

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
CIÊNCIA DA COMPUTAÇÃO

Pablo Lopes Teixeira

**Avaliação de heurísticas para um problema do caixeiro viajante com veículos  
elétricos**

Florianópolis  
25 de novembro de 2025



Pablo Lopes Teixeira

**Avaliação de heurísticas para um problema do caixeiro viajante com  
veículos elétricos**

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Pedro Belin Castellucci,  
Dr.

Florianópolis  
25 de novembro de 2025

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.  
Dados inseridos pelo próprio autor.

Teixeira, Pablo Lopes

Avaliação de heurísticas para um problema do caixeiro viajante com veículos elétricos / Pablo Lopes Teixeira ; orientador, Pedro Belin Castellucci, 2025.

59 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro Tecnológico,  
Graduação em Ciências da Computação, Florianópolis, 2025.

Inclui referências.

1. Ciências da Computação. 2. Problema do Caixeiro Viajante. 3. Veículos Elétricos. 4. Janelas de Tempo. 5. Otimização. I. Castellucci, Pedro Belin. II. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. III. Título.

Pablo Lopes Teixeira  
**Avaliação de heurísticas para um problema do caixeiro viajante com veículos elétricos**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Graduação em Ciência da Computação.

Florianópolis, 25 de novembro de 2025.

---

Profa. Lúcia Helena Martins Pacheco, Dra  
Coordenadora do Curso

**Banca Examinadora:**

---

Prof. Pedro Belin Castellucci, Dr.  
Orientador  
Universidade Federal de Santa Catarina

---

Prof. Álvaro Junio Pereira Franco, Dr.  
Avaliador  
Universidade Federal de Santa Catarina

---

Prof. Rafael de Santiago, Dr.  
Avaliador  
Universidade Federal de Santa Catarina



## **AGRADECIMENTOS**

Aos meus pais, Paulo e Dulcineia, por serem minha base e meu porto seguro. Agradeço por todo o amor incondicional, pelos sacrifícios feitos para que eu pudesse estudar e por sempre acreditarem no meu potencial. Tudo o que sou e conquistei devo a vocês.

À minha namorada, Izabella, pelo companheirismo, carinho e paciência, especialmente durante os momentos mais intensos desta jornada. Obrigado por estar sempre ao meu lado, me incentivando a seguir em frente e celebrando cada pequena vitória.

Aos meus amigos de curso, pela parceria, pelos momentos de estudo compartilhados e pelas risadas que tornaram a caminhada na graduação mais leve. Agradeço pela amizade construída ao longo destes anos.

Ao meu orientador, Pedro, pela oportunidade, confiança e pela excelente condução deste trabalho. Agradeço pelos ensinamentos, pela paciência e por todo o conhecimento compartilhado, fundamentais para o meu crescimento acadêmico e profissional.

Por fim, agradeço o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, 405247/2023-0).



## LISTA DE FIGURAS

- Figura 1 – Representação de um único passo do SLS para o TSP utilizando 2-opt. Fonte: Hoos e Stützle (2005). . . . . 28
- Figura 2 – A rota inicia e termina no depósito (ícone de armazém), visitando clientes (casas) dentro de suas janelas de tempo e utilizando estações de recarga (ícones verdes) para manter a autonomia. Fonte: Autor. . . . . 40
- Figura 3 – Possíveis reconexões do operador 3-opt . . . . . 43



## LISTA DE TABELAS

- Tabela 1 – A tabela apresenta os resultados obtidos pela heurística de Busca Local, comparando os operadores 2-opt e 3-opt. Para cada operador, são exibidas a média e a mediana do custo da solução, do *gap*  $\left(\frac{\text{custo}-z^*}{z^*}\right)$  e do tempo computacional (em segundos). Em que  $z^*$  é a melhor solução encontrada por Roberti e Wen (2016) e cada grupo contém cinco instâncias. . . . . 47
- Tabela 2 – A tabela apresenta os resultados obtidos pela ILS, comparando os operadores 2-opt e 3-opt. Para cada operador, são exibidas a média e a mediana do custo da solução, do *gap*  $\left(\frac{\text{custo}-z^*}{z^*}\right)$  e do tempo computacional (em segundos). Em que  $z^*$  é a melhor solução encontrada por Roberti e Wen (2016) e cada grupo contém cinco instâncias. . . 48



## RESUMO

O problema do caixeiro viajante, clássico da otimização, busca determinar a rota mais curta para que um caixeiro visite um conjunto de cidades e retorne ao ponto de origem. No entanto, em um cenário mais moderno, essa questão ganha novas dimensões com o uso de veículos elétricos e a imposição de restrições de tempo para a visitação aos clientes, como no problema do caixeiro viajante com veículos elétricos e janelas de tempo (ETSPTW). Essa variante do problema não só leva em conta as limitações de autonomia das baterias, mas também a necessidade de recarga dos veículos e a organização das visitas dentro de horários específicos. Devido aos desafios envolvidos, soluções rápidas e eficientes são essenciais. Nesse contexto, este trabalho explora métodos heurísticos, especificamente a Busca Local e a Busca Local Iterada (ILS). A investigação avalia comparativamente os operadores de vizinhança 2-opt e 3-opt, e os resultados demonstram que o 3-opt, por possuir uma estrutura de vizinhança mais rica, é mais eficaz para encontrar soluções factíveis. Conclui-se que a combinação da ILS com o operador 3-opt gera as soluções de maior qualidade, apresentando uma abordagem robusta para cenários de transporte sustentável e logística moderna.

**Palavras-chave:** Problema do Caixeiro Viajante. Veículos Elétricos. Janelas de Tempo. Otimização.



## ABSTRACT

The Traveling Salesperson Problem, a classic in optimization, seeks to determine the shortest route for a salesperson to visit a set of cities and return to the point of origin. However, in a more modern scenario, this issue gains new dimensions with the use of electric vehicles and the imposition of time constraints for visiting customers, as seen in the Electric Traveling Salesperson Problem with Time Windows (ETSPTW). This variant of the problem takes into account not only battery range limitations but also the need for vehicle recharging and the scheduling of visits within specific time slots. Due to the challenges involved, fast and efficient solutions are essential. In this context, this work explores heuristic methods, specifically Local Search and Iterated Local Search (ILS). The investigation comparatively evaluates the 2-opt and 3-opt neighborhood operators, and the results demonstrate that 3-opt, due to its richer neighborhood structure, is more effective in finding feasible solutions. It is concluded that combining ILS with the 3-opt operator generates the highest quality solutions, presenting a robust approach for sustainable transport and modern logistics scenarios.

**Keywords:** Traveling Salesperson Problem. Electric Vehicles. Time Windows. Optimization.



## SUMÁRIO

	<b>Lista de Figuras</b> . . . . .	<b>7</b>
	<b>Lista de tabelas</b> . . . . .	<b>9</b>
<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>17</b>
1.1	OBJETIVOS . . . . .	18
1.1.1	<b>Objetivo Geral</b> . . . . .	<b>18</b>
1.1.2	<b>Objetivos Específicos</b> . . . . .	<b>18</b>
1.2	MÉTODO DE PESQUISA . . . . .	18
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>21</b>
2.1	OTIMIZAÇÃO COMBINATÓRIA . . . . .	22
2.2	MÉTODOS PARA SOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO: EXATOS E HEURÍSTICOS . . . . .	23
2.2.1	<b>Busca Perturbativa vs. Construtiva</b> . . . . .	<b>23</b>
2.2.2	<b>Busca Sistemática versus Busca Local</b> . . . . .	<b>24</b>
2.2.3	<b>Relação entre Busca Local e Busca Perturbativa</b> . . . . .	<b>24</b>
2.2.4	<b>Vantagens e Desvantagens da Busca Local</b> . . . . .	<b>25</b>
2.3	HEURÍSTICAS: BUSCA LOCAL, VIZINHANÇA, ÓTIMOS LOCAIS . . . . .	26
2.3.1	<b>Busca Local Estocástica (SLS)</b> . . . . .	<b>26</b>
2.3.2	<b>Vizinhanças e Grafos de Vizinhança</b> . . . . .	<b>27</b>
2.3.3	<b>Estratégias, Passos e Trajetórias de Busca</b> . . . . .	<b>27</b>
2.3.4	<b>SLS Não-Informada: Seleção Aleatória e Caminhada Aleatória</b> . . . . .	<b>29</b>
2.3.5	<b>Funções de Avaliação</b> . . . . .	<b>30</b>
2.3.6	<b>Melhoria Iterativa (<i>Iterative Improvement</i>)</b> . . . . .	<b>30</b>
2.3.7	<b>Mínimo Local</b> . . . . .	<b>31</b>
2.3.8	<b>Estratégias de Escape</b> . . . . .	<b>32</b>
2.3.9	<b>Intensificação versus Diversificação</b> . . . . .	<b>33</b>
2.4	COMENTÁRIOS FINAIS . . . . .	34

<b>3</b>	<b>TRABALHOS RELACIONADOS . . . . .</b>	<b>35</b>
3.1	O PROBLEMA DE ROTEAMENTO DE VEÍCULOS ELÉTRICOS COM MÚLTIPLOS VEÍCULOS . . . . .	35
3.2	GERAÇÃO DE SOLUÇÕES FACTÍVEIS PARA O ETSPTW . . . . .	36
3.3	FORMULAÇÃO E SOLUÇÃO HEURÍSTICA PARA O ETSPTW . . . . .	37
<b>4</b>	<b>DEFINIÇÃO DO PROBLEMA . . . . .</b>	<b>39</b>
<b>5</b>	<b>PROPOSTA DE SOLUÇÃO . . . . .</b>	<b>41</b>
<b>6</b>	<b>EXPERIMENTOS COMPUTACIONAIS . . . . .</b>	<b>45</b>
<b>7</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS . . . . .</b>	<b>49</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>51</b>

## 1 INTRODUÇÃO

O problema do caixeiro viajante (*Traveling Salesman Problem* - TSP) é um problema clássico da otimização combinatória e consiste em encontrar a rota mais curta que permita a um caixeiro viajante visitar um conjunto de cidades exatamente uma vez, retornando ao ponto de partida. A motivação por trás desse problema está relacionada ao desafio enfrentado por um caixeiro viajante que precisa visitar vários clientes localizados em diferentes cidades, buscando encontrar o menor percurso que complete essa tarefa. De forma mais geral e abstrata, o TSP pode ser formulado como: dado um grafo direcionado e ponderado nas arestas, encontrar o ciclo mais curto que visite cada nó do grafo exatamente uma vez (HOOS; STÜTZLE, 2005).

Uma das variantes mais recentes e desafiadoras do TSP é o problema do caixeiro viajante com veículo elétrico e janelas de tempo (*Electric Traveling Salesman Problem with Time Windows* - ETSPTW). Essa variante, também classificada como NP-Difícil, combina elementos de roteamento de veículos com restrições adicionais, como autonomia limitada das baterias e horários específicos para atender os clientes. Apesar de sua relevância prática, principalmente em cenários de transporte sustentável, e de seu potencial no contexto de métodos de decomposição, ainda há uma lacuna significativa no desenvolvimento de métodos eficientes para resolver essa variante (CASTELLUCCI; FRANCO; SANTIAGO, 2024).

Dada a dificuldade de resolver exatamente instâncias de tamanho intermediário ou grande, os métodos heurísticos surgem como uma alternativa eficaz. Diferente das abordagens exatas, que buscam garantir a solução ótima, os métodos heurísticos priorizam a obtenção de soluções de boa qualidade em tempos computacionais reduzidos. Entre os métodos heurísticos, destacam-se aqueles baseados em busca local, que operam explorando o espaço de soluções a partir de uma solução inicial, realizando pequenas modificações sucessivas até que não seja possível encontrar melhorias significativas.

Neste trabalho, serão avaliados métodos heurísticos baseados em busca local para resolver uma variante do ETSPTW. A pesquisa busca aprofundar os conceitos fundamentais, implementar e avaliar soluções, e analisar seus resultados em instâncias adaptadas da literatura, contribuindo tanto para o avanço do conhecimento acadêmico quanto para aplicações práticas na área de logística sustentável.

## 1.1 OBJETIVOS

Nesta seção estão apresentados os objetivos geral e específicos definidos para o trabalho.

### 1.1.1 Objetivo Geral

O objetivo geral deste trabalho é avaliar métodos heurísticos baseados em busca local para resolver uma variante do problema do caixeiro viajante que envolve veículos elétricos e janelas de tempo para atendimento aos clientes.

### 1.1.2 Objetivos Específicos

Visando atingir o objetivo geral, foram definidos os seguintes objetivos específicos.

- Descrever os conceitos relacionados a métodos heurísticos.
- Apresentar os fundamentos do problema do caixeiro viajante e suas variantes.
- Investigar métodos heurísticos aplicados ao problema do caixeiro viajante com veículos elétricos e janelas de tempo (ETSPTW).
- Desenvolver e implementar uma solução heurística para a variante do ETSPTW.
- Analisar e comparar os resultados obtidos utilizando instâncias adaptadas da literatura.
- Elaborar a monografia, documentando os métodos, implementações e resultados obtidos no trabalho.

## 1.2 MÉTODO DE PESQUISA

A metodologia adotada neste trabalho segue os passos a seguir: (i) Revisão da literatura; (ii) Desenvolvimento e proposição de estratégias de solução; (iii) Avaliação experimental das estratégias propostas; (iv) Elaboração de relatórios técnicos e da monografia final.

A revisão da literatura será conduzida de maneira abrangente, buscando identificar problemas similares, características relevantes do problema do caixeiro viajante e suas variantes, métodos heurísticos promissores e suas limitações. A revisão será mantida atualizada ao longo do desenvolvimento do trabalho, garantindo uma base sólida para as demais etapas.

A fase de desenvolvimento e proposição de estratégias de solução incluirá a formulação e implementação de métodos heurísticos baseados em busca local. Essa etapa será intercalada com a avaliação experimental para ajustar e aprimorar as estratégias. Durante esse processo, serão exploradas diferentes estruturas de vizinhança e abordagens de busca local para otimização do problema.

Na etapa de avaliação experimental, as soluções desenvolvidas serão comparadas com abordagens disponíveis na literatura, utilizando instâncias de *benchmark* amplamente conhecidas. Serão analisados critérios como a qualidade das soluções, o tempo computacional e a capacidade de lidar com as restrições específicas do problema. A avaliação será realizada de forma sistemática e detalhada, buscando caracterizar o desempenho das estratégias propostas.

Por fim, os resultados obtidos serão organizados em relatórios técnicos e na monografia final, com a possibilidade de elaboração de artigos científicos para divulgação dos principais achados do trabalho. Este processo busca garantir que o conhecimento gerado seja amplamente disseminado e contribua para o avanço na área de otimização logística sustentável.



## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo estabelece a base teórica para a avaliação de heurísticas aplicadas ao Problema do Caixeiro Viajante com Veículos Elétricos e Janelas de Tempo (ETSPTW). Dada a complexidade NP-Difícil do problema que torna até o momento busca por soluções ótimas inviável para instâncias de tamanho prático, os métodos heurísticos, em particular os algoritmos de busca local, emergem como uma abordagem adequada. O objetivo deste capítulo é, portanto, detalhar os princípios, as estratégias e os desafios fundamentais inerentes a essa classe de algoritmos.

A exposição inicia-se justificando a escolha da busca local em detrimento de métodos sistemáticos, destacando suas vantagens em cenários com restrições de tempo e a capacidade de fornecer soluções de alta qualidade rapidamente. Em seguida, o capítulo aprofunda-se no arcabouço da Busca Local Estocástica (SLS), definindo seus componentes essenciais, como espaços de busca, estruturas de vizinhança e funções de avaliação, que guiam o processo de otimização.

Com esses fundamentos estabelecidos, são apresentados os algoritmos e os conceitos-chave de forma progressiva. Partindo da estratégia básica de Melhoria Iterativa (*Iterative Improvement*), discute-se seu principal obstáculo: a estagnação em mínimos locais. A partir deste desafio, são exploradas as estratégias de escape, que permitem ao algoritmo superar ótimos locais e continuar a exploração do espaço de soluções. Finalmente, esses mecanismos são contextualizados dentro do dilema central do projeto de heurísticas de alto desempenho: o equilíbrio entre a intensificação, que refina soluções em regiões promissoras, e a diversificação, que garante uma exploração abrangente do espaço de busca.

Ao final deste capítulo, o leitor terá a compreensão necessária sobre como os algoritmos de busca local operam, quais são seus componentes, desafios e as estratégias utilizadas para superá-los. Esse conhecimento é crucial para a implementação, a análise e a avaliação das heurísticas propostas para o ETSPTW nas seções subsequentes deste trabalho.

## 2.1 OTIMIZAÇÃO COMBINATÓRIA

Problemas combinatoriais normalmente envolvem achar combinações, ordenações ou atribuições de um conjunto discreto e finito de objetos que satisfazem certas condições ou restrições. Por exemplo, encontrar uma ordem para visitar um determinado conjunto de clientes, respeitando o horário de disponibilidade de cada um. É frequente que, para problemas de otimização combinatoria, o espaço de busca tenha tamanho exponencial em relação ao tamanho da instância. Até o momento, isso acontece com problemas NP-difíceis. Vale ressaltar que muitos dos problemas combinatoriais são problemas de otimização e não problemas de decisão.

Normalmente, a abordagem mais natural para a maioria dos problemas de decisão e otimização combinatoria envolve a exploração do espaço de soluções candidatas. Um exemplo clássico é o problema do caixeiro-viajante (TSP) (JÜNGER; REINELT; RINALDI, 1995), em que, dado um grafo direcionado com arestas ponderadas, o objetivo é encontrar um ciclo Hamiltoniano com o caminho de menor peso em  $G$

Entretanto, como mencionado logo acima, é necessário ter atenção que, para um dado problema combinatorial, o espaço de soluções candidatas é muito grande, normalmente o exponencial do tamanho da instância do problema. Por exemplo, no caso de uma instância do problema do Caixeiro Viajante (TSP) com 20 cidades, o tamanho do conjunto de soluções candidatas seria de  $(20 - 1)! = 19! \approx 1,2 \cdot 10^{17}$  rotas possíveis. Para conseguir procurar por soluções em conjuntos tão grandes, é interessante utilizar algoritmos de busca local, que são empregados para resolver problemas combinatoriais muito difíceis em que não se conhece algoritmos eficientes (nesse caso eficiente significa ser resolvível em tempo polinomial).

Essa complexidade computacional evidencia que praticamente todas as abordagens computacionais para resolver problemas combinatoriais difíceis podem ser caracterizadas como algoritmos de busca. A ideia fundamental por trás desses algoritmos é gerar e avaliar soluções candidatas de forma iterativa. É importante observar que, mesmo problemas combinatoriais NP-Difícil possuindo uma complexidade temporal que cresce exponencialmente para encontrar soluções, a avaliação de uma solução viável normalmente pode ser realizada de maneira muito mais eficiente, em tempo polinomial (HOOS; STÜTZLE, 2005).

## 2.2 MÉTODOS PARA SOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO: EXATOS E HEURÍSTICOS

Para resolver problemas de otimização combinatória, como o ETSPTW, a literatura apresenta duas classes principais de abordagens: métodos exatos e métodos heurísticos. A escolha entre essas duas famílias representa um compromisso fundamental no campo da otimização, balanceando a garantia de encontrar a solução ótima e a viabilidade computacional para obter uma resposta em tempo hábil. Os métodos exatos são algoritmos que garantem encontrar a solução ótima global e provar sua otimalidade. Contudo, para problemas classificados como NP-Difíceis, como é o caso do ETSPTW, o custo computacional desses métodos geralmente cresce de forma exponencial com o tamanho da instância, tornando-os impraticáveis para a maioria dos cenários de aplicação real.

Em contrapartida, os métodos heurísticos abdicam da garantia de otimalidade em favor da eficiência. O objetivo de uma heurística é encontrar soluções de alta qualidade em um tempo de execução polinomial e, portanto, computacionalmente viável. Dada a complexidade do problema abordado, as heurísticas emergem como a abordagem mais pragmática e eficaz. Esta seção, baseando-se amplamente nos conceitos fundamentais apresentados por Hoos e Stützle (2005), irá aprofundar as características dessas duas classes de métodos, com foco especial nas subcategorias de busca que fundamentam o desenvolvimento da solução proposta neste trabalho. Serão exploradas as diferenças entre abordagens construtivas e perturbativas, bem como entre busca sistemática e busca local, estabelecendo a base teórica para justificar a adoção de uma heurística de busca local para o ETSPTW.

### 2.2.1 Busca Perturbativa vs. Construtiva

Em problemas combinatórios, existem dois tipos principais de métodos de busca: perturbativos e construtivos. Os métodos perturbativos começam com soluções candidatas completas e as modificam alterando componentes específicos. Já os métodos construtivos trabalham com soluções parciais, expandindo-as gradualmente até obter soluções completas (HOOS; STÜTZLE, 2005).

O espaço de busca dos métodos perturbativos consiste principalmente em so-

luções candidatas completas, enquanto os construtivos incluem soluções parciais. Um exemplo de método construtivo é a heurística *Nearest Neighbour* (HOOS; STÜTZLE, 2005) para o TSP, que constrói um caminho escolhendo iterativamente o vértice não visitado mais próximo. Embora esta heurística sozinha geralmente não produza soluções ótimas, é frequentemente combinada com métodos perturbativos para melhorar os resultados.

### 2.2.2 Busca Sistemática versus Busca Local

A busca sistemática e a busca local representam duas abordagens fundamentalmente diferentes para problemas de otimização. Os algoritmos de busca sistemática percorrem o espaço de busca de maneira completa, garantindo que uma solução será encontrada se existir, ou confirmando com certeza sua inexistência.

Em contraste, os algoritmos de busca local movem-se de uma localização atual para uma localização vizinha no espaço de busca, baseando-se apenas em conhecimento local e com um número relativamente pequeno de vizinhos. A busca local é tipicamente incompleta, sem garantias de encontrar soluções existentes ou determinar sua inexistência, e pode visitar a mesma localização múltiplas vezes (HOOS; STÜTZLE, 2005).

Algoritmos de busca local frequentemente ficam presos em regiões do espaço de busca, necessitando de mecanismos especiais como reinicialização ou passos de diversificação para escapar. A busca sistemática, apesar de garantir completude, geralmente enfrenta o problema da explosão combinatória em instâncias grandes, tornando-a computacionalmente inviável para muitos problemas práticos.

### 2.2.3 Relação entre Busca Local e Busca Perturbativa

Os métodos de busca local são frequentemente baseados em busca perturbativa, que modifica componentes de soluções candidatas completas para gerar novas soluções. No contexto do TSP, a busca perturbativa tipicamente altera *edges* em *tours* existentes. A busca perturbativa geralmente opera diretamente no espaço de soluções candidatas completas.

Métodos de busca local podem ser combinados eficientemente - como usar a heurística *Nearest Neighbour* para gerar pontos iniciais para algoritmos de busca local

perturbativa para o TSP. A *Ant Colony Optimisation* (DORIGO; CARO, 1999) representa um caso interessante onde a busca perturbativa incorpora elementos construtivos, executando uma ou mais buscas locais construtivas em cada etapa do processo.

Embora a busca perturbativa seja naturalmente associada a métodos de busca local, também pode fundamentar algoritmos de busca sistemática, ainda que atualmente não existam métodos sistemáticos perturbativos com desempenho competitivo em problemas combinatórios difíceis.

#### **2.2.4 Vantagens e Desvantagens da Busca Local**

Embora os algoritmos de busca local sejam incompletos, eles apresentam vantagens significativas em muitas situações práticas. Para problemas construtivos com instâncias sabidamente solúveis, como o TSP ou problemas de otimização, a capacidade de detectar a inexistência de soluções (principal vantagem da busca sistemática) torna-se irrelevante (HOOS; STÜTZLE, 2005).

Em cenários com restrições de tempo real, comuns em quase todos os domínios de aplicação, os algoritmos sistemáticos podem precisar ser interrompidos antes de encontrar uma solução, tornando-os efetivamente incompletos. Quando interrompidos, os algoritmos de busca local geralmente oferecem a melhor solução encontrada até o momento, enquanto muitos algoritmos sistemáticos podem não fornecer qualquer solução candidata.

Em geral, os métodos sistemáticos e locais são complementares, sendo a busca local frequentemente vantajosa quando se necessita de boas soluções em tempo limitado, quando se utiliza processamento paralelo ou quando o conhecimento sobre o domínio do problema é limitado.

As desvantagens da busca local incluem sua incompletude, que impede garantir que uma solução existente seja encontrada ou determinar com certeza a inexistência de soluções. Algoritmos de busca local frequentemente ficam presos em ótimos locais, necessitando de mecanismos especiais como reinicialização ou passos de diversificação para explorar outras regiões do espaço de busca.

Além disso, a qualidade das soluções obtidas por busca local geralmente depende do ponto de partida e da estratégia de exploração da vizinhança, tornando seu desempenho menos previsível comparado aos métodos sistemáticos. Em problemas onde

são necessárias soluções comprovadamente ótimas, a busca local normalmente não é adequada sem extensões ou hibridizações específicas.

## 2.3 HEURÍSTICAS: BUSCA LOCAL, VIZINHANÇA, ÓTIMOS LOCAIS

Dada a aplicabilidade dos métodos heurísticos para problemas como o ETSPTW, esta seção descreve a busca local, técnica utilizada na solução proposta. Essa classe de algoritmos é frequentemente empregada na otimização combinatória, operando sob o princípio de melhoria iterativa a partir de uma solução candidata. O sucesso e o comportamento de qualquer algoritmo de busca local são definidos por três pilares conceituais que serão o foco desta seção.

O primeiro pilar é a própria estratégia de busca local, o processo iterativo de se mover de uma solução para outra. O segundo, e talvez o mais crucial, é a definição de uma relação de vizinhança, que determina o conjunto de soluções "próximas" que podem ser alcançadas a partir da solução atual em um único passo. A escolha da vizinhança define a topologia do espaço de busca e impacta diretamente a eficácia do algoritmo. Finalmente, o terceiro pilar é o conceito de ótimo local — uma solução que é a melhor em sua vizinhança imediata, mas não necessariamente a melhor globalmente. A existência de ótimos locais representa o principal desafio da busca local, pois pode levar à estagnação prematura da busca. Nas subseções seguintes, estes conceitos serão detalhados, fornecendo a base teórica essencial para a construção e avaliação da heurística para o ETSPTW.

### 2.3.1 Busca Local Estocástica (SLS)

A Busca Local Estocástica (SLS) representa uma classe de algoritmos que utiliza escolhas aleatorizadas no processo de busca por soluções em problemas combinatórios. O funcionamento destes algoritmos pode ser descrito da seguinte forma.

O processo de busca ocorre no espaço de soluções candidatas, que pode incluir tanto soluções completas quanto parciais (estas últimas no contexto de algoritmos construtivos). A busca inicia-se com a seleção de uma solução candidata inicial e prossegue por meio de movimentos iterativos entre soluções vizinhas. As decisões em cada etapa são baseadas apenas em informações locais limitadas. A característica distintiva

dos algoritmos SLS é a aleatorização das decisões e da inicialização da busca. Adicionalmente, o processo pode utilizar memória para armazenar um número limitado de soluções visitadas recentemente (HOOS; STÜTZLE, 2005).

Os algoritmos SLS têm demonstrado notável eficácia na resolução de problemas combinatórios complexos, incluindo aplicações clássicas como o Algoritmo Lin-Kernighan (LIN; KERNIGHAN, 1973) para o Problema do Caixeiro Viajante, bem como métodos gerais como Algoritmos Evolutivos (BACK, 1996) e *Simulated Annealing* (KIRKPATRICK; GELATT JR; VECCHI, 1983).

### 2.3.2 Vizinhanças e Grafos de Vizinhança

A escolha adequada de uma relação de vizinhança é crucial para o desempenho de um algoritmo SLS, sendo frequentemente necessário definir essa relação de forma específica para cada problema. Apesar disso, existem tipos padrão de relações de vizinhança que fundamentam muitas aplicações bem-sucedidas de busca local estocástica. Um dos tipos mais utilizados é a vizinhança de k-trocas, onde duas soluções candidatas são consideradas vizinhas se, e somente se, diferirem em no máximo k componentes da solução.

Para o Problema do Caixeiro Viajante (TSP), embora seja possível definir uma vizinhança k-troca baseada na mudança de posições de vértices, descobriu-se que uma abordagem alternativa é superior: considerar as arestas como componentes da solução, onde duas rotas candidatas são vizinhas de k-troca se uma puder ser obtida da outra removendo-se no máximo k arestas e reconectando os *tours* parciais resultantes.

### 2.3.3 Estratégias, Passos e Trajetórias de Busca

Os três primeiros componentes de um algoritmo SLS (espaço de busca, conjunto de soluções e relação de vizinhança) são geralmente específicos ao problema sendo resolvido. Estes componentes fornecem a base para a solução de um problema usando busca local estocástica. Entretanto, a partir destes componentes, diversas estratégias de busca podem ser desenvolvidas através da definição de funções de inicialização e de passo.

Tais estratégias podem, em certa medida, ser independentes do espaço de busca,

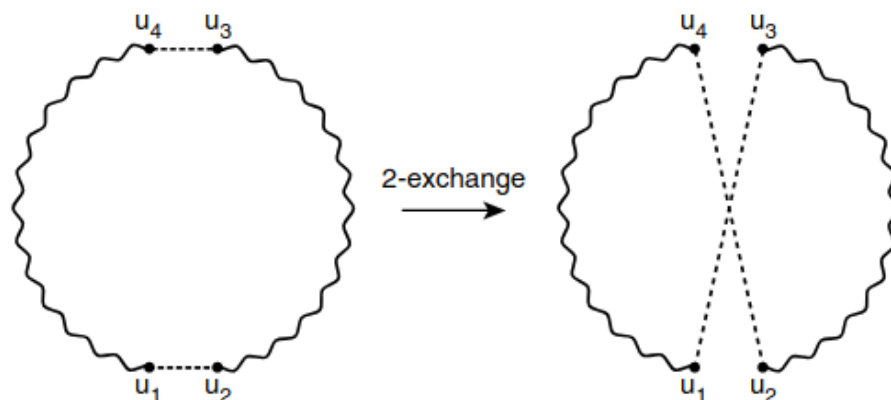


Figura 1 – Representação de um único passo do SLS para o TSP utilizando 2-opt. Fonte: Hoos e Stützle (2005).

conjunto de soluções e vizinhança subjacentes, podendo ser estudadas e apresentadas separadamente. Cada passo de busca representa um movimento entre soluções vizinhas, e uma trajetória de busca corresponde a uma sequência de soluções na qual cada par de elementos sucessivos são vizinhos. Em termos gerais, qualquer trajetória de busca corresponde a um percurso no grafo de vizinhança (HOOS; STÜTZLE, 2005).

Para ilustrar concretamente esses conceitos, podemos tomar como exemplo o operador de vizinhança 2-opt, amplamente utilizado em problemas como o TSP e suas variantes. Como pode ser visto na Figura 1, o 2-opt funciona removendo duas arestas não adjacentes de uma rota, dividindo-a em dois caminhos, e depois reconectando esses caminhos de uma nova maneira para formar um novo ciclo. A partir deste único operador, podemos definir diferentes funções de passo (*step function*) que resultam em estratégias de busca distintas:

- Movimento Aleatório: Escolher aleatoriamente duas arestas e aplicar a troca 2-opt, independentemente do resultado na qualidade da solução.
- Melhoria Iterativa Aleatória: Escolher aleatoriamente duas arestas e aplicar a troca 2-opt apenas se a nova rota for mais curta que a anterior.
- Primeira Melhoria (*First-Improvement*): Percorrer sistematicamente todos os pares de arestas possíveis e aplicar a troca 2-opt no primeiro par que resultar em

uma melhoria. A busca por um novo movimento recomeça a partir deste novo ponto.

- Melhor Melhoria (*Best-Improvement*): Percorrer sistematicamente todos os pares de arestas possíveis e aplicar a troca 2-opt no par que gerar a maior melhoria na qualidade da solução.

Note que cada uma dessas quatro abordagens define uma função de passo distinta, que por sua vez resulta em estratégias de busca com comportamentos completamente diferentes, variando de uma exploração não guiada a uma otimização intensiva e "gulosa". Essa flexibilidade na definição de funções de passo é fundamental para adaptar algoritmos SLS às características específicas de diferentes problemas, como as restrições de janelas de tempo no ETSPTW.

#### **2.3.4 SLS Não-Informada: Seleção Aleatória e Caminhada Aleatória**

As duas estratégias mais simples de Busca Local Estocástica são a Seleção Aleatória Não-Informada e a Caminhada Aleatória Não-Informada. Ambas não utilizam memória e baseiam-se em uma função de inicialização que retorna uma distribuição uniforme sobre todo o espaço de busca, selecionando qualquer elemento com igual probabilidade como ponto inicial (HOOS; STÜTZLE, 2005).

Na Seleção Aleatória Não-Informada, utiliza-se uma relação de vizinhança completa e a função de passo mapeia cada ponto para uma distribuição uniforme sobre todos os seus vizinhos (todo o espaço). Isso efetivamente amostra aleatoriamente o espaço de busca a cada passo.

Já a Caminhada Aleatória Não-Informada utiliza a mesma função de inicialização, mas para uma relação de vizinhança arbitrária, sua função de passo retorna uma distribuição uniforme apenas sobre o conjunto de vizinhos da solução atual. Para uma relação de vizinhança completa, esta estratégia coincide com a Seleção Aleatória.

Embora ineficazes isoladamente por não direcionarem a busca para soluções, ambas as estratégias são importantes quando combinadas com estratégias mais direcionadas, ajudando a prevenir ou superar estagnações prematuras em algoritmos SLS mais complexos e eficazes.

### 2.3.5 Funções de Avaliação

As funções de avaliação são mecanismos essenciais para direcionar a busca em direção às soluções, superando as estratégias SLS não-informadas. Para uma instância  $\pi$  de um problema de decisão, uma função de avaliação mapeia cada posição de busca para um número real, de modo que os ótimos globais correspondam às soluções do problema (HOOS; STÜTZLE, 2005).

Em algoritmos SLS para problemas de otimização combinatória, a função objetivo do problema frequentemente serve como função de avaliação. No entanto, em alguns casos, funções de avaliação diferentes podem fornecer orientação mais eficaz. Para problemas de decisão combinatórios, as funções de avaliação podem ser derivadas das funções objetivo de variantes de otimização. Através do uso de funções de avaliação, problemas de decisão e otimização podem ser tratados de maneira análoga. No entanto, para problemas de decisão, o resultado do algoritmo SLS só é útil se for um ótimo global. Já para problemas de otimização, soluções subótimas (geralmente mínimos locais) podem ser valiosas.

No caso do TSP, a função de avaliação é simplesmente o comprimento total de um percurso. Para qualquer rota candidata que visita todas as cidades, a função soma as distâncias entre cada cidade consecutiva para obter um custo total. Um algoritmo de busca usa esse custo para comparar diferentes percursos, preferindo sempre aqueles com menor distância, sendo assim guiado a encontrar a rota mais curta possível, que é a solução do problema.

### 2.3.6 Melhoria Iterativa (*Iterative Improvement*)

A Melhoria Iterativa é um dos algoritmos SLS básicos que utiliza função de avaliação. Dado um espaço de busca  $S$ , conjunto de soluções  $S'$ , relação de vizinhança  $N$  e função de avaliação  $g$ , o algoritmo inicia de um ponto aleatoriamente selecionado no espaço de busca e tenta melhorar a solução candidata atual em relação a  $g$  (HOOS; STÜTZLE, 2005).

A função de inicialização é tipicamente a mesma da Seleção Aleatória Não-Informada, atribuindo uma probabilidade uniforme a cada ponto do espaço, de modo que para toda solução  $s \in S$ , temos  $init(s) := 1/\#S$ . Para uma solução candidata  $s$ , se

$I(s)$  é o conjunto de todas as soluções vizinhas  $s' \in N(s)$  para as quais  $g(s') < g(s)$ , então a função de passo é definida como:

$$\text{step}(s)(s') := \begin{cases} 1/\#I(s) & \text{se } s' \in I(s) \\ 0 & \text{caso contrário} \end{cases} \quad (2.1)$$

Esta estratégia também é conhecida como descida iterativa ou *hill-climbing* (quando aplicada a problemas de maximização). O processo termina quando nenhum vizinho oferece melhoria.

Para implementação eficiente, os valores da função de avaliação são tipicamente mantidos usando atualizações incrementais (ou avaliações delta) após cada passo. No caso do TSP, com vizinhança 2-troca, rotas vizinhas  $p$  e  $p'$  diferem em duas arestas. O peso  $w(p')$  pode ser calculado incrementalmente subtraindo o peso das arestas contidas em  $p$  mas não em  $p'$  e adicionando o peso das arestas contidas em  $p'$  mas não em  $p$ . Isso reduz significativamente o número de operações aritméticas necessárias.

### 2.3.7 Mínimo Local

No contexto dos métodos de busca local, um conceito central é o de mínimo local. Uma solução candidata é considerada um mínimo local quando nenhuma solução em sua vizinhança imediata apresenta uma melhoria na função objetivo. Em outras palavras, a partir de um mínimo local, nenhum movimento único definido pela estrutura de vizinhança é capaz de gerar uma solução de melhor qualidade.

Essa característica representa o principal desafio para os algoritmos de melhoria iterativa. Ao encontrar um mínimo local que não seja a solução ótima global do problema, o algoritmo pode ficar preso, encerrando a busca prematuramente. Para o ETSP, isso significa que o algoritmo pode encontrar uma rota válida da qual nenhuma troca simples de clientes ou outra pequena modificação resulta em um percurso mais curto ou eficiente, mesmo que uma rota globalmente melhor exista.

Uma dificuldade inerente é que, para problemas de otimização combinatória complexos como o ETSP, geralmente não existem garantias teóricas sobre a qualidade de uma solução de mínimo local. Embora seja teoricamente possível garantir que todo mínimo local seja também um mínimo global utilizando vizinhanças completas,

o tamanho exponencial dessas vizinhanças as torna computacionalmente impraticáveis (HOOS; STÜTZLE, 2005).

No entanto, como apontado por Hoos e Stützle (2005), a evidência empírica demonstra que, para instâncias típicas de problemas de otimização, é possível encontrar mínimos locais de alta qualidade de maneira eficiente por meio de algoritmos de busca local bem projetados. Portanto, a eficácia de uma abordagem heurística está diretamente ligada à qualidade dos mínimos locais que ela é capaz de encontrar.

### 2.3.8 Estratégias de Escape

A constatação de que algoritmos de busca local podem estagnar em mínimos locais de qualidade insuficiente é um ponto central no projeto de heurísticas eficazes. Para contornar essa limitação, foram desenvolvidas estratégias de escape, mecanismos cujo objetivo é permitir que a busca prossiga para outras regiões do espaço de soluções, evitando a estagnação prematura. Conforme discutido por Hoos e Stützle (2005), a seguir são apresentadas duas abordagens fundamentais para escapar de mínimos locais.

Uma das estratégias mais diretas consiste em reiniciar a busca sempre que um mínimo local é alcançado. Nesse método, o algoritmo simplesmente descarta a solução atual e gera uma nova solução inicial, a partir da qual o processo de melhoria iterativa recomeça. Embora simples, essa abordagem pode ser ineficaz se o número de mínimos locais for muito grande ou se o custo computacional para inicializar uma nova solução for elevado (HOOS; STÜTZLE, 2005).

Uma alternativa é flexibilizar o critério de aceitação de movimentos. Em vez de parar, ao atingir um mínimo local, o algoritmo pode executar um movimento de não-melhora, ou seja, aceitar uma solução vizinha de qualidade igual ou inferior. Isso pode ser feito de forma aleatória, escolhendo qualquer rota vizinha (caracterizando uma caminhada aleatória), ou de maneira mais criteriosa, selecionando o movimento que cause a menor piora possível na função objetivo (uma abordagem conhecida como *mildest ascent* ou "subida mais suave"). A intenção é perturbar a solução atual o suficiente para sair da bacia de atração do mínimo local e encontrar um novo caminho para um ótimo de melhor qualidade.

Entretanto, não há garantias de que essas estratégias simples sejam sempre eficazes. É possível que, após um passo de escape, o único movimento de melhora

disponível leve o algoritmo de volta ao mesmo mínimo local do qual acabou de sair. Além disso, a busca pode encontrar platôs: regiões do espaço de busca onde múltiplas rotas vizinhas possuem o mesmo valor de função objetivo. Nesses platôs, o algoritmo pode realizar vários movimentos sem obter melhora, tornando difícil determinar se a busca está progredindo ou simplesmente vagando por uma região sem saídas promissoras (HOOS; STÜTZLE, 2005). No contexto do ETSPTW, um platô pode ocorrer quando diferentes sequências de visitação de clientes resultam no mesmo tempo total de viagem, mas com configurações de janelas de tempo distintas.

### **2.3.9 Intensificação *versus* Diversificação**

O desempenho e a robustez dos algoritmos de busca local estão intimamente ligados a um dilema fundamental em seu projeto: o equilíbrio entre intensificação e diversificação. Esses dois conceitos representam forças opostas que, quando combinadas de forma eficaz, permitem que a busca explore o espaço de soluções de maneira inteligente e abrangente (HOOS; STÜTZLE, 2005).

A intensificação refere-se a estratégias que focam em explorar intensivamente uma região promissora do espaço de busca. O objetivo é refinar uma solução existente para encontrar o melhor resultado possível naquela vizinhança. No contexto do ETSPTW, uma estratégia de intensificação consistiria em, a partir de uma rota inicial, realizar uma série de pequenas melhorias, como trocas de clientes adjacentes, para reduzir ao máximo seu custo total. A própria abordagem de melhoria iterativa, discutida anteriormente, é um exemplo clássico de intensificação.

Por outro lado, a diversificação visa garantir que o algoritmo não fique permanentemente preso em uma única região do espaço de busca, especialmente se essa região não contiver a solução ótima. As estratégias de diversificação forçam a busca a se mover para áreas ainda não exploradas. Para o problema em questão, isso poderia significar realizar uma grande perturbação na rota atual (por exemplo, invertendo uma longa sequência de visitas) ou reiniciar a busca a partir de um ponto completamente novo, como visto nas estratégias de escape. O objetivo é levar a busca para tipos de rotas fundamentalmente diferentes, mesmo que isso implique uma piora temporária na solução.

Conforme destacado por Hoos e Stützle (2005), o sucesso de uma heurística

de busca local avançada raramente vem de uma abordagem pura. A melhoria iterativa (intensificação pura) pode estagnar rapidamente, enquanto uma caminhada aleatória (diversificação pura) não explora o conhecimento adquirido. A chave para o alto desempenho reside em combinar e equilibrar essas duas forças de forma dinâmica e contextualizada.

É importante notar que encontrar o balanço ideal entre intensificação e diversificação muitas vezes depende mais da intuição e da experimentação do que de princípios teóricos rígidos.

## 2.4 COMENTÁRIOS FINAIS

A fundamentação teórica apresentada neste capítulo estabelece os conceitos necessários para o desenvolvimento deste trabalho. A exploração dos métodos de otimização, desde os exatos até os heurísticos, permitiu justificar de maneira formal e técnica a escolha de uma heurística de busca local como a abordagem mais pragmática e adequada para a natureza NP-Difícil do ETSPTW. Sem essa base, a seleção do método seria arbitrária e carente de rigor científico.

Os conceitos detalhados nesta seção são importantes porque eles formam a espinha dorsal tanto da implementação quanto da análise dos resultados. O entendimento de busca local, vizinhança e função de avaliação é diretamente aplicado na formulação da proposta de solução (Capítulo 4): a escolha do operador 3-opt é uma decisão sobre a estrutura de vizinhança, e a função de avaliação é o mecanismo que traduzirá as complexas restrições do ETSPTW em um guia numérico para a busca. Adicionalmente, a compreensão do fenômeno dos mínimos locais e das estratégias de escape é crucial para antecipar os desafios do algoritmo e interpretar seus resultados. Finalmente, o dilema entre intensificação e diversificação servirá como um guia analítico para avaliar o comportamento da heurística, permitindo diagnosticar se a busca está explorando novas regiões do espaço de soluções de forma eficiente ou se está estagnada.

### 3 TRABALHOS RELACIONADOS

A resolução do Problema do Caixeiro Viajante com Veículos Elétricos e Janelas de Tempo (ETSPTW) se apoia em contribuições cruciais da literatura recente. Este capítulo revisa os trabalhos que constituem o fundamento do presente estudo, com especial atenção a duas contribuições centrais: o trabalho seminal de Roberti e Wen (2016) e o estudo de Castellucci, Franco e Santiago (2024).

O primeiro é fundamental por ter sido pioneiro na formalização desta variante específica do ETSPTW e por ter proposto um conjunto de instâncias para *benchmark*, o qual é adotado nos experimentos deste trabalho. Por sua vez, a heurística de busca local proposta por Castellucci, Franco e Santiago (2024) constitui a base metodológica sobre a qual a presente investigação foi desenvolvida, motivando a exploração e a comparação aprofundada dos operadores de vizinhança. Ao examinar estas e outras contribuições, o capítulo contextualiza o estado da arte e justifica a relevância da abordagem proposta.

#### 3.1 O PROBLEMA DE ROTEAMENTO DE VEÍCULOS ELÉTRICOS COM MÚLTIPLOS VEÍCULOS

O artigo de Kucukoglu, Dewil e Cattrysse (2021) introduziu e formalizou o Problema de Roteamento de Veículos Elétricos com Janelas de Tempo (E-VRPTW), que se tornou uma referência na literatura. Diferente do E-TSPTW abordado nesta monografia, o E-VRPTW lida com o desafio de otimizar uma frota de múltiplos veículos elétricos para atender às demandas de um conjunto de clientes, respeitando suas respectivas janelas de tempo, a capacidade de carga dos veículos e, crucialmente, a autonomia limitada das baterias.

Para solucionar este complexo problema, os autores propuseram uma sofisticada meta-heurística híbrida, que combina a Busca em Vizinhança Variável (VNS) com a Busca Tabu (TS). A VNS é utilizada para diversificar a busca, explorando diferentes estruturas de vizinhança, enquanto a Busca Tabu atua como um mecanismo de intensificação, realizando uma exploração mais fina do espaço de soluções e evitando ciclos. Uma das contribuições mais duradouras deste artigo foi a criação e disponibilização de um conjunto de instâncias de *benchmark*, que se tornaram o padrão para a avaliação de novos algoritmos para o E-VRPTW.

A relevância do trabalho de Kucukoglu, Dewil e Cattrysse (2021) para esta monografia é fundamental para a contextualização do problema. Ao definir e resolver o E-VRPTW, eles estabeleceram a base para a análise de problemas mais específicos, como o E-TSPTW, que pode ser visto como um caso particular do E-VRPTW para um único veículo e sem restrições de capacidade de carga. O trabalho deles evidencia a complexidade inerente ao roteamento elétrico e a necessidade de métodos heurísticos poderosos para a obtenção de soluções de alta qualidade em tempo hábil, o que justifica a nossa abordagem de focar na avaliação detalhada de operadores de busca local para uma variante fundamental deste problema.

### 3.2 GERAÇÃO DE SOLUÇÕES FACTÍVEIS PARA O ETSPTW

O artigo de Castellucci, Franco e Santiago (2024) aborda o ETSPTW. Os autores destacam que, apesar da crescente atenção aos problemas de roteamento com veículos elétricos na literatura de otimização, o ETSPTW permanece como uma variante menos explorada. Um dos principais gargalos para a resolução deste problema, especialmente por meio de métodos exatos como *branch-and-cut*, é a dificuldade intrínseca de se encontrar uma solução inicial factível. Uma solução é considerada factível somente quando respeita simultaneamente todas as restrições do problema, que incluem:

- A sequência de visitas: Cada cliente deve ser visitado exatamente uma vez.
- Janelas de tempo: O atendimento a cada cliente  $i$  deve ocorrer dentro de seu intervalo de tempo  $[e_i, l_i]$ .
- Autonomia da bateria: O veículo não pode percorrer um arco se a carga da bateria for insuficiente.
- Recarga: O veículo pode visitar estações de recarga, onde a bateria é totalmente recarregada.

Para atacar o desafio da geração de soluções factíveis, Castellucci, Franco e Santiago (2024) propõem uma heurística de busca local simples, mas eficaz. O método parte de uma permutação aleatória dos clientes e estações de recarga e aplica iterativamente um operador de vizinhança 2-opt. A principal inovação do trabalho reside na

sua função de avaliação (função *value*), que não visa minimizar a distância percorrida (o objetivo clássico do TSP), mas sim minimizar a infactibilidade da rota. A função de valor é definida como uma soma ponderada das penalidades por violações, que incluem a penalidade de bateria, calculada sempre que a carga do veículo se torna negativa em um trecho, e a penalidade de janela de tempo, aplicada quando a chegada a um cliente ocorre após o limite superior de sua janela de tempo.

O algoritmo busca, portanto, uma rota com valor de penalidade igual a zero, que, por definição, é uma solução factível. A eficácia desta abordagem foi demonstrada experimentalmente nas instâncias de *benchmark* propostas por Roberti e Wen (2016). A heurística conseguiu encontrar soluções factíveis para 49 das 50 instâncias testadas, um resultado expressivo que valida o método como uma ferramenta poderosa para inicializar algoritmos mais complexos.

### 3.3 FORMULAÇÃO E SOLUÇÃO HEURÍSTICA PARA O ETSPTW

O trabalho Roberti e Wen (2016) propõem dois modelos de programação linear inteira mista (MILP) para o ETSPTW: um modelo compacto e um modelo alternativo com um número exponencial de variáveis, baseado em "caminhos de recarga", e os autores desenvolvem uma Heurística de Três Fases (3P-Heu), um algoritmo sofisticado projetado para encontrar soluções de alta qualidade em tempo computacional reduzido. A heurística se desdobra em três etapas sequenciais. A fase 1 (Viabilidade de Janela de Tempo) utiliza uma Descida em Vizinhança Variável (VND) para ajustar a rota inicial até que ela se torne factível em relação às janelas de tempo. Uma vez que a rota é viável nesse aspecto, a fase 2 (Otimização do Custo) aplica outra busca local VND para minimizar a distância percorrida. Finalmente, a fase 3 (Viabilidade de Bateria) executa um algoritmo de programação dinâmica para inserir paradas de recarga de forma ótima, garantindo a factibilidade da bateria sem alterar a sequência de clientes.

A relevância deste artigo para o nosso trabalho é multifacetada. Primeiro, ele estabelece as instâncias de teste que formam a base para a avaliação experimental. Segundo, sua heurística 3P-Heu demonstra a eficácia de uma abordagem que separa a busca pela factibilidade da otimização do custo, uma filosofia similar à adotada por Castellucci, Franco e Santiago (2024). Por fim, o fato de a 3P-Heu ser fortemente baseada em mecanismos de busca local (como VND e 2-opt) reforça a importância de se

estudar esses operadores de forma isolada.

## 4 DEFINIÇÃO DO PROBLEMA

O problema central deste trabalho é o Problema do Caixeiro Viajante com Veículos Elétricos e Janelas de Tempo (ETSPTW), uma variante NP-Difícil do clássico TSP. Sua complexidade reside não apenas na minimização da distância total percorrida, mas na dificuldade de se encontrar uma rota que satisfaça simultaneamente um conjunto de restrições operacionais rigorosas. A definição formal do problema, que adotamos neste trabalho, segue a proposta de Roberti e Wen (2016). Nesta formulação, o ETSPTW é modelado sobre um grafo  $G = (V, A)$ , onde  $V$  é o conjunto de vértices e  $A$  o conjunto de arcos. O conjunto de vértices  $V$  é composto por um depósito (representado por um vértice de origem  $o$  e um vértice de destino  $d$ ), um conjunto de clientes  $C$  a serem visitados, e um conjunto de estações de recarga  $S$ . Cada arco em  $A$  possui custos associados de distância ( $d_{ij}$ ), tempo de viagem ( $t_{ij}$ ) e consumo de bateria ( $q_{ij}$ ).

O objetivo do ETSPTW, conforme descrito por Roberti e Wen (2016), é encontrar a rota de menor distância que parte do depósito  $o$ , visita todos os clientes em  $C$  exatamente uma vez e retorna ao depósito  $d$ , obedecendo a um conjunto estrito de restrições. Cada cliente  $i$  possui uma janela de tempo rígida  $[e_i, l_i]$ , que define o intervalo de tempo permitido para o início do serviço. Adicionalmente, o veículo elétrico opera com uma bateria de capacidade máxima  $Q$ , e seu nível de carga deve permanecer sempre entre zero e  $Q$  durante todo o percurso. O consumo de bateria é considerado linearmente proporcional à distância, com uma taxa de consumo  $h$  constante. Da mesma forma, o tempo de recarga é linear em relação à energia reabastecida, com uma taxa  $g$  uniforme para todas as estações. Este trabalho adota a política de recarga total (*full-recharge*), na qual, ao visitar uma estação, a bateria do veículo é sempre recarregada até sua capacidade máxima.

A complexidade dessas restrições impõe uma interdependência rígida entre a sequência de visitas e a viabilidade da rota. Para ilustrar, a Figura 2 apresenta visualmente uma instância do problema. Nela, é possível observar o depósito central, os clientes com suas respectivas janelas de tempo (indicadas abaixo de cada local) e as estações de recarga estrategicamente posicionadas. As setas indicam o fluxo da rota, demonstrando como o veículo deve alternar entre visitas a clientes e paradas para recarga, respeitando os horários limites de cada nó.

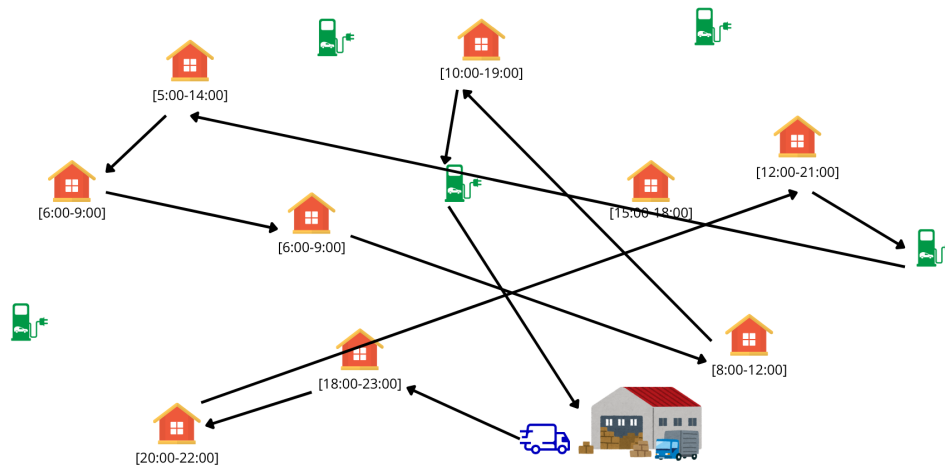


Figura 2 – A rota inicia e termina no depósito (ícone de armazém), visitando clientes (casas) dentro de suas janelas de tempo e utilizando estações de recarga (ícones verdes) para manter a autonomia. Fonte: Autor.

Dada a interdependência e a rigidez dessas restrições, o desafio primário muitas vezes transcende a otimização da distância, concentrando-se na busca por uma solução factível, ou seja, uma rota que não viole nenhuma condição.

## 5 PROPOSTA DE SOLUÇÃO

A proposta de solução deste trabalho consiste na implementação e avaliação comparativa de duas estratégias de busca estocástica: a Busca Local e a Busca Local Iterada (ILS). O foco da avaliação reside no impacto dos operadores de vizinhança 2-opt e 3-opt no desempenho de ambas as estratégias. A implementação da Busca Local toma como ponto de partida o Algoritmo 1 apresentado por Castellucci, Franco e Santiago (2024), que se baseia no operador 2-opt e em uma função de avaliação que mede a infactibilidade da rota. Por sua vez, a Busca Local Iterada (ILS), que estende a primeira abordagem com mecanismos de perturbação, é apresentada no Algoritmo 2.

A hipótese central deste trabalho é que o operador 3-opt, apesar de seu maior custo computacional por iteração, proporcionará um desempenho superior ao 2-opt em ambas as estratégias de busca (Busca Local e ILS). Essa premissa fundamenta-se na estrutura de vizinhança mais complexa e rica do 3-opt, que oferece um mecanismo de exploração mais robusto. Espera-se que essa robustez aumente a capacidade do algoritmo de escapar de mínimos locais (soluções com alta penalidade) e, conseqüentemente, de encontrar soluções factíveis de maior qualidade ou com maior frequência.

O operador 3-opt funciona de forma mais abrangente que o 2-opt. Em vez de remover duas arestas, ele remove três arestas não adjacentes do *tour*. Isso tem o efeito de dividir a rota em três subcaminhos desconectados. A principal vantagem do

Algoritmo 1 – Busca Local proposta

---

```

1  FUNCTION local-search(solution)
2      STATE best <- solution
3      WHILE TRUE
4          STATE candidate <- f2opt(best)
5          IF value(candidate) < value(best)
6              best <- candidate // Atualiza a melhor solução
7          ELSE
8              // Nenhuma melhoria encontrada, ótimo local atingido
9              RETURN best
10         ENDIF
11     ENDWHILE
12 ENDFUNCTION

```

---

Fonte: Adaptado de Castellucci, Franco e Santiago (2024).

Algoritmo 2 – Busca Local Iterada (ILS) proposta

---

```

1  FUNCTION iterated-local-search(initial_solution, max_iter)
2  STATE s_best <- initial_solution
3  STATE s <- initial_solution
4  FOR i = 1, ..., max_iter
5  STATE s <- local-search(s)
6  IF value(s) < value(s_best)
7  STATE s_best <- s
8  ENDIF
9  // Perturbar a melhor solução
10 STATE s <- disturb(s_best)
11 ENDFOR
12 RETURN s_best
13 ENDFUNCTION

```

---

Fonte: Autor

3-opt reside no fato de que existem oito maneiras de reconectar esses três subcaminhos para formar um novo *tour* hamiltoniano válido, como pode ser visto na Figura 3. Uma dessas combinações resulta no *tour* original, enquanto as outras sete geram topologias de rota completamente novas, que podem incluir a inversão de um ou mais subcaminhos. Esta multiplicidade de opções de reconexão a cada passo cria uma vizinhança de busca muito mais ampla e diversificada, permitindo movimentos mais disruptivos no espaço de soluções.

A qualidade de uma solução é avaliada por meio de uma função de valor (*value*), que se baseia em penalidades aplicadas pela violação da capacidade da bateria e das janelas de tempo  $[e_i, \ell_i]$  para cada cliente  $i \in C$ . Em cada segmento da rota sem paradas para recarga, calcula-se o nível de energia do veículo ( $q_i$ ) ao alcançar uma estação ou o destino. Se esse valor for negativo ( $q_i < 0$ ), seu módulo ( $|q_i|$ ) é somado à penalidade de bateria. De modo similar, considerando  $\tau_i$  como o horário de chegada ao cliente  $i$ , o valor de  $\max(\tau_i - \ell_i, 0)$  é adicionado à penalidade de tempo. A Equação (5.1) apresenta o cálculo da penalidade total da rota.

$$\text{value(rota)} = \alpha \sum_{i \in C} \max(\tau_i - \ell_i, 0) + \beta \sum_{i \in C: q_i < 0} |q_i|. \quad (5.1)$$

Para validar esta hipótese, será conduzida uma análise experimental comparativa. Serão implementadas duas estratégias de busca, a Busca Local e a Busca Local

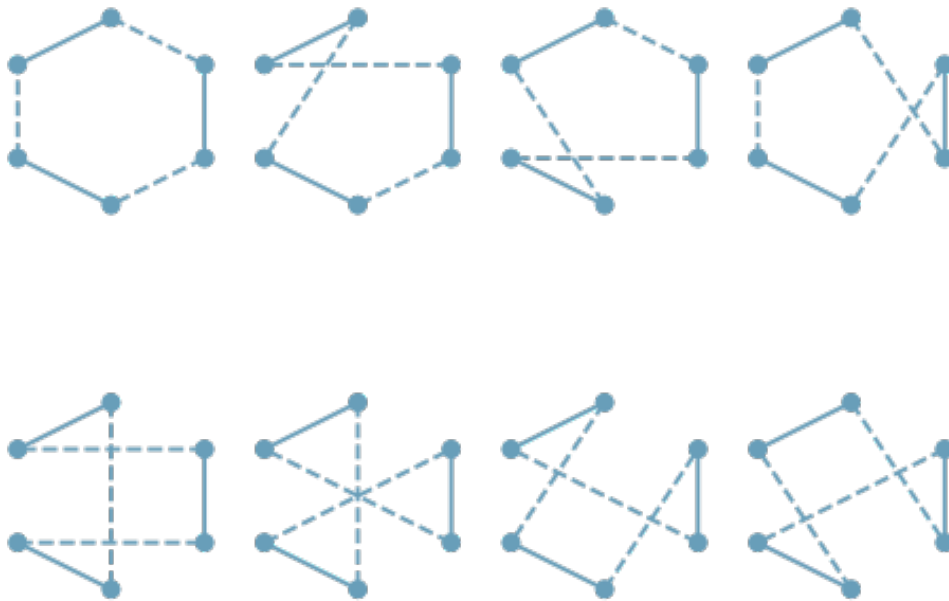


Figura 3 – As oito possíveis reconexões válidas do operador 3-opt. A primeira (canto superior esquerdo) representa o *tour* original. As outras sete representam as novas topologias possíveis, que podem incluir a inversão de um ou mais subcaminhos. Fonte: Mahéo (2017).

Iterada (ILS), avaliando-se o impacto dos operadores de vizinhança 2-opt e 3-opt em ambas. Essa metodologia resulta em quatro configurações de algoritmo, cujo desempenho será aferido utilizando o conjunto de instâncias de *benchmark* proposto por Roberti e Wen (2016).



## 6 EXPERIMENTOS COMPUTACIONAIS

Os algoritmos de Busca Local e Busca Local Iterada (ILS) foram implementados na linguagem de programação C++. Para garantir a reprodutibilidade dos experimentos, o código-fonte completo está disponível no repositório: <https://codigos.ufsc.br/pablo.lobes/iterative-local-search-opt>. Todo o processo de compilação foi realizado utilizando o compilador g++ (GCC), versão 11.4.0, com os seguintes argumentos: `-std=c++17 -Wall -march=native -O3 -flto -MMD -MP`, visando a máxima otimização de desempenho. Os experimentos foram executados em um computador equipado com um processador AMD Ryzen 7 5800H CPU @ 4.4GHz, 16 GB de memória RAM, e utilizando o sistema operacional Ubuntu 22.04 LTS. Nos experimentos foram utilizados  $\alpha = \beta = 100$  na Equação (5.1).

Para a avaliação dos resultados, as instâncias utilizadas foram as mesmas propostas no artigo de Roberti e Wen (2016), que serve como base de comparação para este trabalho. A fim de comparar a qualidade das soluções, foi calculado o *gap* utilizando a fórmula:  $gap = (custo - z^*)/z^*$ , em que  $z^*$  é a melhor solução conhecida encontrada por Roberti e Wen (2016), de acordo com as Tabelas 11 e 13 do artigo, e *custo* é o valor obtido pela heurística no teste.

Os resultados apresentados nas Tabelas 1 e 2 representam a média de 10 execuções para cada uma das 5 instâncias que compõem cada grupo. Para os testes com ILS, o número máximo de iterações utilizado no algoritmo foi de 200. As linhas que contêm o símbolo '-' indicam que, para aquele grupo de instâncias, nenhuma das 50 execuções totais foi capaz de encontrar uma solução factível.

A análise dos resultados inicia-se com a abordagem de Busca Local, na qual o operador 3-opt foi marcadamente superior. Enquanto o 2-opt obteve apenas 28% de sucesso em encontrar uma solução factível, o 3-opt alcançou 60% de sucesso. O tempo de execução médio do 3-opt, de 0,18 segundos, reflete uma exploração de vizinhança mais complexa e eficaz, que se mostrou crucial para superar mínimos locais de inviabilidade.

A transição da busca local simples para a meta-heurística ILS representou um salto qualitativo. Comparando a Busca Local 3-opt com o ILS 3-opt, o benefício fica evidente. A taxa de sucesso em encontrar soluções factíveis subiu de 60% para 100%, tornando o método robusto. Além disso, o *gap* médio foi significativamente reduzido de 0,49 para 0,37. Essa melhoria substancial na qualidade e confiabilidade, no entanto,

exigiu um investimento de tempo maior, com a média subindo de 0,18 segundos para 9,38 segundos, devido às múltiplas iterações de busca e perturbação.

No ILS, a comparação entre os operadores, 2-opt e 3-opt, reforça a eficácia do último. O ILS 3-opt obteve o menor *gap* médio, igual a 0,37, e mediana, igual a 0,35, superando o ILS 2-opt. Contudo, essa vantagem na qualidade implica um claro *trade-off* de desempenho: o ILS 3-opt é consideravelmente mais lento (9,38 segundos em média) que o ILS 2-opt, que apresentou uma velocidade excepcional de 0,05 segundos em média.

Finalmente, ao comparar o algoritmo que teve o melhor desempenho, o ILS 3-opt, com os resultados de Roberti e Wen (2016), a principal vantagem desta abordagem é a eficiência computacional. Embora a heurística de referência produza soluções de maior qualidade, como o *gap* médio de 0,37 indica, ela o faz com tempos de execução que podem atingir centenas ou até milhares de segundos. Em contraste, o ILS 3-opt entrega soluções de alta qualidade em um tempo médio de apenas 9,38 segundos.

Tabela 1 – A tabela apresenta os resultados obtidos pela heurística de Busca Local, comparando os operadores 2-opt e 3-opt. Para cada operador, são exibidas a média e a mediana do custo da solução, do  $gap \left( \frac{\text{custo} - z^*}{z^*} \right)$  e do tempo computacional (em segundos). Em que  $z^*$  é a melhor solução encontrada por Roberti e Wen (2016) e cada grupo contém cinco instâncias.

Classe		2-opt			3-opt		
		Custo	gap	t(s)	Custo	gap	t(s)
n20w120s5	Média	344,00	0,38	0,00	344,50	0,38	0,13
	Mediana	344,00	0,38	0,00	344,50	0,38	0,13
n20w140s5	Média	384,50	0,62	0,00	340,13	0,46	0,13
	Mediana	384,50	0,62	0,00	340,13	0,46	0,12
n20w160s5	Média	375,25	0,48	0,00	349,50	0,52	0,12
	Mediana	375,25	0,48	0,00	349,50	0,52	0,12
n20w180s5	Média	413,25	0,58	0,00	393,67	0,45	0,12
	Mediana	413,25	0,58	0,00	402,00	0,47	0,12
n20w200s5	Média	345,75	0,44	0,00	342,58	0,35	0,12
	Mediana	345,75	0,44	0,00	321,75	0,34	0,12
n20w120s10	Média	-	-	-	388,38	0,51	0,25
	Mediana	-	-	-	388,38	0,51	0,25
n20w140s10	Média	343,50	0,51	0,00	341,31	0,50	0,24
	Mediana	322,50	0,47	0,00	326,38	0,47	0,24
n20w160s10	Média	347,50	0,53	0,00	339,94	0,50	0,25
	Mediana	347,50	0,53	0,00	341,75	0,49	0,24
n20w180s10	Média	-	-	-	396,56	0,59	0,24
	Mediana	-	-	-	386,25	0,56	0,24
n20w200s10	Média	352,58	0,54	0,00	370,55	0,51	0,23
	Mediana	351,75	0,56	0,00	360,50	0,55	0,23

Tabela 2 – A tabela apresenta os resultados obtidos pela ILS, comparando os operadores 2-opt e 3-opt. Para cada operador, são exibidas a média e a mediana do custo da solução, do  $gap \left( \frac{\text{custo} - z^*}{z^*} \right)$  e do tempo computacional (em segundos). Em que  $z^*$  é a melhor solução encontrada por Roberti e Wen (2016) e cada grupo contém cinco instâncias.

Classe		2-opt			3-opt		
		Custo	gap	t(s)	Custo	gap	t(s)
n20w120s5	Média	370,43	0,34	0,03	364,30	0,32	5,72
	Mediana	381,50	0,32	0,03	381,50	0,31	5,62
n20w140s5	Média	307,64	0,30	0,02	306,10	0,29	5,48
	Mediana	319,50	0,29	0,02	319,50	0,29	5,52
n20w160s5	Média	324,90	0,43	0,03	324,62	0,43	5,15
	Mediana	346,25	0,38	0,03	346,25	0,38	5,10
n20w180s5	Média	334,88	0,37	0,03	334,03	0,37	5,38
	Mediana	347,18	0,41	0,03	345,50	0,40	5,30
n20w200s5	Média	324,76	0,30	0,03	323,05	0,29	5,32
	Mediana	304,35	0,27	0,04	302,00	0,27	5,22
n20w120s10	Média	392,09	0,44	0,07	369,50	0,36	13,99
	Mediana	386,90	0,42	0,07	386,18	0,38	13,93
n20w140s10	Média	328,08	0,39	0,07	319,75	0,36	13,20
	Mediana	320,55	0,39	0,06	318,88	0,35	13,42
n20w160s10	Média	335,13	0,50	0,07	332,27	0,49	13,46
	Mediana	348,02	0,45	0,07	342,00	0,45	14,08
n20w180s10	Média	352,18	0,47	0,07	345,36	0,44	13,23
	Mediana	374,22	0,45	0,07	364,52	0,45	13,32
n20w200s10	Média	332,55	0,35	0,06	327,93	0,33	12,85
	Mediana	316,92	0,34	0,06	314,00	0,32	12,40

## 7 CONCLUSÃO E TRABALHOS FUTUROS

O presente trabalho dedicou-se à avaliação de heurísticas de busca local para o Problema do Caixeiro Viajante com Veículos Elétricos e Janelas de Tempo (ETSPTW), uma variante NP-Difícil de grande relevância teórica e prática no contexto da logística sustentável. O objetivo central foi comparar o desempenho dos operadores de vizinhança 2-opt e 3-opt, tanto em uma estrutura de Busca Local quanto em uma meta-heurística de ILS, a fim de determinar qual abordagem oferece o melhor equilíbrio entre qualidade da solução e eficiência computacional.

Os experimentos computacionais, realizados sobre instâncias de *benchmark* da literatura, forneceram resultados claros e conclusivos. A análise inicial, focada na Busca Local, demonstrou a superioridade marcante do operador 3-opt, que obteve uma taxa de sucesso de 60% na obtenção de soluções factíveis, em contraste com os 28% alcançados pelo 2-opt. Este resultado sugere que a vizinhança mais rica e complexa do 3-opt é fundamental para escapar de mínimos locais de inviabilidade, que representam o primeiro grande desafio na resolução do ETSPTW.

A transição para a meta-heurística de ILS representou um salto qualitativo para ambas as abordagens. A combinação da Busca Local com um mecanismo de perturbação permitiu que ambas as configurações encontrassem soluções factíveis em 100% das execuções, validando a eficácia da ILS em promover uma exploração mais abrangente do espaço de busca. Neste contexto, a comparação entre os operadores evidenciou um claro *trade-off*: enquanto o ILS com 2-opt se destacou pela sua excepcional velocidade, o ILS com 3-opt consistentemente produziu soluções de maior qualidade, alcançando o menor *gap* médio (0,37) em relação às melhores soluções conhecidas.

Diante do exposto, pode-se afirmar que a hipótese central deste trabalho foi corroborada. O operador 3-opt, apesar de seu maior custo computacional por iteração, provou ser mais robusto e eficaz, gerando soluções superiores tanto em termos de factibilidade quanto de custo final. A abordagem ILS 3-opt, embora mais lenta que sua contraparte 2-opt, estabeleceu-se como a mais performática, obtendo menor *gap* médio, e oferecendo uma alternativa competitiva aos métodos da literatura ao entregar soluções de alta qualidade em um tempo computacional significativamente inferior.

Como trabalhos futuros, diversas direções de pesquisa podem ser exploradas. Sugere-se a investigação de estruturas de vizinhança variáveis (VNS), que poderiam al-

ternar dinamicamente entre operadores mais simples como o 2-opt e mais complexos como o 3-opt para otimizar o balanço entre intensificação e diversificação. Adicionalmente, a implementação de outras meta-heurísticas, como *Simulated Annealing* ou Busca Tabu, poderia ser avaliada para o ETSPTW.

Outra possibilidade seria o desenvolvimento de mecanismos de perturbação mais sofisticados para o ILS. Nesse contexto, seria pertinente investigar a convergência do algoritmo em função do número de iterações, correlacionando o esforço computacional com a qualidade dos resultados obtidos, além de validar o método utilizando instâncias de maior dimensão. Por fim, a hibridização da heurística com métodos exatos para otimizar subproblemas específicos poderia refinar ainda mais a qualidade das soluções encontradas.

## REFERÊNCIAS

- BACK, T. **Evolutionary algorithms in theory and practice**. New York, NY: Oxford University Press, 1996.
- CASTELLUCCI, P.; FRANCO Álvaro; SANTIAGO, R. Soluções factíveis para o problema do caixeiro viajante com veículo elétrico e janelas de tempo. In: **Anais do IX Encontro de Teoria da Computação**. Porto Alegre, RS, Brasil: SBC, 2024. p. 114–118. ISSN 2595-6116.
- DORIGO, M.; CARO, G. D. Ant colony optimization: a new meta-heuristic. In: **Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)**. [S.l.: s.n.], 1999. v. 2, p. 1470–1477 Vol. 2.
- HOOS, H. H.; STÜTZLE, T. Stochastic local search. In: **Handbook of Approximation Algorithms and Metaheuristics**. [S.l.]: Chapman and Hall/CRC, 2005. p. 297–307.
- JÜNGER, M.; REINELT, G.; RINALDI, G. The traveling salesman problem. In: **Handbooks in operations research and management science**. [S.l.]: Elsevier, 1995. v. 7, p. 225–330.
- KIRKPATRICK, S.; GELATT JR, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, American Association for the Advancement of Science (AAAS), v. 220, n. 4598, p. 671–680, maio 1983.
- KUCUKOGLU, I.; DEWIL, R.; CATTRYSSE, D. The electric vehicle routing problem and its variations: A literature review. **Computers & Industrial Engineering**, v. 161, p. 107650, 2021. ISSN 0360-8352.
- LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. **Oper. Res.**, Institute for Operations Research and the Management Sciences (INFORMS), v. 21, n. 2, p. 498–516, abr. 1973.
- MAHÉO, A. **Solving the TSP with local search heuristics**. 2017. <https://arthur.maheo.net/python-local-tsp-heuristics/>. Acesso em: 2024-05-21.
- ROBERTI, R.; WEN, M. The electric traveling salesman problem with time windows. **Transportation Research Part E: Logistics and Transportation Review**, v. 89, p. 32–52, 05 2016.

# Avaliação de heurísticas para um problema do caixeiro viajante com veículos elétricos

<sup>1</sup> Departamento de Informática e Estatística - Universidade Federal de Santa Catarina (UFSC)  
Florianópolis - Santa Catarina - Brasil

pablo.lopes@grad.ufsc.br, pedro.castellucci@ufsc.br

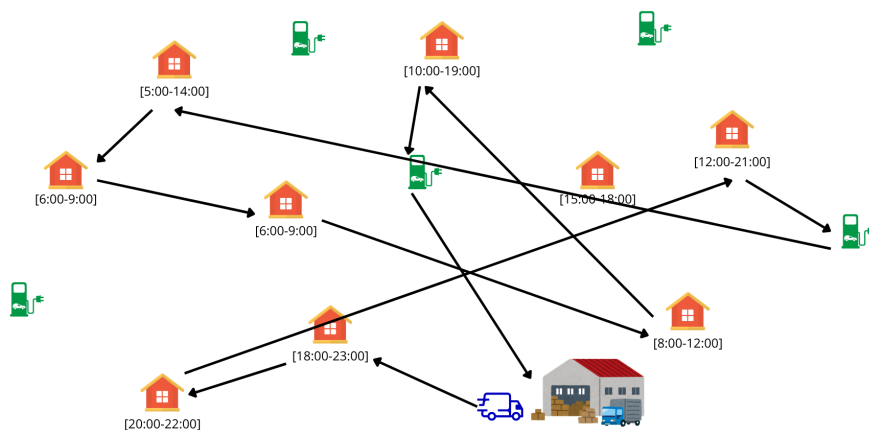
**Abstract.** *The Electric Traveling Salesman Problem with Time Windows (ETSPTW) poses optimization challenges beyond distance minimization, requiring strict adherence to battery autonomy and customer service constraints. This paper investigates the performance of Local Search and Iterated Local Search (ILS) heuristics in solving the ETSPTW, focusing on a comparative analysis between 2-opt and 3-opt neighborhood operators. Computational experiments on standard benchmark instances indicate that, although computationally more expensive, the 3-opt operator offers greater robustness in finding feasible solutions, effectively overcoming the tendency of the 2-opt operator to stagnate in infeasible local optima. The ILS approach combined with the 3-opt operator achieved the best trade-off between solution quality and consistency, validating the importance of complex neighborhood structures in highly constrained problems.*

**Resumo.** *O Problema do Caixeiro Viajante com Veículos Elétricos e Janelas de Tempo (ETSPTW) impõe desafios de otimização que transcendem a minimização de distâncias, exigindo o cumprimento estrito de restrições de bateria e horários de atendimento. Este trabalho investiga o desempenho de heurísticas de Busca Local e Busca Local Iterada (ILS) na resolução do ETSPTW, com foco em uma análise comparativa entre os operadores de vizinhança 2-opt e 3-opt. Os experimentos computacionais realizados em instâncias de referência da literatura indicam que, embora computacionalmente mais custoso, o operador 3-opt oferece maior robustez na obtenção de soluções factíveis, superando a tendência do 2-opt de estagnar em mínimos locais inviáveis. A abordagem ILS combinada com o operador 3-opt demonstrou o melhor compromisso entre qualidade da solução e consistência, validando a importância de estruturas de vizinhança mais complexas em problemas altamente restritos.*

## 1. Introdução

A crescente eletrificação das frotas de logística de última milha renovou o interesse acadêmico e industrial em variantes do clássico Problema do Caixeiro Viajante (TSP). O Problema do Caixeiro Viajante com Veículos Elétricos e Janelas de Tempo (ETSPTW) modela esse cenário ao introduzir restrições operacionais críticas, como a autonomia limitada das baterias, a necessidade de paradas para recarga e o respeito às janelas de tempo dos clientes. A complexidade do ETSPTW reside não apenas em sua natureza NP-Difícil,

mas na dificuldade prática de encontrar rotas que sejam factíveis, visto que a violação de qualquer restrição de tempo ou energia invalida a solução. A Figura 1 ilustra os componentes principais e a dinâmica de uma instância deste problema.



**Figura 1. Representação esquemática de uma instância do ETSPTW. A rota inicia e termina no depósito, visitando clientes com janelas de tempo definidas  $[e_i, l_i]$  e recorrendo a estações de recarga (ícones verdes) para manter a autonomia do veículo.**

Métodos exatos enfrentam limitações de escalabilidade em instâncias de tamanho realista, o que motiva o uso de métodos heurísticos baseados em busca local. No entanto, a eficácia dessas heurísticas depende intrinsecamente da topologia do espaço de busca definida pelos operadores de vizinhança. Vizinhanças simples podem não oferecer caminhos de escape para configurações inviáveis, enquanto vizinhanças complexas podem onerar o tempo de processamento. Nesse contexto, este trabalho avalia o impacto da escolha entre os operadores 2-opt e 3-opt dentro de estruturas de Busca Local e Busca Local Iterada (ILS), visando quantificar o ganho de robustez proporcionado pela maior complexidade do 3-opt frente à eficiência temporal do 2-opt.

## 2. Definição do Problema e Metodologia

O ETSPTW é definido sobre um grafo onde o objetivo é determinar uma rota de custo mínimo que visite cada cliente exatamente uma vez, respeitando a capacidade máxima da bateria e as janelas de tempo predefinidas. O modelo assume uma política de recarga total nas estações e penaliza linearmente qualquer violação de restrição. A abordagem metodológica adotada baseia-se na proposta de [Castellucci et al. 2024], utilizando uma função de avaliação que prioriza a minimização da infactibilidade. Diferente da função objetivo clássica do TSP, o custo da solução aqui é composto pela soma ponderada das violações de bateria e atrasos nas janelas de tempo, sendo uma solução considerada factível apenas quando esse valor é nulo. A função de avaliação adotada é formalizada pela equação:

$$\text{value(rota)} = \alpha \sum_{i \in C} \max(\tau_i - \ell_i, 0) + \beta \sum_{i \in C: q_i < 0} |q_i|. \quad (1)$$

onde  $\tau_i$  representa o horário de chegada no cliente  $i$ ,  $l_i$  o limite superior da janela de tempo e  $q_i$  a carga da bateria. Os parâmetros  $\alpha$  e  $\beta$  são pesos de penalidade que direcionam a busca para regiões factíveis.

Para a resolução do problema, foram implementados dois algoritmos principais em linguagem C++. A Busca Local utiliza uma estratégia de descida para encontrar ótimos locais, enquanto a Busca Local Iterada (ILS) estende esse processo aplicando perturbações à melhor solução encontrada, permitindo a exploração de novas vizinhanças e mitigando a estagnação prematura. Vale ressaltar que, para avaliar a robustez dos operadores, todas as execuções partiram de soluções iniciais geradas por permutações aleatórias dos clientes e estações, sem o uso de heurísticas construtivas sofisticadas. O estudo compara o operador 2-opt (Figura 2), que remove duas arestas e inverte o subcaminho resultante, com o operador 3-opt (Figura 3). Este último remove três arestas e permite oito formas de reconexão, oferecendo uma exploração mais abrangente da topologia da rota ao custo de uma complexidade cúbica. As instâncias utilizadas para validação seguem o padrão proposto por [Roberti and Wen 2016], amplamente aceito na literatura.

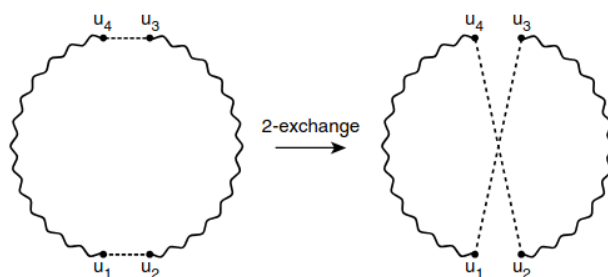


Figura 2. Representação de um único passo do SLS para o TSP utilizando 2-opt. [Hoos and Stützle 2005]

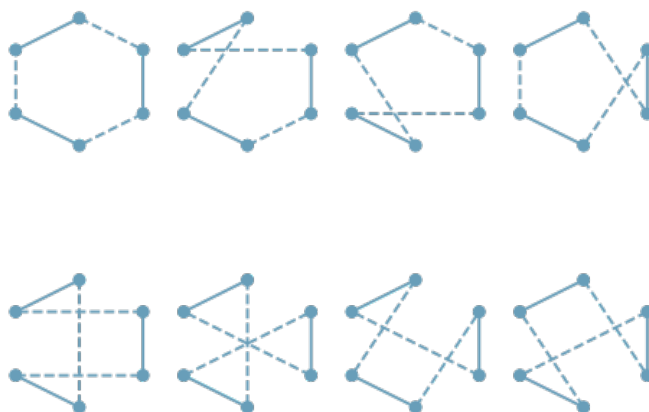


Figura 3. As oito possíveis reconexões válidas do operador 3-opt. A primeira (canto superior esquerdo) representa o *tour* original. As outras sete representam as novas topologias possíveis, que podem incluir a inversão de um ou mais subcaminhos. [Mahéo 2017]

Para os experimentos, os pesos de penalidade na Equação (1) foram definidos como  $\alpha = \beta = 100$ , priorizando fortemente a correção de violações em detrimento de pequenas melhorias na distância durante as fases iniciais da busca. O critério de parada para o ILS foi fixado em 200 iterações, e os testes foram executados em um processador AMD Ryzen 7 5800H.

Para avaliar a qualidade das soluções, utiliza-se a métrica do *gap*, calculada conforme a Equação (2):

$$\text{gap} = \frac{\text{custo} - z^*}{z^*} \quad (2)$$

onde  $z^*$  representa o valor da melhor solução conhecida para a instância, conforme reportado por [Roberti and Wen 2016]. Um *gap* próximo de zero indica que a solução encontrada é competitiva com o estado da arte.

### 3. Análise de Resultados

As Tabelas 1 e 2 apresentam os resultados detalhados dos experimentos. A análise comparativa revelou comportamentos distintos entre os operadores quando submetidos às restrições rígidas do ETSPTW.

**Tabela 1. Resultados obtidos pela heurística de Busca Local**

Classe	Métrica	2-opt			3-opt		
		Custo	Gap	t(s)	Custo	Gap	t(s)
n20w120s5	Média	344,00	0,38	0,00	344,50	0,38	0,13
	Mediana	344,00	0,38	0,00	344,50	0,38	0,13
n20w140s5	Média	384,50	0,62	0,00	340,13	0,46	0,13
	Mediana	384,50	0,62	0,00	340,13	0,46	0,12
n20w160s5	Média	375,25	0,48	0,00	349,50	0,52	0,12
	Mediana	375,25	0,48	0,00	349,50	0,52	0,12
n20w180s5	Média	413,25	0,58	0,00	393,67	0,45	0,12
	Mediana	413,25	0,58	0,00	402,00	0,47	0,12
n20w200s5	Média	345,75	0,44	0,00	342,58	0,35	0,12
	Mediana	345,75	0,44	0,00	321,75	0,34	0,12
n20w120s10	Média	-	-	-	388,38	0,51	0,25
	Mediana	-	-	-	388,38	0,51	0,25
n20w140s10	Média	343,50	0,51	0,00	341,31	0,50	0,24
	Mediana	322,50	0,47	0,00	326,38	0,47	0,24
n20w160s10	Média	347,50	0,53	0,00	339,94	0,50	0,25
	Mediana	347,50	0,53	0,00	341,75	0,49	0,24
n20w180s10	Média	-	-	-	396,56	0,59	0,24
	Mediana	-	-	-	386,25	0,56	0,24
n20w200s10	Média	352,58	0,54	0,00	370,55	0,51	0,23
	Mediana	351,75	0,56	0,00	360,50	0,55	0,23

Na aplicação da Busca Local (Tabela 1), a limitação do operador 2-opt tornou-se evidente, obtendo sucesso em encontrar soluções factíveis em apenas 28% das execuções. Essa baixa taxa deve-se à tendência do 2-opt de estagnar em mínimos locais onde a rota é inviável e nenhuma troca simples de duas arestas é suficiente para resolver o conflito de

restrições. Em contrapartida, o operador 3-opt, beneficiando-se de uma vizinhança mais rica, alcançou 60% de sucesso na obtenção de factibilidade sob as mesmas condições.

A introdução da meta-heurística ILS elevou o desempenho de ambos os operadores, permitindo que atingissem 100% de sucesso na geração de soluções factíveis devido ao mecanismo de perturbação. Contudo, observou-se um *trade-off* claro entre qualidade e tempo, conforme detalhado na Tabela 2.

**Tabela 2. Resultados obtidos pela heurística ILS**

Classe	Métrica	2-opt			3-opt		
		Custo	Gap	t(s)	Custo	Gap	t(s)
n20w120s5	Média	370,43	0,34	0,03	364,30	0,32	5,72
	Mediana	381,50	0,32	0,03	381,50	0,31	5,62
n20w140s5	Média	307,64	0,30	0,02	306,10	0,29	5,48
	Mediana	319,50	0,29	0,02	319,50	0,29	5,52
n20w160s5	Média	324,90	0,43	0,03	324,62	0,43	5,15
	Mediana	346,25	0,38	0,03	346,25	0,38	5,10
n20w180s5	Média	334,88	0,37	0,03	334,03	0,37	5,38
	Mediana	347,18	0,41	0,03	345,50	0,40	5,30
n20w200s5	Média	324,76	0,30	0,03	323,05	0,29	5,32
	Mediana	304,35	0,27	0,04	302,00	0,27	5,22
n20w120s10	Média	392,09	0,44	0,07	369,50	0,36	13,99
	Mediana	386,90	0,42	0,07	386,18	0,38	13,93
n20w140s10	Média	328,08	0,39	0,07	319,75	0,36	13,20
	Mediana	320,55	0,39	0,06	318,88	0,35	13,42
n20w160s10	Média	335,13	0,50	0,07	332,27	0,49	13,46
	Mediana	348,02	0,45	0,07	342,00	0,45	14,08
n20w180s10	Média	352,18	0,47	0,07	345,36	0,44	13,23
	Mediana	374,22	0,45	0,07	364,52	0,45	13,32
n20w200s10	Média	332,55	0,35	0,06	327,93	0,33	12,85
	Mediana	316,92	0,34	0,06	314,00	0,32	12,40

O algoritmo ILS com 3-opt produziu as soluções de melhor qualidade global, apresentando um *gap* médio de 0,37 em relação às melhores soluções conhecidas da literatura, com um tempo médio de execução de 9,38 segundos. Já o ILS com 2-opt, embora tenha apresentado um *gap* médio superior, destacou-se pela velocidade extrema, convergindo em uma média de 0,05 segundos.

#### 4. Conclusão

Os experimentos corroboram a hipótese de que a estrutura de vizinhança do 2-opt, apesar de eficiente em tempo, é frequentemente insuficiente para transpor barreiras de inviabilidade no ETSPW sem o auxílio de mecanismos de diversificação robustos. O operador 3-opt provou ser superior na navegação do espaço de busca restrito, gerando soluções de maior qualidade e demonstrando maior capacidade autônoma de recuperação de estados inviáveis. Conclui-se que a utilização do ILS com 3-opt é a estratégia recomendada para cenários onde a qualidade da solução é prioritária. Entretanto, para aplicações logísticas em tempo real que demandam resposta imediata, o ILS com 2-opt apresenta-se como uma alternativa viável, entregando soluções factíveis com celeridade. Trabalhos futuros poderão explorar o uso de vizinhanças variáveis (VNS) para alternar dinamicamente entre esses operadores, buscando otimizar o balanço entre intensificação e diversificação.

## Referências

- Castellucci, P., Álvaro Franco, and Santiago, R. (2024). Soluções factíveis para o problema do caixeiro viajante com veículo elétrico e janelas de tempo. In *Anais do IX Encontro de Teoria da Computação*, pages 114–118, Porto Alegre. SBC.
- Hoos, H. H. and Stützle, T. (2005). Stochastic local search. In Gonzalez, T. F., editor, *Handbook of Approximation Algorithms and Metaheuristics*, pages 297–307. Chapman and Hall/CRC.
- Mahéo, A. (2017). Solving the TSP with local search heuristics. <https://arthur.maheo.net/python-local-tsp-heuristics/>. Acesso em: 21 maio 2024.
- Roberti, R. and Wen, M. (2016). The electric traveling salesman problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 89:32–52.