e uso mais racional dos recursos e respostas mais ágeis às demandas urbanas.

Palavras-chave: Problema do Caixeiro Viajante. Serviço Público. Otimização. Google OR-Tools.

1 INTRODUÇÃO

Em meio às transformações climáticas globais e ao rápido desenvolvimento urbano, as cidades se configuram como sistemas complexos, marcados por interdependências entre sua estrutura, atividades e fluxos. Essas interações geram impactos encadeados que afetam a

* Autor, Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus Aracati (IFCE) Rodovia CE- 040, Km 137,1 s/n Aeroporto, CE – 62.800-000 – Aracati – CE – Brazil, francisca.jessica.oliveira01@aluno.ifce.edu.br .

** Orientador, Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus Aracati (IFCE) Rodovia CE-040, Km 137,1 s/n Aeroporto, CE – 62.800-000 – Aracati – CE – Brazil, diego.rocha@ifce.edu.br .

sustentabilidade urbana (LEMOS, 2010). Além de sofrerem os impactos diretos dessas mudanças, as cidades também desempenham um papel crucial na intensificação desse fenômeno. Dessa forma, torna-se fundamental que as estratégias de desenvolvimento urbano integrem práticas de preservação ambiental, com ênfase na adaptação das cidades às mudanças climáticas (CARTER et al., 2015).

Nesse contexto, destaca-se o papel da fiscalização urbana na harmonização entre o crescimento urbano e a preservação ambiental, que é essencial para assegurar que as práticas adotadas estejam alinhadas com a sustentabilidade e garantir não apenas a adaptação às mudanças climáticas, mas também o atendimento às crescentes demandas da população, ao considerar a complexa teia de interações que molda a sustentabilidade nas cidades.

Ao observar o município de Aracati-CE, percebe-se não apenas um exemplar microcosmo desse desafio global, mas também um ambiente propício para o desenvolvimento de soluções inovadoras. Aracati possui 1.229 km² e dados do IBGE revelam uma evolução demográfica, registrando, no censo de 2022, uma população de 75.113 habitantes, em comparação aos 69.159 de 2010 ((IBGE), 2022). Esse crescimento demanda uma abordagem estratégica, especialmente em relação à expansão e adequação das infraestruturas urbanas.

Dentro deste cenário, destaca-se o Instituto de Qualidade e Meio Ambiente do Aracati (IQUAMA), criado em 2017 com a Lei Complementar nº 06/2017 uma autarquia municipal comprometida com o objetivo de garantir o uso sustentável dos recursos naturais e a preservação do equilíbrio ecossistêmico local, promovendo a melhoria da qualidade de vida da população por meio de ações de fiscalização ambiental e urbana, licenciamento e monitoramento. Diante do desafio de conciliar o crescimento urbano e a preservação ambiental, a atuação do IQUAMA¹ ganha ainda mais relevância, tornando-se um elemento chave para assegurar que as mudanças na paisagem urbana se alinhem aos preceitos legais e ambientais.

No panorama da preservação ambiental e gestão urbana sustentável, a eficiência na gestão de recursos públicos refere-se à otimização da relação entre produtos/serviços gerados e insumos utilizados, além de envolver a utilização criteriosa dos recursos disponíveis, focando não apenas na eficácia do alcance de objetivos, mas também na execução eficiente das tarefas. Em contraste, a eficácia concentra-se em fazer a coisa certa, enquanto a efetividade avalia o impacto tangível das ações, considerando como a execução de um programa pode alterar efetivamente a realidade (CHIAVENATO, 2006). Na gestão eficiente de recursos públicos, a otimização da relação entre resultados e insumos destaca a importância de realizar as atividades de fiscalização de forma eficaz, garantindo que os recursos sejam empregados de maneira eficiente. Em um contexto urbano, isso implica não apenas no cumprimento de normas e regulamentações, mas também na busca por métodos eficazes para realizar inspeções, monitorar o cumprimento das leis urbanas e, consequentemente, promover mudanças tangíveis no ambiente urbano.

Atualmente, o setor de fiscalização urbana do município dispõe de um veículo para atender às demandas, o que representa um desafio de gerenciamento de recursos públicos para

¹ <<https://iquama.aracati.ce.gov.br>>

atendimento às demandas que chegam ao setor, de forma que a eficácia operacional deste setor está intrinsecamente ligada ao modo como as rotas de atendimento são planejadas e executadas.

A abordagem atual, embasada na experiência empírica dos funcionários do setor, carece da garantia de otimização. O planejamento das rotas, crucial para o atendimento eficiente das demandas, é realizado sem a aplicação de métodos científicos de otimização. Nesse contexto, a implementação de estratégias baseadas na resolução do Problema do Caixeiro Viajante (PCV) se apresenta como uma oportunidade para aprimorar a eficiência operacional.

O PCV, também conhecido como *Traveling Salesman Problem (TSP)* em inglês, é um desafio clássico na área de otimização combinatória. Ele consiste em determinar a rota mais eficiente para um viajante percorrer um conjunto de cidades exatamente uma vez e retornar à cidade de origem. O objetivo central é minimizar a distância total percorrida durante essa jornada, (FLOOD, 1956). Em essência, o PCV requer a identificação da ordem ideal para visitar as cidades, garantindo que a rota total seja a mais curta possível, apresentando-se como um problema na busca por soluções eficazes em logística e planejamento de rotas.

Desta forma, este trabalho se propõe a explorar a aplicação do PCV para otimizar a gestão de rotas no setor de Fiscalização Urbana. Para atingir esse objetivo, foi utilizada a ferramenta Google *Or-Tools*, uma suíte de software de código aberto que oferece um conjunto de ferramentas e algoritmos para resolver problemas de otimização, incluindo roteamento de veículos, fluxos, programação inteira e linear, e programação por restrições (GOOGLE, 2023a).

Essa abordagem visa não apenas reduzir custos operacionais, economizando combustível e tempo, mas também aumentar a eficácia na resposta às demandas da comunidade. Isso beneficia não apenas a administração pública, mas também o bem-estar social dos munícipes, promovendo um ambiente urbano mais propício para o crescimento sustentável e desenvolvimento urbano organizado.

O artigo segue uma estrutura composta por seis seções, com a parte introdutória sendo o capítulo 1. No capítulo 2, serão explorados os trabalhos relacionados relevantes à pesquisa. O capítulo 3 abordará em detalhes os principais conceitos relacionados ao PCV e suas aplicações. No capítulo 4, será apresentada a metodologia adotada neste estudo. O capítulo 5 se dedicará à implementação e resultados. Por fim, o capítulo 6 trará a apresentação da análise dos resultados obtidos e a discussão das descobertas da pesquisa.

2 TRABALHOS RELACIONADOS

Neste capítulo será apresentada uma seleção de trabalhos relacionados que abordam a otimização de rotas, empregando o PCV como uma ferramenta fundamental. Estes estudos foram selecionados devido à sua relevância para o problema deste presente estudo.

No trabalho de Silva (2021) é destacada a necessidade das empresas, como a Agência de Correios de Russas, Ceará, se adaptarem ao crescimento econômico e à diversificação do mercado, impulsionados pelo comércio eletrônico em constante expansão. Para se manterem competitivas, as empresas devem oferecer serviços de entrega eficientes, o que depende do

planejamento otimizado de rotas. Nesse contexto, o trabalho propôs a modelagem do problema de roteirização de entregas como um PCV e utilizou o conjunto de software Google *Or-Tools* para resolvê-lo com base em dados reais da Agência de Correios de Russas. Uma comparação entre as rotas geradas pela ferramenta e as rotas manuais dos carteiros revelou uma redução média de 23,16% nas distâncias totais percorridas pelas rotas otimizadas. Além disso, as métricas de atratividade visual indicaram que as rotas geradas eram igualmente eficazes ou até melhores em comparação com as rotas tradicionais. Entretanto, o estudo identificou a necessidade de expandir a metodologia para distritos convencionais com mais clientes e realizar uma comparação direta com as rotas do sistema de gerenciamento de tráfego (TMS) em uso nos Correios. O TMS é uma ferramenta proprietária usada para otimizar o planejamento de rotas de entrega, mas não está implantado em todas as unidades dos Correios, incluindo a unidade estudada neste trabalho.

No estudo de Lucindo (2021) é apresentado o contexto da Universidade Federal de Juiz de Fora, Minas Gerais, em que a expansão física da instituição devido a programas governamentais resultou em um aumento substancial nas áreas a serem mantidas, gerando uma crescente demanda por serviços de manutenção, como manutenção predial e elétrica. O setor de manutenção, no entanto, opera com um número fixo de funcionários e veículos para atender a essas demandas em constante crescimento. A logística de transporte era tradicionalmente planejada de maneira empírica, baseando-se na experiência e conhecimento dos funcionários, o que frequentemente levava a subutilização de recursos e custos operacionais desnecessariamente elevados. Como resposta a essa situação, a pesquisa propõe a implementação de uma nova política de atendimento às demandas de manutenção juntamente com a adoção de uma ferramenta de roteirização de veículos denominada VRP Spreadsheet Solver. A implementação dessa ferramenta visa otimizar o planejamento das rotas de atendimento, permitindo uma alocação mais eficiente dos recursos. Os resultados apresentaram reduções significativas nos custos mensais observados, variando de 17,86% a 66,27%, o que reflete em maior eficiência operacional e custos mais baixos.

O trabalho de Diniz e Almeida (2022) aborda a otimização das rotas de entrega de uma empresa de serviço postal em Londrina, Paraná, que atualmente enfrenta desafios como atrasos, baixa produtividade e custos operacionais elevados devido à organização manual e intuitiva das rotas. O propósito deste trabalho é aplicar estratégias de otimização de rotas, utilizando a biblioteca *Or-Tools*, por meio de três abordagens distintas: o PCV, o Problema de Roteirização de Veículos (PRV) e a utilização do agrupamento k-Means. O estudo envolveu a construção de um conjunto de dados com base nas distâncias entre os pontos de entrega, a aplicação das estratégias de otimização e a avaliação dos resultados em termos de melhoria na eficiência do serviço. Os resultados demonstram reduções variando de 21,38% a 30,21% nas distâncias percorridas nas entregas; embora os tempos de processamento dos códigos tenham sido relativamente elevados, o que deve ser considerado na implementação prática das estratégias.

No estudo de Junior, Santos e Fernandes (2017), é proposta uma metodologia para otimizar a roteirização das viaturas da Marinha do Brasil na região metropolitana do Rio de Janeiro, com o objetivo de reduzir os custos logísticos associados ao transporte de suprimentos entre o centro de distribuição e as Organizações Militares (OM). A justificativa do trabalho

baseia-se no fato de que, sem uma roteirização eficiente, as entregas são realizadas sob demanda, de maneira desorganizada e antieconômica, aumentando significativamente os gastos com combustível e tempo de operação. Para solucionar esse problema, a pesquisa adota uma abordagem baseada no Problema do Caixeiro Viajante (PCV) e na Teoria dos Grafos, utilizando clusterização para agrupar OM próximas e modelando matematicamente a melhor sequência de visitas. A metodologia incluiu a coleta de dados sobre localização e distâncias, a formação de *clusters* e a aplicação de um software de otimização para encontrar a rota mais eficiente. Os resultados indicam que a aplicação desse modelo permitiu uma significativa economia de recursos operacionais, reduzindo a quilometragem percorrida e, consequentemente, os custos com transporte. Além disso, a otimização proposta pode ser replicada para diferentes cenários logísticos dentro da Marinha e outras instituições, contribuindo para um planejamento mais estratégico e eficiente do uso das viaturas.

No estudo de Sämy (2019), é apresentada uma metodologia para otimizar as rotas de distribuição de água na Operação Carro-Pipa, realizada pelo Exército Brasileiro no Nordeste do Brasil. O objetivo do trabalho é reduzir os custos operacionais do programa, que ultrapassa R\$ 1 bilhão anual, através da melhoria na roteirização dos caminhões-pipa. A justificativa se baseia na identificação de ineficiências, onde muitas entregas eram feitas a partir de pontos mais distantes do que o necessário. A metodologia adotada envolve modelagem matemática baseada em Programação Linear Inteira, implementada em MATLAB, para encontrar soluções ótimas de roteirização, levando em conta restrições como capacidade de retirada e demanda dos pontos de abastecimento. O estudo avaliou diferentes cenários e demonstrou que a aplicação do modelo poderia gerar economias expressivas, reduzindo distâncias percorridas e otimizando o uso dos recursos. Em alguns casos, a economia foi superior a 20%, o que representa um impacto significativo para a operação. O estudo conclui que a ferramenta desenvolvida pode ser aplicada a diferentes regiões, contribuindo para a eficiência logística do programa.

3 REFERENCIAL TEÓRICO

Este capítulo oferece uma base teórica para a compreensão das temáticas centrais deste trabalho. Conceitos fundamentais, como PCV, serão apresentados. Além disso, serão vistas brevemente as abordagens de solução, com destaque para a eficácia da suíte de software *Or-Tools*.

3.1 Fiscalização Urbana

No contexto do município de Aracati, as atribuições do fiscal de obras e serviços públicos são regulamentadas pela Classificação Brasileira de Ocupações (CBO), sob o código nº 2545-05, conforme estabelecido pela Lei Complementar nº 009/2017. O fiscal desempenha diversas atividades, incluindo vistorias, lavratura de autos, exercício de poder de polícia administrativa, fiscalização do ordenamento urbano, diligências e auditoria de processos, abrangendo áreas

urbanas e rurais. Essas atribuições exigem planejamento e execução eficientes de rotas para atender às demandas de fiscalização.

No âmbito das atividades laborais que envolvem vistorias externas, destaca-se o atendimento às demandas da fiscalização urbana, provenientes da Ouvidoria Municipal, Ouvidoria e *e-mail* do IQUAMA, onde os munícipes podem protocolar diversas denúncias. Além do atendimento às denúncias, há o monitoramento rotativo de áreas da cidade para averiguar a conformidade da legislação urbanística. Adicionalmente, são realizadas visitas para verificar a conformidade das obras aprovadas com os alvarás expedidos, encerrando o processo de obtenção do Habite-se, que é um certificado que atesta que a construção foi concluída de acordo com as normas e regulamentos locais, sendo apta para ocupação e utilização. Outras visitas são realizadas em atendimento às requisições de numeração, onde o fiscal verifica as condições e a numeração adequada do imóvel solicitado.

Essas atividades exemplificam algumas das demandas motivadoras para os deslocamentos externos dos fiscais. Atualmente, a organização e execução dessas demandas são feitas de maneira empírica pela equipe de fiscalização, que planeja a rota de acordo com prioridades e setores da cidade.

3.2 Problema do Caixeiro Viajante

O PCV representa um desafio clássico em otimização combinatória, onde o objetivo é encontrar a rota mais eficiente para um vendedor visitar um conjunto de cidades exatamente uma vez, retornando ao ponto de partida. Dado o número de cidades n , o total de rotas possíveis que cobrem todas as cidades é expresso abaixo, Matai, Singh e Mittal (2010):

$$\frac{(n-1)!}{2} \quad (1)$$

No PCV, as cidades são representadas como nós em um grafo, com linhas conectando todas as duplas de nós. Cada linha possui uma distância associada. Um percurso que visita todas as cidades é chamado de *tour* ou ciclo Hamiltoniano.

A complexidade do PCV reside na busca pela sequência ideal de visitas às cidades para minimizar a distância total percorrida. Devido à complexidade combinatória, abordagens heurísticas ou métodos aproximativos são comumente empregados, já que encontrar a solução ótima é considerado um problema NP-difícil, carecendo de um método eficiente para resolver todas as instâncias em tempo polinomial, Dahiya e Sangwan (2018).

O PCV pode ser caracterizado em um grafo completo não direcionado $G = (V, E)$ se for simétrico ou em um grafo direcionado $G = (V, A)$ se for assimétrico. O conjunto de vértices $V = \{1, \dots, n\}$ representa as cidades a serem visitadas, enquanto as arestas $E_{ij} \in E$ e os arcos $A_{ij} \in A$ denotam as conexões e trajetos possíveis entre as cidades, respectivamente.

Uma matriz de custo $C = (c_{ij})$ é definida nas arestas E ou nos arcos A , e essa matriz deve obedecer à desigualdade triangular:

$$c_{ij} \leq c_{ik} + c_{kj}, \quad \forall k. \quad (2)$$

Essa desigualdade triangular é especialmente relevante em problemas planares, onde as cidades são representadas como pontos $P_i = (X_i, Y_i)$, e a distância euclidiana c_{ij} é calculada como:

$$c_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}. \quad (3)$$

Essa mesma desigualdade se aplica quando c_{ij} representa o comprimento de um caminho mais curto de i para j no grafo G , Matai, Singh e Mittal (2010).

3.2.1 Aplicações

Em Matai, Singh e Mittal (2010), vê-se que o PCV encontra uma variedade de aplicações práticas em diferentes setores, evidenciando a versatilidade do PCV em otimizar processos e encontrar soluções eficientes.

Uma aplicação direta surge no contexto da furação de Placas de Circuito Impresso, onde neste cenário, as cidades do problema correspondem aos furos que precisam ser feitos na placa. O objetivo é minimizar o tempo total de deslocamento da máquina de furação, tratando cada diâmetro de furo como uma instância separada do PCV.

Outra aplicação é encontrada na revisão de motores a turbina a gás de aeronaves, onde a problemática está em posicionar corretamente os conjuntos de pás guia do bocal em cada estágio da turbina. Modelando esse problema como um PCV, busca-se encontrar uma disposição ótima das pás para garantir um fluxo de gás uniforme e, assim, obter benefícios substanciais, como a redução de vibrações e o aumento da uniformidade do fluxo.

A cristalografia por raios-X também se beneficia do PCV. Em experimentos para analisar a estrutura de materiais cristalinos, um difratômetro de raios-X é usado para medir a intensidade de reflexões de raios-X de um cristal em várias posições. Embora a sequência das medições não afete o resultado do experimento, o tempo total necessário para o experimento depende da sequência escolhida. Isso conduz a uma formulação do problema como um PCV, onde a minimização do tempo total de posicionamento é o objetivo.

A aplicação no cabeamento de computadores aborda a conexão de componentes em uma placa de computador. Nesse contexto, a exigência é que não mais do que dois fios estejam conectados a cada pino. Isso se traduz em encontrar o caminho Hamiltoniano mais curto, considerando pontos de entrada e saída não especificados.

O setor de logística também se beneficia do PCV, especificamente no problema de picking em armazéns, que se refere à separação de mercadorias. A busca pela rota mais curta para coletar itens de um pedido com o tempo mínimo de coleta é formulada como um problema de PCV, onde os locais de armazenamento dos itens representam as cidades.

O roteamento de veículos apresenta outra aplicação interessante. Seja para a coleta diária de correspondências em caixas de correio ou para a entrega de mercadorias a clientes, o PCV pode ser empregado para determinar a quantidade mínima de caminhões necessários e otimizar o tempo total de coleta ou entrega.

Este problema possui duas variantes que serão melhor detalhadas a seguir, o PCV simétrico e assimétrico.

3.2.2 PCV Simétrico e Assimétrico

No PCV, é dado um conjunto de cidades a serem visitadas, $C = \{c_1, c_2, c_3, \dots, c_m\}$, onde a cada par de cidades (i, j) é atribuído um valor de distância (ou custo) d_{ij} . Como restrições, cada cidade deve ser visitada apenas uma vez, e o caixeiro deve iniciar e finalizar sua viagem na mesma cidade. Dessa forma, o PCV é dito simétrico se $d_{ij} = d_{ji}$, ou seja, se a distância entre as cidades i e j for a mesma, independentemente do sentido da rota adotada. Ele é assimétrico se $d_{ij} \neq d_{ji}$, ou seja, se a distância entre as cidades i e j variar de acordo com o sentido da rota adotada, Ottoni, Nepomuceno e Oliveira (2016).

Para o PCV assimétrico com m cidades, introduzimos variáveis binárias 0-1 para representar as arestas do percurso. Seja x_{ij} uma variável binária associada à aresta direcionada de i para j , definida como:

$$x_{ij} = \begin{cases} 1, & \text{se a aresta } i \rightarrow j \text{ está no percurso,} \\ 0, & \text{caso contrário.} \end{cases} \quad (4)$$

Dada a necessidade de cada nó ter exatamente uma aresta de entrada e uma de saída, o problema pode ser formulado da seguinte forma:

$$\min \sum_{j=1}^m \sum_{i=1}^m c_{ij} x_{ij} \quad (5)$$

$$\text{sujeito a } \sum_{j=1}^m x_{ij} = 1 \quad \forall i = 1, \dots, m \quad (6)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad \forall j = 1, \dots, m \quad (7)$$

$$\sum_{i \in K} \sum_{j \in K} x_{ij} \leq |K| - 1 \quad \forall K \subset \{1, \dots, m\}, K \neq \emptyset \quad (8)$$

A função objetivo (5) busca minimizar o custo total do percurso, considerando os pesos c_{ij} das arestas selecionadas. As restrições (6) e (7) garantem que cada cidade tenha exatamente uma aresta de saída e uma de entrada, respectivamente.

A restrição (8) evita a formação de sub-rotas desconectadas do trajeto principal. Ela garante que, para qualquer subconjunto K de cidades, o número de arestas dentro desse subconjunto seja no máximo $|K| - 1$, assegurando que o percurso do caixeiro viajante forme um único ciclo que passe por todas as cidades sem gerar ciclos menores isolados.

No PCV simétrico, considerando um conjunto com m cidades, a ideia é determinar a melhor rota que o caixeiro viajante deve seguir para visitar cada cidade exatamente uma vez e retornar ao ponto de origem. Para expressar essa situação matematicamente, usa-se variáveis binárias x_{ij} para representar se existe uma aresta entre a cidade i e a cidade j .

A formulação começa considerando todas as combinações possíveis de pares de cidades não direcionadas. Se tivermos m cidades, o número total de pares únicos que pode-se formar é dado por $m \cdot (m - 1)$. Isso ocorre porque, ao escolher a primeira cidade, há m opções, e ao

escolher a segunda cidade após a primeira ter sido escolhida, restam $(m - 1)$ opções distintas. Então, $m \cdot (m - 1)$ representa o número total de pares ordenados (onde a ordem importa) que podemos formar.

Entretanto, devido à natureza simétrica do PCV, onde a ordem das cidades na rota não importa, é necessário ajustar essa contagem. Considerando-se que o par (i, j) , é o mesmo que (j, i) no contexto do PCV simétrico, então há uma duplicação desnecessária de pares equivalentes.

Para corrigir essa duplicação, realiza-se a divisão por 2. Dividindo $m \cdot (m - 1)$ por 2, obtemos a fórmula $\frac{m \cdot (m - 1)}{2}$, que representa o número total de pares únicos que podem ser formados a partir de m elementos não direcionados no contexto do PCV simétrico. Dessa forma, o problema pode ser expresso como:

$$\min \quad \frac{1}{2} \sum_{j=1}^m \sum_{k \in J(j)} c_k x_k \quad (9)$$

$$\text{s.a.} \quad \sum_{k \in J(j)} x_k = 2 \quad \forall j = 1, \dots, m \quad (10)$$

$$\sum_{j \in E(k)} x_j \leq |k| - 1 \quad \forall k = 1, \dots, m \quad (11)$$

$$x_j \in \{0, 1\} \quad \forall j \in E(k) \quad \forall k = 1, \dots, m \quad (12)$$

$J(j)$ é o conjunto de todas as arestas não direcionadas conectadas ao nó j , e $E(K)$ é o subconjunto das arestas não direcionadas que conectam as cidades em qualquer subconjunto adequado e não vazio K de todas as cidades.

A equação 9 representa a função objetivo a ser minimizada, a Equação 10 refere-se à restrição de que cada vértice j deve ter exatamente dois arcos incidentes, a Equação 11 impõe restrições para eliminar sub-rotas, e a Equação 12 define que as variáveis x_j são binárias.

Embora o problema simétrico seja um caso específico do problema assimétrico, algoritmos desenvolvidos para o PCV assimétrico geralmente apresentam desempenho ruim quando aplicados ao PCV simétrico. Isso ocorre porque o PCV simétrico exige uma abordagem e formulação de solução distintas. Além disso, uma instância do PCV assimétrico pode ser transformada em uma instância simétrica com o dobro do número de nós, Hoffman et al. (2013).

3.3 Abordagens de solução

Diversos algoritmos, como busca tabu, tem sido explorados para oferecer soluções eficientes para o PCV, Sathya e Muthukumaravel (2015).

3.3.1 Busca Tabu

Uma meta-heurística é uma abordagem de otimização que busca encontrar soluções boas para problemas complexos, especialmente aqueles de grande escala ou combinatórios, sem garantir a solução ótima. Elas são técnicas gerais, aplicáveis a diversos tipos de problemas, e

funcionam explorando o espaço de soluções por meio de estratégias adaptativas. Essas técnicas são particularmente úteis quando a busca exata é computacionalmente impraticável.

A busca tabu é uma meta-heurística projetada para evitar que outros métodos fiquem presos em soluções localmente ótimas ao lidar com problemas de otimização. Ela utiliza estruturas de memória flexíveis, restrições tabu estratégicas e critérios de aspiração, bem como funções de memória com diferentes intervalos de tempo para intensificar e diversificar a busca. Essa abordagem tem sido eficaz na obtenção de soluções ótimas ou quase ótimas para uma variedade de problemas práticos, desde escalonamento até telecomunicações e reconhecimento de caracteres. A busca tabu pode ser integrada a outros métodos e é capaz de começar com uma implementação simples que pode ser aprimorada ao longo do tempo, Glover (1990).

O PCV é NP-difícil, o que significa que nenhum algoritmo conhecido garante resolver todas as instâncias de forma ótima em um tempo razoável. Heurísticas e meta-heurísticas, como busca tabu, buscam "boas" soluções em tempos razoáveis. Inicia com uma solução inicial (uma rota no PCV), busca a melhor solução em uma vizinhança definida e reinicia o processo com a melhor solução encontrada, Basu (2012).

3.3.2 *Or-Tools*

O *Google Operational Research Tools*, também conhecido como *Or-Tools*, desenvolvido pela *Google*, é uma suíte de software de código aberto projetada para resolver problemas de otimização combinatória. Dentre as principais características do *Or-Tools*, destacam-se dois solucionadores de programação por restrições (CP* e CP-SAT) e dois solucionadores de programação linear (Glop e PDLP). Além disso, a suíte inclui *wrapper* para solucionadores comerciais e de código aberto, abrangendo também problemas de programação inteira mista. O *Or-Tools* também disponibiliza algoritmos dedicados a problemas como empacotamento de bins, mochila, o PCV e o Problema de Roteamento de Veículos. Além de algoritmos para grafos, como caminhos mais curtos, fluxo de custo mínimo, fluxo máximo e atribuição linear de soma. Embora tenha sido originalmente escrito em C++, a suíte oferece *wrapper* em Python, C# e Java, Google (2023b).

A ferramenta ganhou todos os anos desde 2013 o MiniZinc Challenge, uma competição anual de solucionadores de programação por restrições em uma variedade de benchmarks. A competição ocorre todos os anos desde 2008, e os vencedores são anunciados na Conferência Internacional sobre Princípios e Práticas de Programação por Restrições Google (2023a).

Dentro do contexto do solucionador de roteamento, o "limite de pesquisa" refere-se a restrições ou condições que são aplicadas ao processo de busca de soluções. Esses limites são usados para controlar o tempo ou o número de iterações dedicadas à resolução do problema, evitando que o solucionador gaste um tempo excessivo em problemas complexos. Existem alguns limites de pesquisa disponíveis para ajustar o comportamento do solucionador. O *solution_limit* define o número máximo de soluções a serem geradas durante a pesquisa. Já o *time_limit.seconds* estabelece um limite em segundos para o tempo total gasto na pesquisa,

enquanto o *Ins_time_limit.seconds* determina o tempo máximo permitido para a pesquisa de conclusão em cada vizinho Google (2023c).

Já o *First_Solution_Strategy* é o método usado para encontrar a solução inicial de um problema. Nesse estágio, o solucionador faz a escolha do método que será utilizado para encontrar a solução inicial. Dentre as opções disponíveis, destaca-se a abordagem *AUTOMATIC*, que permite que o próprio solucionador determine a estratégia mais apropriada com base nas características do modelo em resolução.

Outras alternativas incluem a estratégia *PATH_CHEAPEST_ARC*, que conecta o nó inicial da rota ao nó que gera o segmento de trajeto mais econômico, expandindo a rota iterativamente. A opção *PATH_MOST_CONSTRAINED_ARC* é semelhante, mas utiliza um seletor de comparação para avaliar os arcos, priorizando aqueles mais restritos. A estratégia *EVALUATOR_STRATEGY*, por sua vez, assemelha-se à *PATH_CHEAPEST_ARC*, exceto pelo fato de que os custos dos arcos são avaliados através de uma função específica definida pelo usuário. Além dessas, há abordagens como *SAVINGS*, baseada no algoritmo de poupança de Clarke e Wright, e *SWEEP*, que utiliza o algoritmo de limpeza de Wren e Holliday. *CHRISTOFIDES* apresenta uma variante do algoritmo homônimo, estendendo uma rota até que não seja mais possível inserir nós. Outras estratégias incluem *ALL_UNPERFORMED*, que torna todos os nós inativos, e *BEST_INSERTION*, que cria uma solução inserindo iterativamente o nó mais barato na posição mais adequada, com base na função de custo global do modelo de roteamento.

As opções *PARALLEL_CHEAPEST_INSERTION*, *LOCAL_CHEAPEST_INSERTION*, *GLOBAL_CHEAPEST_ARC*, *LOCAL_CHEAPEST_ARC* e *FIRST_UNBOUND_MIN_VALUE* apresentam variações na abordagem de inserção de nós e na conexão de arcos, proporcionando diferentes formas de lidar com o problema.

Além dessas estratégias, é possível obter o status da pesquisa utilizando o método *status* do modelo de roteamento. Essa funcionalidade fornece informações sobre o andamento da resolução, indicando se o problema foi solucionado com sucesso, se atingiu um máximo local, se falhou ou se ultrapassou o limite de tempo estabelecido. Essas opções oferecem flexibilidade para adaptar a estratégia de solução de acordo com as características específicas do problema em questão.

Há também as opções de busca local. As metaheurísticas disponíveis incluem *AUTOMATIC*, que permite que o solucionador selecione a metaheurística automaticamente com base no modelo em questão, e *GREEDY_DESCENT*, que aceita a melhoria da vizinhança (redução de custos) até que um mínimo local seja atingido. Além disso, *GUIDED_LOCAL_SEARCH* utiliza busca local guiada para escapar do mínimo local, sendo considerada, em geral, a metaheurística mais eficiente para roteamento de veículos. *SIMULATED_ANNEALING* emprega a simulação de escape para evitar mínimos locais, enquanto *TABU_SEARCH* utiliza a busca Tabu para o mesmo propósito. Por fim, *GENERIC_TABU_SEARCH* utiliza a busca tabular do valor objetivo da solução para evitar mínimos locais.

Uma abordagem eficaz para otimizar a resolução de problemas complexos, como o Problema de Roteamento de Veículos, é a combinação de uma estratégia de solução inicial com

uma metaheurística. Estratégias de solução inicial, como *SAVINGS* ou *PATH_CHEAPEST_ARC*, geram uma solução inicial razoável, que pode ser refinada por uma metaheurística. Por outro lado, algoritmos genéticos começam com uma população inicial aleatória de soluções e iteram para melhorar a qualidade dessas soluções. Ao aplicar metaheurísticas como *GUIDED_LOCAL_SEARCH* ou *SIMULATED_ANNEALING*, o solucionador pode explorar mais amplamente o espaço de soluções, escapando de ótimos locais e possivelmente encontrando uma solução mais próxima do ótimo global.

3.3.3 Atratividade Visual

A atratividade visual em problemas de roteirização de veículos refere-se a características das rotas que as tornam mais intuitivas e aceitáveis para planejadores e operadores, facilitando sua adoção em contextos práticos. Embora não possua uma definição única e universalmente aceita, a literatura identifica esse conceito por meio de três propriedades fundamentais: compactidade, que assegura a proximidade entre os clientes dentro de uma mesma rota, evitando trajetos excessivamente dispersos; não sobreposição/não cruzamento, que reduz interseções entre diferentes trajetos, garantindo uma separação clara entre as rotas e minimizando conflitos operacionais; e baixa complexidade estrutural, que refere-se à simplicidade geométrica das rotas, evitando curvas acentuadas, mudanças bruscas de direção e trechos sinuosos que possam dificultar a navegação, Rossit et al. (2019).

Na prática, soluções visualmente organizadas são mais facilmente compreendidas e aceitas pelos motoristas, reduzindo o tempo necessário para instruções operacionais e facilitando a adaptação a mudanças inesperadas, como congestionamentos ou restrições viárias. Além disso, rotas compactas e bem distribuídas promovem a especialização dos motoristas em determinadas áreas geográficas, favorecendo a previsibilidade do tráfego e otimizando a eficiência logística. Estudos indicam que a integração de técnicas de agrupamento de dados ao processo de roteirização pode potencializar essas características, produzindo soluções naturalmente mais organizadas e minimizando a necessidade de ajustes manuais Rocha et al. (2022).

4 METODOLOGIA

Este capítulo descreve de forma detalhada o processo de coleta, armazenamento, tratamento, integração e implementação dos dados que são a base deste projeto. Além disso, apresenta os métodos de otimização de rotas adotados, as métricas utilizadas para avaliação dos resultados e a estratégia para a condução dos testes, utilizando as bibliotecas adequadas, com foco na solução de roteamento fornecida pelo Google *Or-Tools*.

4.1 Modelagem do problema

A modelagem do problema da fiscalização urbana no município de Aracati-CE considera as especificidades das atividades realizadas pelo IQUAMA, que utiliza um único veículo

para atender às demandas de fiscalização urbana e ambiental na cidade. O veículo é capaz de transportar todo o material necessário para a fiscalização, sem restrições de capacidade de carga. Nesse contexto, propõe-se a aplicação do Problema do Caixeiro Viajante (PCV), um modelo de otimização combinatória que visa determinar a rota mais eficiente para visitar os pontos de fiscalização, minimizando a distância e o tempo de deslocamento.

A modelagem do problema não leva em consideração restrições de capacidade ou horários específicos, já que o único veículo disponível é suficiente para atender às fiscalizações diárias, e não há exigência de horários agendados para as vistorias. A solução proposta busca otimizar o uso do veículo, a fim de garantir uma fiscalização mais eficiente e uma resposta mais rápida às demandas da população. A implementação da solução proposta é feita utilizando a linguagem de programação Python, a biblioteca Google *Or-Tools* em conjunto com o ambiente de desenvolvimento Google Colab, o que facilita o desenvolvimento sem a necessidade de infraestrutura de *hardware* dedicado.

4.2 Coleta, Armazenamento e Tratamento

A coleta de dados foi realizada ao longo de um período de 30 dias, com o objetivo de registrar informações sobre as atividades de fiscalização urbana. Durante esse período, foram registrados os endereços visitados e os trajetos realizados. Para garantir uma coleta precisa e eficiente, foi utilizado um aparelho celular Xiaomi Redmi Note 7, exclusivo para o registro dos dados. O aparelho, conectado à internet, executou uma aplicação de GPS responsável pela gravação das rotas percorridas durante as atividades de fiscalização. Ao iniciar cada jornada de trabalho, o aplicativo era acionado para registrar a rota do veículo. Ao término de cada percurso, as informações coletadas eram organizadas em uma planilha no Google Sheets, que incluía os endereços atendidos e suas coordenadas geográficas (latitude e longitude), verificadas e extraídas diretamente do Google Maps para evitar erros.

Além disso, os arquivos *KML* (Keyhole Markup Language), gerados pelo aplicativo de GPS e contendo os trajetos realizados, foram armazenados na plataforma Google Drive para futura análise de atratividade visual. O formato *KML* é amplamente utilizado para representar dados geográficos, como coordenadas e rotas, de forma estruturada e compatível com programas de visualização, como o *Google Earth*. Esses programas permitem observar os trajetos de maneira gráfica, facilitando a análise visual das rotas.

4.3 Implementação

4.3.1 Métricas

A etapa seguinte consistiu no processamento dos dados, iniciada com a elaboração de uma matriz de distâncias relativas às coordenadas geográficas (latitude e longitude) dos endereços atendidos por rota gravada. Essa matriz é formada por valores que indicam as distâncias entre todos os pontos vinculados aos endereços. Para realizar esse cálculo, é empregada a métrica

euclidiana, considerando que, após a preparação dos dados, os pontos são representados por coordenadas de latitude e longitude. Para facilitar esse processo, foi utilizado o serviço de Matriz de Distâncias da API do Google.

Com a matriz de distâncias preparada, é realizada a implementação do modelo de roteamento no ambiente *Google Colab*, utilizando a linguagem *Python* e a biblioteca *Or-Tools*. Inicialmente, é configurada a estrutura do problema, que inclui a definição do número de veículos (um único veículo) e do ponto de partida (a base da fiscalização).

Em seguida executam-se diversas iterações utilizando diferentes combinações de estratégias de solução inicial, *First_Solution_Strategy*, e heurísticas de busca local, *Local Search*, fornecidas pela biblioteca *Or-Tools*. Para isso, foram testadas todas as estratégias de solução inicial disponíveis e paralelamente, as heurísticas de busca local também foram aplicadas, o que permite explorar as combinações entre estratégias de solução inicial e meta-heurísticas para determinar a rota mais eficiente em termos de distância total percorrida.

Além da otimização da distância, também foi considerada a duração das rotas de fiscalização. Para isso, foi criada uma matriz de tempo que leva em conta as distâncias entre os pontos e o tempo de viagem de carro, desconsiderando o tempo em que o veículo pode estar parado (como em semáforos ou congestionamentos). Esse procedimento foi adotado para permitir uma análise mais eficiente quando comparada com o cenário real, já que o tempo em que o veículo fica parado entre a fiscalização de um endereço e outro pode variar significativamente, ao considerar o tipo de demanda a ser atendida. Para fins de análise comparativa, foi considerado apenas o tempo em que o veículo esteve em movimento. Além disso, o tempo computacional foi igualmente avaliado. Para cada combinação de estratégias de solução inicial e busca local, o custo computacional foi registrado, permitindo a análise da relação entre a redução de distância, tempo e custo computacional.

Os métodos e parâmetros do solucionador, como mencionado na Seção 3.4, englobam:

- *First_Solution_Strategy*:
 - *AUTOMATIC*
 - *PATH_CHEAPEST_ARC*
 - *PATH_MOST_CONSTRAINED_ARC*
 - *EVALUATOR_STRATEGY*
 - *SAVINGS*
 - *SWEEP*
 - *CHRISTOFIDES*
 - *ALL_UNPERFORMED*
 - *BEST_INSERTION*
 - *PARALLEL_CHEAPEST_INSERTION*

- *LOCAL_CHEAPEST_INSERTION*
- *GLOBAL_CHEAPEST_ARC*
- *LOCAL_CHEAPEST_ARC*
- *FIRST_UNBOUND_MIN_VALUE*
- *Local Search:*
 - *AUTOMATIC*
 - *GREEDY_DESCENT*
 - *GUIDED_LOCAL_SEARCH*
 - *SIMULATED_ANNEALING*
 - *TABU_SEARCH*
 - *GENERIC_TABU_SEARCH*

Os resultados de cada iteração foram armazenados e organizados em tabelas que apresentam a eficiência de cada estratégia. Os indicadores avaliados incluem a distância total, tempo de deslocamento, tempo total de execução e a melhoria percentual obtida em relação ao cenário real.

4.3.2 Estrutura

Neste ponto, é apresentada uma visão geral simplificada da implementação do modelo de roteamento, mostrando-se um pseudocódigo que descreve, de forma abstrata e organizada, os passos principais da implementação. Na sequência, é exibida a estrutura do código em bloco, detalhando as funções e interações principais. O código implementa uma solução para o PCV utilizando a API do Google Maps e a biblioteca *OR-Tools*. Inicialmente, são definidas as coordenadas das localidades de interesse, e a matriz de distâncias e tempos é construída a partir dos dados fornecidos pela API do Google Maps. Em seguida, o modelo de roteamento é configurado com diferentes estratégias de solução inicial e heurísticas de busca local, que são parametrizadas para encontrar a rota mais eficiente.

No modelo implementado, a maioria das configurações do *OR-Tools* foi mantida em seus valores padrão, *default*, garantindo que a solução fosse baseada no comportamento esperado da biblioteca. No entanto, algumas modificações foram realizadas para adaptar o modelo às necessidades específicas do problema. O tempo limite para a busca foi configurado manualmente para 10 segundos, evitando tempos de execução excessivos. Além disso, diferentes estratégias de solução inicial e meta-heurísticas de busca local foram testadas de forma explícita, permitindo uma comparação entre abordagens. Outra modificação relevante foi a criação de callbacks personalizados para distâncias e tempos com base em dados obtidos da API do Google Maps, garantindo maior realismo na modelagem. Por fim, o algoritmo calcula e exibe as métricas de

desempenho, como a distância total percorrida e o tempo de deslocamento, permitindo comparar diferentes abordagens para otimização logística.

Algorithm 1 Fluxo de Implementação

```

1: Inicializar o cliente Google Maps com a API Key
2: Definir as coordenadas das localizações de origem e destino
3: Obter a matriz de distâncias utilizando a API do Google Maps
4: Obter a matriz de tempos utilizando a API do Google Maps
5: Selecionar as estratégias iniciais e metaheurísticas para otimização
6: for all combinações de estratégias iniciais e metaheurísticas do
7:   Criar o modelo de roteamento e o gerenciador de índices
8:   Configurar os callbacks para cálculo de distâncias e tempos
9:   Adicionar restrições de tempo e capacidade ao modelo
10:  Definir parâmetros de busca: estratégia inicial e metaheurística
11:  Executar a resolução do problema com os parâmetros configurados
12:  if solução for encontrada then
13:    Calcular a rota otimizada, distância e tempo total
14:    Armazenar os resultados na estrutura de dados
15:  else
16:    Registrar que nenhuma solução viável foi encontrada
17:  end if
18: end for
19: Salvar os resultados em um arquivo CSV
20: Exibir as informações finais e o tempo total de execução
  
```

5 RESULTADOS

Nesta seção, são apresentados os resultados obtidos a partir da implementação da metodologia, organizados em tabelas que permitem a análise do desempenho de cada meta-heurística, *Local Search*, aplicada às diferentes soluções iniciais, *First_Solution_Strategy*, bem como seu impacto na otimização das rotas avaliadas.

As tabelas seguem um formato padronizado para apresentar os resultados da otimização das rotas analisadas. Cada tabela contém informações sobre a rota original, as estratégias de solução inicial aplicadas, as meta-heurísticas utilizadas na otimização e os resultados obtidos para diferentes configurações testadas.

Na parte superior da tabela, há duas células: “DISTÂNCIA (KM)” e “TEMPO (MIN)”. A primeira indica a extensão total da rota no cenário real, ou seja, antes da aplicação de qualquer método de otimização. A segunda representa a duração total da rota real em minutos. Esses valores servem como referência para a comparação com os resultados otimizados.

A tabela também apresenta a lista de estratégias de solução inicial, que determinam a configuração inicial da rota antes da aplicação das técnicas de otimização. Essas estratégias são utilizadas como ponto de partida para as meta-heurísticas, que então refinam a solução inicial em busca de uma rota mais eficiente, dentro do contexto da biblioteca utilizada neste trabalho.

Os métodos de busca local empregados na otimização são listados no campo correspondente, apresentando os resultados obtidos para cada combinação de estratégia de solução inicial e meta-heurística. Para cada configuração testada, são reportados quatro valores principais, sendo o primeiro referente à extensão da rota otimizada, em quilômetros que indica o comprimento total da rota após a otimização, o segundo é referente à redução percentual da distância que representa a diminuição da extensão da rota otimizada em relação à rota original, o terceiro é referente à duração da rota otimizada em minutos que indica o tempo total necessário para percorrer a rota após a otimização e o quarto é referente à redução percentual da duração da rota que representa a diminuição do tempo necessário para percorrer a rota otimizada em comparação com a rota original.

Além disso, cada tabela possui em destaque os resultados referentes à extensão da rota otimizada com maior redução e a última linha informa o tempo total de execução do algoritmo, o que evidencia o custo computacional do processo.

Na Tabela 1, os resultados indicam que a abordagem *GUIDED_LOCAL_SEARCH* obteve o melhor desempenho, reduzindo a distância percorrida em 3,92% e o tempo total de deslocamento em 7,59% para todos os métodos aplicados à solução inicial. Estratégias como *SIMULATED_ANNEALING*, *TABU_SEARCH* e *GENERIC_TABU_SEARCH* apresentaram desempenhos semelhantes, com redução média de 3,87% na distância e 6,75% no tempo. Já os métodos de solução inicial *EVALUATOR_STRATEGY* e *SWEEP* não geraram soluções viáveis, evidenciando limitações na sua aplicação ao problema em questão. Ademais, os métodos de solução inicial *SAVINGS*, *CHRISTOFIDES*, *ALL_UNPERFORMED*, *BEST_INSERTION*, *PARALLEL_CHEAPEST_INSERTION*, *LOCAL_CHEAPEST_INSERTION* e *GLOBAL_CHEAPEST_ARC* obtiveram valores, apresentando maior redução, independente da meta-heurística aplicada. Além disso, observa-se que o tempo total de execução do algoritmo foi de 400,57 segundos, refletindo o custo computacional envolvido na aplicação dessas técnicas.

FIRST SOLUTION STRATEGY															
107	DISTÂNCIA (KM)	>AUTOMATIC >PATH_CHEAPEST_ARC >PATH_MOST_CONSTRAINED_ARC >FIRST_UNBOUND_MIN_VALUE				>SAVINGS >CHRISTOFIDES >ALL_UNPERFORMED >BEST_INSERTION >PARALLEL_CHEAPEST_INSERTION >LOCAL_CHEAPEST_INSERTION >GLOBAL_CHEAPEST_ARC				>EVALUATOR_STRATEGY >SWEEP		>LOCAL_CHEAPEST_ARC			
		KM	%	MINUTOS	%	KM	%	MINUTOS	%			KM	%	MINUTOS	%
138,48		102,806	-3,92	127,99	-7,59	102,806	-3,92	127,88	-7,59	nan		102,806	-3,92	127,96	-7,59
		102,863	-3,87	129,13	-6,75	102,806	-3,92	127,88	-7,59	nan		102,806	-3,92	127,96	-7,59
		102,863	-3,87	129,13	-6,75	102,806	-3,92	127,88	-7,59	nan		102,806	-3,92	127,96	-7,59
		102,863	-3,87	129,13	-6,75	102,806	-3,92	127,88	-7,59	nan		102,806	-3,92	127,96	-7,59
		102,863	-3,87	129,13	-6,75	102,806	-3,92	127,88	-7,59	nan		102,863	-3,87	129,13	-6,75
		102,863	-3,87	129,13	-6,75	102,806	-3,92	127,88	-7,59	nan		102,863	-3,87	129,13	-6,75
Tempo de execução		400,57 s													

Tabela 1 - Rota 1

Os dados apresentados nas Tabelas 2, 3, 4, 5, 6 e 7 evidenciam uma convergência nos resultados das diferentes combinações de estratégias, onde todas as metodologias de *First_Solution_Strategy* levaram à mesma solução final, independentemente da meta-heurística utilizada. Essa consistência sugere que o problema modelado apresenta um ótimo global bem definido ou que a estrutura das rotas leva naturalmente a essa solução. Além disso, as abordagens de busca local, como *GUIDED_LOCAL_SEARCH*, *SIMULATED_ANNEALING* e *TABU_SEARCH* não impactaram a otimização, podendo inferir que a solução inicial já está muito próxima da ótima. Novamente, apenas os métodos *EVALUATOR_STRATEGY* e *SWEEP* não produziram resultados numéricos.

Ademais, é possível salientar que as tabelas 1, 5 e 6 apresentam resultados que obtiveram os menores percentuais de redução após a otimização, o que pode-se dar em razão do cenário específico em que ocorreram, abrangendo atendimento de demandas em diversas localidades que possuem poucas vias de acesso entre elas, não havendo assim muita margem para otimização.

DISTÂNCIA (KM)		FIRST SOLUTION STRATEGY				
		>AUTOMATIC >PATH_CHEAPEST_ARC >PATH_MOST_CONSTRAINED_ARC >SAVINGS >ALL_UNPERFORMED >BEST_INSERTION >PARALLEL_CHEAPEST_INSERTION >LOCAL_CHEAPEST_INSERTION >GLOBAL_CHEAPEST_ARC >FIRST_UNBOUND_MIN_VALUE >CHRISTOFIDES >LOCAL_CHEAPEST_ARC				>EVALUATOR_STRATEGY >SWEEP
TEMPO (MIN)		KM	%	MINUTOS	%	
14,72						
42,13						
LOCAL SEARCH	GUIDED_LOCAL_SEARCH	9,373	-36,32	28,4	-32,59	nan
	SIMULATED_ANNEALING	9,373	-36,32	28,4	-32,59	nan
	TABU_SEARCH	9,373	-36,32	28,4	-32,59	nan
	GENERIC_TABU_SEARCH	9,373	-36,32	28,4	-32,59	nan
	GREEDY_DESCENT	9,373	-36,32	28,4	-32,59	nan
	AUTOMATIC	9,373	-36,32	28,4	-32,59	nan
Tempo de execução		400,59 s				

Tabela 2 - Rota 2

DISTÂNCIA (KM)	58,34	FIRST SOLUTION STRATEGY				
		>AUTOMATIC >PATH_CHEAPEST_ARC >PATH_MOST_CONSTRAINED_ARC >SAVINGS >CHRISTOFIDES >ALL_UNPERFORMED >BEST_INSERTION >PARALLEL_CHEAPEST_INSERTION >LOCAL_CHEAPEST_INSERTION >GLOBAL_CHEAPEST_ARC >LOCAL_CHEAPEST_ARC >FIRST_UNBOUND_MIN_VALUE				>EVALUATOR_STRATEGY >SWEEP
TEMPO (MIN)	92,9	KM	%	MINUTOS	%	
LOCAL SEARCH	GUIDED_LOCAL_SEARCH	33,236	-43,03	57,63	-37,97	nan
	SIMULATED_ANNEALING	33,236	-43,03	57,63	-37,97	nan
	TABU_SEARCH	33,236	-43,03	57,63	-37,97	nan
	GENERIC_TABU_SEARCH	33,236	-43,03	57,63	-37,97	nan
	GREEDY_DESCENT	33,236	-43,03	57,63	-37,97	nan
	AUTOMATIC	33,236	-43,03	57,63	-37,97	nan
Tempo de execução		400,35 s				

Tabela 3 - Rota 3

DISTÂNCIA (KM)	9,3	FIRST SOLUTION STRATEGY				
		>AUTOMATIC >PATH_CHEAPEST_ARC >PATH_MOST_CONSTRAINED_ARC >SAVINGS >PARALLEL_CHEAPEST_INSERTION >GLOBAL_CHEAPEST_ARC >CHRISTOFIDES >ALL_UNPERFORMED >BEST_INSERTION >LOCAL_CHEAPEST_INSERTION >LOCAL_CHEAPEST_ARC >FIRST_UNBOUND_MIN_VALUE				>EVALUATOR_STRATEGY >SWEEP
TEMPO (MIN)	29,4	KM	%	MINUTOS	%	
LOCAL SEARCH	GUIDED_LOCAL_SEARCH	6,892	-25,89	21,7	-26,19	nan
	SIMULATED_ANNEALING	6,892	-25,89	21,7	-26,19	nan
	TABU_SEARCH	6,892	-25,89	21,7	-26,19	nan
	GENERIC_TABU_SEARCH	6,892	-25,89	21,7	-26,19	nan
	GREEDY_DESCENT	6,892	-25,89	21,7	-26,19	nan
	AUTOMATIC	6,892	-25,89	21,7	-26,19	nan
Tempo de execução		400,32 s				

Tabela 4 - Rota 8

DISTÂNCIA (KM)	31,65	FIRST SOLUTION STRATEGY				
		>AUTOMATIC >PATH_CHEAPEST_ARC >PATH_MOST_CONSTRAINED_ARC >SAVINGS >CHRISTOFIDES >ALL_UNPERFORMED >BEST_INSERTION >PARALLEL_CHEAPEST_INSERTION >LOCAL_CHEAPEST_INSERTION >GLOBAL_CHEAPEST_ARC >LOCAL_CHEAPEST_ARC >FIRST_UNBOUND_MIN_VALUE				>EVALUATOR_STRATEGY >SWEEP
TEMPO (MIN)	56,18	KM	%	MINUTOS	%	
LOCAL SEARCH	GUIDED_LOCAL_SEARCH	30,066	-5	52,13	-7,2	nan
	SIMULATED_ANNEALING	30,066	-5	52,13	-7,2	nan
	TABU_SEARCH	30,066	-5	52,13	-7,2	nan
	GENERIC_TABU_SEARCH	30,066	-5	52,13	-7,2	nan
	GREEDY_DESCENT	30,066	-5	52,13	-7,2	nan
	AUTOMATIC	30,066	-5	52,13	-7,2	nan
Tempo de execução		400,32 s				

Tabela 5 - Rota 5

DISTÂNCIA (KM)	69,21	FIRST SOLUTION STRATEGY				
		>AUTOMATIC >PATH_CHEAPEST_ARC >PATH_MOST_CONSTRAINED_ARC >SAVINGS >CHRISTOFIDES >ALL_UNPERFORMED >BEST_INSERTION >PARALLEL_CHEAPEST_INSERTION >LOCAL_CHEAPEST_INSERTION >GLOBAL_CHEAPEST_ARC >LOCAL_CHEAPEST_ARC >FIRST_UNBOUND_MIN_VALUE				>EVALUATOR_STRATEGY >SWEEP
TEMPO (MIN)	86,58	KM	%	MINUTOS	%	
LOCAL SEARCH	GUIDED_LOCAL_SEARCH	61,743	-10,8	81,98	-5,3	nan
	SIMULATED_ANNEALING	61,743	-10,8	81,98	-5,3	nan
	TABU_SEARCH	61,743	-10,8	81,98	-5,3	nan
	GENERIC_TABU_SEARCH	61,743	-10,8	81,98	-5,3	nan
	GREEDY_DESCENT	61,743	-10,8	81,98	-5,3	nan
	AUTOMATIC	61,743	-10,8	81,98	-5,3	nan
Tempo de execução		400,15 s				

Tabela 6 - Rota 6

DISTÂNCIA (KM)	18,15	FIRST SOLUTION STRATEGY				
		>AUTOMATIC >PATH_CHEAPEST_ARC >PATH_MOST_CONSTRAINED_ARC >SAVINGS >CHRISTOFIDES >ALL_UNPERFORMED >BEST_INSERTION >PARALLEL_CHEAPEST_INSERTION >LOCAL_CHEAPEST_INSERTION >GLOBAL_CHEAPEST_ARC >LOCAL_CHEAPEST_ARC >FIRST_UNBOUND_MIN_VALUE			>EVALUATOR_STRATEGY >SWEEP	
TEMPO (MIN)	27,4	KM	%	MINUTOS	%	
LOCAL SEARCH	GUIDED_LOCAL_SEARCH	12,044	-33,63	23,11	-15,69	nan
	SIMULATED_ANNEALING	12,044	-33,63	23,11	-15,69	nan
	TABU_SEARCH	12,044	-33,63	23,11	-15,69	nan
	GENERIC_TABU_SEARCH	12,044	-33,63	23,11	-15,69	nan
	GREEDY_DESCENT	12,044	-33,63	23,11	-15,69	nan
	AUTOMATIC	12,044	-33,63	23,11	-15,69	nan
Tempo de execução		400,18 s				

Tabela 7 - Rota 7

Na Tabela 8, dentre todas as rotas analisadas, observou-se a maior variação nos resultados. A maior redução foi alcançada com o uso da estratégia *PARALLEL_CHEAPEST_INSERTION* de *First_Solution_Strategy*, que obteve resultados idênticos para todas as metaheurísticas aplicadas.

		FIRST SOLUTION STRATEGY																														
DISTÂNCIA (KM)	TEMPO (MIN)	>AUTOMATIC >PATH_CHEAPEST_ARC >PATH_MOST_CONSTRAINED_ARC >CHRISTOFIDES >LOCAL_CHEAPEST_ARC						>EVALUATOR_STRATEGY >SWEEP >ALL_UNPERFORMED >BEST_INSERTION						>GLOBAL_CHEAPEST_ARC						>PARALLEL_CHEAPEST_INSERTION						>FIRST_UNBOUND_MIN_VALUE						
		KM	%	MINUTOS	%	KM	%	MINUTOS	%	KM	%	MINUTOS	%	KM	%	MINUTOS	%	KM	%	MINUTOS	%	KM	%	MINUTOS	%	KM	%	MINUTOS	%			
100	156.82	GUIDED_LOCAL_SEARCH	61,055	-38,95	105,31	-32,94	nan	nan	61,055	-38,95	105,31	-32,94	61,055	-38,95	105,31	-32,94	61,055	-38,95	104,83	-32,94	61,055	-38,95	104,83	-32,94	61,055	-38,95	105,31	-32,94	61,055	-38,95	105,31	-32,94
		SIMULATED_ANNEALING	61,917	-38,08	105,88	-32,94	nan	nan	61,917	-38,08	105,88	-32,94	61,792	-38,21	109,85	-29,96	61,055	-38,95	104,83	-32,94	61,055	-38,95	104,83	-32,94	61,306	-38,69	107,11	-31,69	61,306	-38,69	107,11	-31,69
		TABU_SEARCH	61,917	-38,08	105,88	-32,94	nan	nan	61,917	-38,08	105,88	-32,94	61,055	-38,95	105,31	-32,94	61,055	-38,95	104,83	-32,94	61,055	-38,95	104,83	-32,94	61,306	-38,69	107,11	-31,69	61,306	-38,69	107,11	-31,69
		GENERIC_TABU_SEARCH	61,792	-38,21	109,85	-29,95	nan	nan	61,917	-38,08	105,88	-32,94	61,055	-38,95	105,31	-32,94	61,055	-38,95	104,83	-32,94	61,055	-38,95	104,83	-32,94	61,306	-38,69	107,11	-31,69	61,306	-38,69	107,11	-31,69
		GREEDY_DESCENT	61,917	-38,08	105,88	-32,48	nan	nan	61,917	-38,08	105,88	-32,94	62,435	-32,57	106,46	-32,11	61,055	-38,95	104,83	-32,94	61,055	-38,95	104,83	-32,94	61,917	-38,08	105,88	-32,94	61,917	-38,08	105,88	-32,94
		AUTOMATIC	61,917	-38,08	105,88	-32,48	nan	nan	61,917	-38,08	105,88	-32,94	62,435	-32,57	106,46	-32,11	61,055	-38,95	104,83	-32,94	61,055	-38,95	104,83	-32,94	61,917	-38,08	105,88	-32,94	61,917	-38,08	105,88	-32,94
Tempo de execução		400,54 s																														

Tabela 8 - Rota 4

A partir da rota otimizada que obteve a maior redução de distância, a Rota 3, foi realizada uma análise da atratividade visual do percurso, que buscou avaliar não apenas a eficiência em termos de distância, mas também a qualidade visual. Na Figura 1, são apresentadas duas representações: a rota real (a) e a rota otimizada (b). Na imagem da rota real (a), é possível observar um trajeto com diversos cruzamentos, o que pode resultar em uma experiência menos fluida e mais demorada. Em contraste, na imagem da rota otimizada (b), nota-se um trajeto mais direto e simplificado, com menos interferências e uma redução significativa na distância total percorrida de 43,03%, o que representa um ganho considerável em termos de eficiência e tempo de deslocamento.

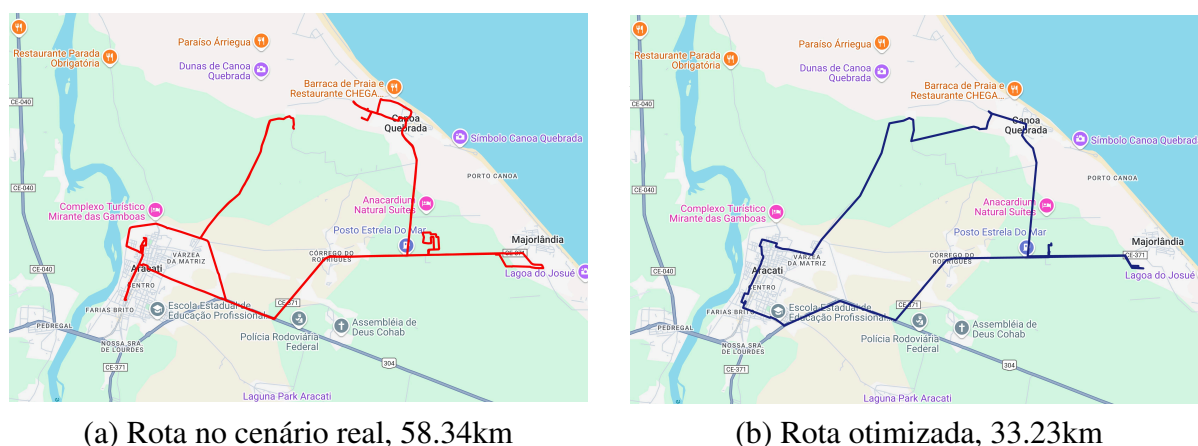


Figura 1 – Trajeto das rotas

6 CONCLUSÃO

Nas análises, é possível observar que, apesar das diferenças significativas entre as rotas, o impacto da otimização foi claramente visível em todas as situações. Contudo, a magnitude dessa otimização variou conforme a rota e as condições específicas do cenário de fiscalização. As situações mais interessantes ocorreram em rotas feitas por motoristas com pouca experiência na cidade, onde a otimização gerou uma redução considerável no tempo de percurso e no custo operacional, evidenciando a relevância da aplicação de estratégias eficientes nesses casos.

Por outro lado, rotas executadas em cenários que concentram o atendimento de demandas na sede, ou com pontos diversos de atendimento concentrados na sede e outros distritos, mais suscetíveis a melhorias significativas por meio da otimização. Isso pode ser atribuído ao fato de haver nas localidades que possuem malha viária mais desenvolvida e maior quantidade de vias o cenário ideal para otimizar os trajetos. Dessa forma, a otimização, embora ainda eficaz, teve uma redução mais modesta no desempenho em rotas de atendimentos de demandas não tão concentradas nas localidades em si, mas em pontos dispersos entre distritos distantes, que possuem entre si apenas uma ou duas vias de acesso, como BRs ou CEs, diminuindo assim a gama de opções de vias para otimizar o percurso.

Essa diferenciação nos resultados destaca a complexidade do problema de otimização de rotas de fiscalização urbana e a necessidade de uma abordagem personalizada, que leve em consideração as particularidades do contexto em que as rotas são realizadas.

Além disso, observou-se que as estratégias *EVALUATOR_STRATEGY* e *SWEEP* resultaram em valores *NaN* para todas as execuções. Isso pode ser explicado por características intrínsecas a essas abordagens e à natureza dos dados utilizados.

A primeira estratégia exige que o usuário defina explicitamente uma função de custo personalizada para avaliar os arcos da rota, o que não foi especificado no modelo implementado. Como consequência, o *OR-Tools* não possui um critério definido para calcular os custos dos deslocamentos, impossibilitando a geração de uma solução válida. A ausência de uma função de custo personalizada se deve à escolha metodológica de adotar abordagens padronizadas já implementadas no *OR-Tools*, garantindo reprodutibilidade e comparabilidade entre as estratégias testadas. Como o foco do estudo está na avaliação do desempenho de diferentes heurísticas para otimização de rotas, optou-se por utilizar as métricas de distância e tempo fornecidas diretamente pela API do Google Maps, sem definir um custo arbitrário específico.

Já a estratégia *SWEEP*, baseada no algoritmo de varredura, *sweep algorithm*, é tradicionalmente utilizado para o Problema de Roteamento de Veículos (PRV) e funciona organizando os nós em ordem crescente de ângulo polar a partir de um ponto central, normalmente o depósito ou ponto inicial da rota. No entanto, para que essa abordagem funcione corretamente, os pontos devem estar distribuídos de maneira favorável à heurística radial do método. Em contextos urbanos, onde os locais a serem visitados não seguem um padrão espacial regular, a aplicação do *SWEEP* pode ser limitada.

Uma consideração relevante a ser feita é que, embora a otimização tenha gerado reduções significativas nos tempos de deslocamento, é possível que, no cenário real, devido às especificidades operacionais da fiscalização, haja discrepâncias entre as rotas otimizadas e aquelas efetivamente executadas. Em situações em que a demanda de fiscalização não apresenta uma localização precisa — como no caso de denúncias de obras clandestinas, onde a informação disponível é imprecisa ou vaga, indicando apenas a proximidade de um local — é provável que a rota otimizada não seja totalmente aplicável. Nesses cenários, o fiscal, ao chegar à área indicada, pode ser obrigado a percorrer um trecho adicional não otimizado, como parte de seu trabalho investigativo, a fim de adotar as diligências necessárias para atender adequadamente à demanda. Assim, embora a otimização seja um instrumento de grande relevância, ela não elimina a necessidade de adaptação e flexibilidade durante a execução das atividades fiscais.

À vista do exposto sugere-se explorar novos cenários e possibilidades em trabalhos futuros, como alterações nos padrões *default*, adição de restrições como vias interditadas ou mesmo a inclusão de mais veículos na fiscalização, o que já mudaria a modelagem do problema.

REFERÊNCIAS

BASU, S. Tabu search implementation on traveling salesman problem and its variations: a literature survey. Scientific Research Publishing, 2012.

- CARTER, J. G. et al. Climate change and the city: Building capacity for urban adaptation. **Progress in planning**, Elsevier, v. 95, p. 1–66, 2015.
- CHIAVENATO, I. **Administração geral e pública**. [S.l.]: Elsevier Brasil, 2006.
- DAHIYA, C.; SANGWAN, S. Literature review on travelling salesman problem. **International Journal of Research**, v. 5, n. 16, p. 1152–1155, 2018.
- DINIZ, J. W. d. S.; ALMEIDA, M. F. d. **Otimização de rotas de entrega de uma empresa de serviço postal de Londrina**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2022.
- FLOOD, M. M. The traveling-salesman problem. **Operations research**, INFORMS, v. 4, n. 1, p. 61–75, 1956.
- GLOVER, F. Tabu search: A tutorial. **Interfaces**, INFORMS, v. 20, n. 4, p. 74–94, 1990.
- GOOGLE. **Google OR-Tools Documentation**. 2023. Accessed on: 20 11 2023. Disponível em: <<https://developers.google.com/optimization/>>.
- GOOGLE. **Google OR-Tools GitHub**. 2023. Accessed on: 27 11 2023. Disponível em: <<https://github.com/google/or-tools/blob/stable/README.md>>.
- GOOGLE. **Google OR-Tools GitHub**. 2023. Accessed on: 27 11 2023. Disponível em: <<https://developers.google.com/optimization/routing?hl=pt-br>>.
- HOFFMAN, K. L. et al. Traveling salesman problem. **Encyclopedia of operations research and management science**, Springer New York, v. 1, p. 1573–1578, 2013.
- (IBGE), I. B. de Geografia e E. **Panorama de Aracati no site do IBGE**. 2022. Disponível em: <<https://cidades.ibge.gov.br/brasil/ce/aracati/panorama>>.
- JUNIOR, L. R.; SANTOS, M.; FERNANDES, M. d. C. Aplicação do travelling salesman problem na roteirização das viaturas da marinha do brasil: uma abordagem da teoria dos grafos. **Anais do IX SIMPROD**, Departamento de Engenharia de Produção-Universidade Federal de Sergipe, 2017.
- LEMOES, M. F. R. C. **Adaptação de cidades para mudança climática: uma metodologia de análise para os planos diretores municipais**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 11 2010. Tese (doutorado) - Faculdade de Arquitetura e Urbanismo, Programa de Pós-Graduação em Urbanismo. Disponível em: <<http://objdig.ufrj.br/21/teses/774046.pdf>>.
- LUCINDO, H. C. **Roteirização de veículos no processo de atendimento às demandas de manutenção de uma universidade pública**. Tese (Doutorado) — PUC-Rio, 2021.
- MATAI, R.; SINGH, S. P.; MITTAL, M. L. Traveling salesman problem: an overview of applications, formulations, and solution approaches. **Traveling salesman problem, theory and applications**, InTech, Croatia, v. 1, n. 1, p. 1–25, 2010.
- OTTONI, A. L. C.; NEPOMUCENO, E. G.; OLIVEIRA, M. S. de. Aprendizado por reforço na solução do problema do caixeiro viajante assimétrico: Uma comparação entre os algoritmos q-learning e sarsa. **XII Simpósio de Mecânica Computacional**, 2016.
- ROCHA, D. et al. Visual attractiveness in vehicle routing via bi-objective optimization. **Computers & Operations Research**, Elsevier, v. 137, p. 105507, 2022.

ROSSIT, D. G. et al. Visual attractiveness in routing problems: A review. **Computers & Operations Research**, Elsevier, v. 103, p. 13–34, 2019.

SÄMY, D. B. C. Operação carro-pipa: otimização das rotas de distribuição de água. 2019.

SATHYA, N.; MUTHUKUMARAVEL, A. A review of the optimization algorithms on traveling salesman problem. **Indian Journal of Science and Technology**, v. 8, n. 29, p. 1–4, 2015.

SILVA Éolo Charles da. AnÁlise do uso da ferramenta or-tools para gerar rotas otimizadas na agÊncia dos correios de russas-ce tendo como base o problema do caixeiro viajante. 2021. Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Campus Aracati.