



Universidade Federal da Paraíba
Centro de Informática
Graduação em Engenharia da Computação

Uma heurística eficiente para problemas do caixeiro viajante com janelas de tempo

Bruno Sidnei Guerra Passeti

João Pessoa - PB
2023

Bruno Sidnei Guerra Passeti

Uma heurística eficiente para problemas do caixeiro viajante com janelas de tempo

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal da Paraíba (UFPB), como requisito para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: Prof. Anand Subramanian

Catálogo na publicação
Seção de Catalogação e Classificação

P287h Passeti, Bruno Sidnei Guerra.

Uma heurística eficiente para problemas do caixeiro viajante com janelas de tempo / Bruno Sidnei Guerra Passeti. - João Pessoa, 2023.

37 f. : il.

Orientação: Anand Subramanian.

TCC (Graduação) - UFPB/CI.

1. Roteamento. 2. Problema do caixeiro viajante com janelas de tempo. 3. Makespan. 4. Iterated Local Search. 5. Algoritmos. I. Subramanian, Anand. II. Título.

UFPB/CI


CDU 004.421

Bruno Sidnei Guerra Passeti

Uma heurística eficiente para problemas do caixeiro viajante com janelas de tempo

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal da Paraíba (UFPB), como requisito para obtenção do grau de Bacharel em Engenharia da Computação.

Trabalho aprovado. João Pessoa - PB, 13 de Novembro de 2023:

Documento assinado digitalmente
 **ANAND SUBRAMANIAN**
Data: 25/11/2023 16:53:09-0300
Verifique em <https://validar.iti.gov.br>

Prof. Anand Subramanian
Orientador

Eduardo Vieira Queiroga
Examinador

Prof. Teobaldo Bulhões
Examinador

João Pessoa - PB
2023

Agradecimentos

Gostaria de expressar minha profunda gratidão à minha família pelo apoio emocional e financeiro que me foi dado. Sua ajuda foi crucial durante os momentos mais desafiadores da minha graduação. Ao corpo docente do curso de Engenharia da Computação da UFPB, minha sincera gratidão pela dedicação e excelência no ensino. À UFPB, agradeço pela infraestrutura de qualidade, pelas oportunidades e pelo suporte financeiro que me permitiram iniciar minha jornada na pesquisa científica e acadêmica.

Aos meus colegas do laboratório de pesquisa e amigos da graduação, meu agradecimento por me acompanharem nas tarefas acadêmicas e por tornarem os momentos de estudo mais leves e alegres.

Um agradecimento especial a Anand Subramanian, Bruno Petrato Bruck e Teobaldo Bulhões, cujo exemplo de dedicação, esforço, responsabilidade e comprometimento transformaram nosso ambiente de trabalho em um espaço de excelência e acolhimento.

Por último, mas não menos importante, agradeço à minha companheira, Vitória Gaspar Bernardo, que entrou em minha vida cedo e me transformou em uma pessoa melhor e mais feliz.

Resumo

Este estudo introduz uma abordagem heurística para o Problema do Caixeiro Viajante com Janelas de Tempo e o Problema do Caixeiro Viajante com Janelas de Tempo e Minimização de Makespan. A heurística implementada é baseada nas técnicas GRASP, ILS e RVND. Além disso, emprega uma técnica eficaz para avaliar as restrições das janelas de tempo, utilizando estruturas específicas para a concatenação de subsequências, apresentando complexidade $O(1)$ amortizada. Diversos experimentos foram realizados conjuntos de instâncias da literatura, revelando que esta metodologia apresenta desempenho comparável aos métodos existentes da literatura para o TSPTW e TSPTW-M e exibe melhorias notáveis em alguns casos nas instâncias do TSPTW.

Palavras-chave: Roteamento. Problema do Caixeiro Viajante com Janelas de Tempo. *Makespan. Iterated Local Search.*

Abstract

This study introduces a heuristic approach for the Traveling Salesman Problem with Time Windows (TSPTW) and the Traveling Salesman Problem with Time Windows and Makespan Minimization (TSPTW-M). The implemented heuristic is based on GRASP, ILS, and RVND techniques. Furthermore, it employs an effective technique for evaluating the constraints of time windows, using specific structures for the concatenation of subsequences, with amortized $O(1)$ complexity. Various experiments were conducted using instance sets from the literature, revealing that this methodology demonstrates performance comparable to existing methods in the literature for TSPTW and TSPTW-M and exhibits notable improvements in some TSPTW instances.

Keywords: Routing. Traveling Salesman Problem with Time Windows. Makespan. Iterated Local Search.

Lista de tabelas

Tabela 1 – Resumo dos trabalhos relacionados, ano e método de resolução.	19
Tabela 2 – Resultados para TSPTW em instâncias propostas por Dumas et al. (1995)	26
Tabela 3 – Resultados para TSPTW em instâncias propostas por Gendreau et al. (1998)	27
Tabela 4 – Resultados para TSPTW em instâncias propostas por Ohlmann e Thomas (2007)	27
Tabela 5 – Resultados para TSPTW em instâncias propostas por Silva e Urrutia (2010)	28
Tabela 6 – Resultados para TSPTW-M em instâncias propostas por Ascheuer (1996)	29
Tabela 7 – Resultados para TSPTW-M em instâncias propostas por Dumas et al. (1995)	30
Tabela 8 – Resultados para TSPTW-M em instâncias propostas por Gendreau et al. (1998)	31
Tabela 9 – Resultados para TSPTW-M em instâncias propostas por Ohlmann e Thomas (2007)	32
Tabela 10 – Resultados para TSPTW-M em instâncias propostas por Pesant et al. (1998)	33
Tabela 11 – Resultados para TSPTW-M em instâncias propostas por Potvin e Bengio (1996)	34
Tabela 12 – Resultados agregados para TSPTW	34
Tabela 13 – Resultados agregados para TSPTW-M.	34

Lista de ilustrações

Figura 1	– Exemplo de reinserção em uma rota com 5 vértices, em que o vértice 2 é reinserido na posição 5.	22
Figura 2	– Exemplo de Swap em uma rota com 5 vértices, em que o vértice 2 é trocado com o vértice 5.	22
Figura 3	– Exemplo do Or-opt2 em uma rota com 5 vértices, em que os vértices 2 e 3 são reinseridos na posição 5.	22
Figura 4	– Exemplo do Or-opt3 em uma rota com 5 vértice, em que os vértices 2, 3 e 4 são reinseridos na posição 5	23
Figura 5	– Exemplo do Double-Bridge em uma rota com 8 clientes, em que arcos em azul são removidos e rearranjados	24

Sumário

1	INTRODUÇÃO	12
1.1	Definição do Tema	12
1.2	Justificativa	13
1.3	Objetivos	13
1.3.1	Objetivo Geral	13
1.3.2	Objetivos específicos	13
1.4	Estrutura do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	15
2.1	Fundamentação Teórica	15
2.1.1	Algoritmos Construtivos	15
2.1.2	Meta-heurísticas	15
2.1.3	Meta-heurísticas Baseadas em Populações	16
2.1.3.1	Algoritmos Genéticos	16
2.1.3.2	Algoritmo Baseado em Colônia de Formigas	16
2.1.3.3	Algoritmo Baseado em Enxame de Partículas	16
2.1.4	Meta-heurísticas Baseadas em Busca Local	17
2.1.4.1	<i>Iterated Local Search</i>	17
2.1.4.2	<i>Greedy Randomized Adaptive Search Procedure</i>	17
2.1.4.3	<i>Simulated Annealing</i>	18
2.1.4.4	<i>Tabu Search</i>	18
2.2	Trabalhos Relacionados	19
3	PROCEDIMENTOS METODOLÓGICOS	20
3.1	Algoritmo Construtivo	20
3.2	Busca local	20
3.2.1	Estruturas de Vizinhanças	22
3.2.2	Avaliação Eficiente	23
3.3	Mecanismo de Perturbação	24
4	EXPERIMENTOS COMPUTACIONAIS	25
4.1	Resultados	25
4.1.1	Resultados para o TSPTW	25
4.1.2	Resultados para o TSPTW-M	26
5	CONSIDERAÇÕES FINAIS	35

REFERÊNCIAS	36
-----------------------	----

1 Introdução

1.1 Definição do Tema

A gestão do tempo de espera e entrega desempenha um papel de extrema relevância na área da logística, especialmente quando se busca minimizar o percurso das rotas. Nesse contexto, são desenvolvidos métodos para realizar o roteamento de veículos, com o objetivo principal de determinar a trajetória mais curta que percorre todos os clientes, dando origem ao que é conhecido na literatura como o Problema do Caixeiro Viajante (TSP, do inglês, *Traveling Salesman Problem*). O TSP é uma classe de problemas que engloba diversos domínios, incluindo logística, redes e transporte. Para acomodar uma variedade de cenários, surgem variantes desse problema, discutido em (GUTIN; PUNNEN, 2006), como o Problema do Caixeiro Viajante com Coleta e Entrega (PDTSP, do inglês, *Pickup and Delivery Traveling Salesman Problem*), o Problema do Caixeiro Viajante com Janelas de Tempo (TSPTW, do inglês, *Traveling Salesman Problem with Time Windows*), entre outros.

Este trabalho acadêmico apresenta uma nova solução para um desafio comum enfrentado no cotidiano, que envolve a otimização da entrega de serviços levando em consideração restrições de janelas de tempo para atender aos clientes. Em particular, será apresentada uma solução heurística para o TSPTW, explorando as dificuldades associadas à entrega eficiente de serviços dentro de intervalos de tempo predefinidos.

Adicionalmente, este estudo se expande para abordar uma variante complexa do problema, denominada Problema do Caixeiro Viajante com Janelas de Tempo e Minimização de *Makespan* (TSPTW-M). Enquanto o TSPTW foca em minimizar o tempo total de viagem, o TSPTW-M se concentra em minimizar o *makespan*, que é o tempo total desde o início até a conclusão de todas as entregas. Essa abordagem é particularmente útil em situações onde a eficiência da distribuição não depende apenas da distância percorrida, mas também do equilíbrio na distribuição do tempo de serviço entre os diferentes pontos de entrega. Uma solução heurística para o TSPTW-M também será discutida e analisada, proporcionando uma visão completa das estratégias empregadas para otimizar as operações de entrega sob diferentes perspectivas e restrições.

O TSPTW e o TSPTW-M são formulados como um problema de grafos, onde um conjunto de vértices $V = \{0, 1, \dots, n\}$ representa locais de entrega, e os arcos $A = \{(i, j) | \forall i, j \in V, i \neq j\}$ representam as conexões entre esses locais. Além disso, é fornecido um conjunto de tempos de travessia t_{ij} associados a cada arco (i, j) , representando o tempo necessário para percorrer o arco entre os locais i e j . O vértice 0 é reservado para representar o depósito onde o veículo parte. O TSP busca minimizar $\sum_{(i,j) \in P} t_{ij}$, onde

$P \subset A$ é o caminho que representa a solução do problema. Sendo assim podendo ser descrito como encontrar o ciclo hamiltoniano mínimo no grafo V . A variante *TSPTW-M* (*Traveling Salesman Problem with Time Windows and Minimizing Makespan*) estende o TSPTW ao focar na minimização do *makespan*, que é o tempo total da rota desde a saída até o retorno ao depósito. Nesta variante, além de considerar os tempos de travessia e as janelas de tempo para as entregas, busca-se também minimizar o tempo total da jornada. Portanto, a maioria da formulação matemática e das restrições do problema são mantidas, porém, a função objetivo é modificada para focar na minimização do *makespan*, em vez de minimizar a soma dos tempos de travessia.

1.2 Justificativa

O TSP é um problema classificado como NP-difícil, assim como o TSPTW e o TSPTW-M, que são extensões do TSP. Portanto, até o momento, não existe um algoritmo conhecido que possa resolver o TSP de forma eficiente (ou seja, em tempo polinomial). Tal complexidade o torna um tópico de grande interesse na comunidade científica, especialmente para aqueles focados em otimização e algoritmos computacionais. A logística desempenha um papel crítico nas operações de empresas e organizações em todo o mundo, e a otimização das entregas é uma das áreas mais desafiadoras e impactantes dentro desse campo. A capacidade de realizar entregas de forma eficiente, cumprindo janelas de tempo específicas, é crucial para garantir a satisfação do cliente, economizar recursos e reduzir os custos operacionais.

Portanto, o desenvolvimento de um algoritmo heurístico eficiente que visa encontrar soluções para o TSP e suas variantes, levando em consideração as restrições das janelas de tempo, melhora a compreensão do problema em questão e fornece ferramentas úteis para solucionar situações reais. Assim, melhorando a eficiência e a qualidade dos serviços em uma variedade de contextos do mundo real.

1.3 Objetivos

1.3.1 Objetivo Geral

Apresentar um algoritmo heurístico eficiente para a resolução do Problema do Caixeiro Viajante com Janelas de Tempo e o Problema de Caixeiro Viajante com Janelas de Tempo e Minimização do *Makespan*.

1.3.2 Objetivos específicos

- Apresentar estruturas de dados auxiliares para acelerar a busca local.

- Testar o algoritmo desenvolvido em instâncias da literatura.
- Comparar os resultados do algoritmo proposto com o melhor método encontrado na literatura.

1.4 Estrutura do Trabalho

Este trabalho é organizado da seguinte maneira:

- O Capítulo 1 consiste na introdução do trabalho e na definição do tema.
- O Capítulo 2 apresenta a fundamentação teórica e os trabalhos relacionados.
- O Capítulo 3 descreve o algoritmo construtivo, a busca local e os mecanismos de perturbação.
- O Capítulo 4 apresenta os experimentos computacionais.
- Por fim, o Capítulo 5 traz as considerações finais.

2 Fundamentação Teórica e Trabalhos Relacionados

2.1 Fundamentação Teórica

Esta seção visa apresentar algumas heurísticas e algoritmos construtivos utilizados na resolução de problemas de otimização. Embora seja impossível explorar exaustivamente todas as heurísticas disponíveis, concentraremos nossa atenção em alguns métodos relevantes. Para uma compreensão mais aprofundada e detalhada sobre meta-heurísticas e suas aplicações, recomenda-se a consulta ao livro Gendreau e Potvin (2018), que serve como um abrangente manual sobre o assunto.

2.1.1 Algoritmos Construtivos

Os algoritmos construtivos são uma classe de algoritmos de otimização que desenvolvem soluções incrementalmente, começando com uma solução vazia ou parcial e adicionando elementos ou componentes de forma sistemática. Esses algoritmos são valiosos quando a busca exaustiva por soluções não é prática devido a um espaço de busca extenso. Eles frequentemente utilizam heurísticas ou regras específicas para determinar como construir cada parte da solução, permitindo a criação de soluções aproximadas para problemas de otimização complexos.

A exemplo disso, a heurística do vizinho mais próximo constrói uma rota, começando sempre com um ponto de origem, conhecido como depósito. A cada passo ou interação do processo, um vértice é selecionado. A escolha é feita com base na proximidade, ou seja, o vértice selecionada é aquela que tem a menor distância em relação ao último vértice que foi adicionada à rota.

A abordagem de inserção mais barata inicia com uma sub-rota inicial que envolve três vértices. À medida que o processo avança, novos vértices são adicionados à rota. A decisão de onde inserir cada vértice é baseada no custo de inserção, ou seja, o vértice é posicionado entre dois vértices na solução em construção ou na solução parcial de maneira que resulte no menor aumento possível da distância total da rota.

2.1.2 Meta-heurísticas

Uma meta-heurística é uma estratégia algorítmica projetada para orientar o processo de busca por soluções ótimas em problemas de otimização complexos. Distinta dos algoritmos de otimização convencionais, que são tipicamente especializados para problemas

específicos, as meta-heurísticas se destacam pela sua aplicabilidade generalizada, podendo ser adaptadas e combinadas para atender a uma ampla variedade de cenários de otimização.

2.1.3 Meta-heurísticas Baseadas em Populações

As meta-heurísticas baseadas em populações são uma categoria de algoritmos de otimização que se inspiram em processos naturais, como evolução, colaboração social e competição para buscar soluções aproximadas para problemas complexos. Nesse contexto, múltiplas soluções candidatas são mantidas e evoluídas ao longo de iterações, permitindo que a busca se expanda em diferentes direções do espaço de soluções.

2.1.3.1 Algoritmos Genéticos

Algoritmos Genéticos (AGs) são uma classe de algoritmos de otimização inspirados no processo de seleção natural e evolução das espécies, conforme discutido em Holland (1984). A abordagem AG assume que as soluções para o TSP podem ser tratadas como indivíduos em uma população. Durante as iterações do algoritmo, novas soluções são geradas por operações de mutação e *crossover* (recombinação genética) entre os indivíduos existentes. Essas novas soluções são avaliadas por meio de uma função de *fitness*, que quantifica a qualidade de cada indivíduo. Os AGs utilizam uma seleção estocástica para escolher os melhores indivíduos para formar a próxima geração, permitindo que a busca pela solução ótima prossiga. Isso possibilita explorar o espaço de busca de maneira eficaz, evitando ficar preso em mínimos locais e encontrando soluções aproximadas para problemas complexos.

2.1.3.2 Algoritmo Baseado em Colônia de Formigas

O Algoritmo Baseado em Colônia de Formigas (ACO, do inglês “*Ant Colony Optimization*”), conforme descrito em Dorigo, Caro e Gambardella (1999), é uma classe de algoritmo de otimização inspirado pelo comportamento das formigas reais que buscam caminhos para encontrar comida. Os ACOs são amplamente utilizados para resolver problemas de otimização combinatória, baseando-se na ideia de que formigas virtuais constroem soluções explorando gradualmente caminhos e depositando feromônios nos arcos visitados. Esses feromônios representam a qualidade das soluções encontradas e orientam as decisões futuras das formigas virtuais, permitindo que elas colaborem na busca da solução ótima.

2.1.3.3 Algoritmo Baseado em Enxame de Partículas

Os Algoritmos Baseados em Enxame de Partículas (PSOs), inspirados no comportamento de enxames de pássaros ou peixes, como mencionado em Kennedy e Eberhart (1992), representam uma abordagem eficaz na otimização de problemas complexos, como o

TSP. Cada partícula no PSO simboliza uma solução potencial e se desloca pelo espaço de busca, objetivando a otimização da rota. A interação entre as partículas, que ajustam suas posições com base em suas experiências individuais e coletivas, é crucial para a eficácia do método. Dessa forma, o PSO procura encontrar a melhor solução possível, explorando extensivamente o espaço de soluções associado ao TSP.

2.1.4 Meta-heurísticas Baseadas em Busca Local

As meta-heurísticas baseadas em busca local são uma classe de algoritmos de otimização que se concentram em aprimorar soluções locais mediante refinamentos iterativos. Diferentemente das abordagens que exploram todo o espaço de soluções, essas meta-heurísticas concentram-se em uma solução inicial e buscam aprimorá-la localmente, explorando sua vizinhança. Normalmente é utilizado com um método de refinamento como o *Randomized Variable Neighborhood Descent* (RVND), introduzido em Subramanian et al. (2010a), emprega a abordagem sistemática para explorar múltiplas estruturas de vizinhança do VND Mladenović e Hansen (1997) de maneira aleatória. No RVND, diferentes vizinhanças são definidas, cada uma representando variações incrementais na solução corrente. A seleção aleatória dessas vizinhanças facilita a exploração diversificada do espaço de soluções, minimizando o risco de convergência prematura em ótimos locais. O processo iterativo do RVND continua até que nenhuma melhoria seja encontrada nas vizinhanças exploradas ou até que critérios predeterminados de parada sejam atingidos. Esta técnica é reconhecida por sua capacidade de gerar soluções de alta qualidade para problemas de otimização, balanceando eficazmente a busca local intensiva com a exploração abrangente do espaço de soluções.

2.1.4.1 Iterated Local Search

O *Iterated Local Search* (ILS), conforme descrito por Lourenço, Martin e Stützle (2019), é uma meta-heurística de otimização que se concentra em melhorar soluções locais iterativamente, explorando diferentes vizinhanças de uma solução inicial. A ideia fundamental por trás do ILS é iniciar com uma solução candidata por meio de um processo de construção. Em seguida, de maneira iterativa, uma perturbação é aplicada a essa solução com o intuito de escapar de ótimos locais. Após a perturbação, a busca é conduzida na vizinhança da solução previamente modificada. Esse processo repetido de perturbação e busca na vizinhança visa aprimorar progressivamente a solução encontrada, explorando eficazmente o espaço de busca em busca da melhor solução possível.

2.1.4.2 Greedy Randomized Adaptive Search Procedure

O *Greedy Randomized Adaptive Search Procedure* (GRASP - Procedimento de Busca Adaptativa Aleatorizada Gananciosa) é uma abordagem que combina elementos

de busca gulosa (*greedy*) com aleatoriedade, como discutido em Feo e Resende (1995). A ideia básica é iniciar com uma solução parcial vazia e, em cada passo, construir a solução adicionando um vértice à rota atual. A escolha do próximo vértice a ser adicionado é feita com base em uma combinação de critérios gulosos e aleatórios.

O aspecto guloso garante que a escolha do próximo vértice leve em consideração fatores como a distância entre o vértice atual e os vértices candidatos, priorizando aqueles mais próximos. No entanto, a aleatoriedade é introduzida para evitar que o algoritmo fique preso em ótimos locais subótimos. Isso é feito de forma controlada, permitindo que a escolha do próximo vértice seja influenciada por um certo grau de aleatoriedade.

O GRASP é uma heurística adaptativa, o que significa que ele pode ajustar seus parâmetros durante a busca para explorar diferentes regiões do espaço de soluções. Isso ajuda a evitar a estagnação em mínimos locais e a melhorar a qualidade das soluções encontradas.

2.1.4.3 *Simulated Annealing*

O *Simulated Annealing* (SA) é uma meta-heurística eficaz para otimização global, particularmente útil em espaços de busca extensos e quando o espaço de busca é discreto, como no Problema do Caixeiro Viajante, conforme discutido em Kirkpatrick, Gelatt e Vecchi (1983). Inspirado em um processo da metalurgia conhecido como “*annealing*”, que envolve o aquecimento e resfriamento controlado de um material para alterar suas propriedades físicas, o SA adota uma abordagem similar para aproximar o ótimo global de uma função.

A técnica probabilística do SA se caracteriza por uma desaceleração gradual na probabilidade de aceitar soluções piores durante a exploração do espaço de soluções. Esta característica torna o SA uma ferramenta poderosa para encontrar soluções próximas do ótimo global em problemas complexos de otimização.

2.1.4.4 *Tabu Search*

O *Tabu Search* (TS) utiliza mecanismos de memória para rastrear soluções já exploradas ou para aplicar regras específicas, conforme detalhado em Glover (1990). Se uma solução potencial foi recentemente explorada ou viola uma dessas regras, ela é marcada como “*tabu*”, evitando assim a repetição da mesma solução. Esta abordagem ajuda a navegar mais eficientemente pelo espaço de soluções, evitando ciclos repetitivos e promovendo uma exploração mais abrangente.

Para superar as limitações da busca local, que frequentemente fica presa em regiões de soluções subótimas ou platôs com soluções de qualidade similar, o *Tabu Search* modifica essa abordagem. Ele permite movimentos que podem deteriorar temporariamente a solução atual, uma estratégia particularmente útil em situações de mínimos locais.

2.2 Trabalhos Relacionados

Na literatura, o TSPTW é tratado de forma semelhante, porém os trabalhos que o mencionam têm sua principal contribuição nas nuances de suas estratégias de resolução. Savelsbergh (1992) propõe uma abordagem heurística ancorada em procedimentos de busca local. Dumas et al. (1995) elucidam uma estratégia resolutive fundamentada em princípios de programação dinâmica. Potvin e Bengio (1996) implementam uma abordagem fundamentada em Algoritmos Genéticos. Pesant et al. (1998) apresentam uma solução por meio do emprego do Algoritmo de Programação Lógica com Restrições Exatas. Gendreau et al. (1998) delineia uma metodologia resolutive empregando a heurística GENIUS, método dividido em duas partes: construção e otimização. Ohlmann e Thomas (2007) desenvolvem uma solução calcada na técnica de Simulated Annealing. Silva e Urrutia (2010) aplicam uma técnica baseada no General Variable Neighborhood Search (GVNS). López-Ibáñez e Blum (2010) introduzem uma estratégia resolutive que combina BEAM-search e um algoritmo fundamentado em colônia de formigas, denominado BEAM-ACO. López-Ibáñez et al. (2013) propõe uma solução utilizando o BEAM-ACO para abordar o problema TSPTW-M.

Subramanian et al. (2010b) introduzem o ILS-RVND para o Problema de Roteamento de Veículos com Coleta e Entrega Simultâneas (VRPSPD, do inglês, *Vehicle Routing Problem with Simultaneous Pickup and Delivery*). Silva et al. (2012) apresentam o GILS-RVND para o Problema da Latência Mínima, apresentando uma nova avaliação de complexidade amortizada $O(1)$, neste trabalho, o mesmo método será aplicado tanto para o TSPTW quanto para o TSPTW-M.

Um resumo dos trabalhos supracitados é apresentado na Tabela 1.

Tabela 1 – Resumo dos trabalhos relacionados, ano e método de resolução.

Autor(es)	Ano	Método de resolução
SAVELSBERGH	1992	Busca Local
DUMAS et al.	1995	Programação Dinâmica
POTVIN; BENGIO	1996	Algoritmo Genético
PESANT et al.	1998	Exato
GENDREAU et al.	1998	GENIUS
OHLMANN; THOMAS	2007	Simulated Annealing
SUBRAMANIAN et al.	2010b	ILS-RVND
SILVA; URRUTIA	2010	GVNS
LÓPEZ-IBÁÑEZ; BLUM	2010	BEAM-ACO
SILVA et al.	2012	GILS-RVND
LÓPEZ-IBÁÑEZ et al.	2013	BEAM-ACO

3 Procedimentos Metodológicos

3.1 Algoritmo Construtivo

O procedimento construtivo, algoritmo 1, pode ser visto como uma extensão do algoritmo do Vizinho Mais Próximo. Ao invés de selecionar consistentemente o vizinho imediatamente mais próximo, este método introduz uma abordagem ligeiramente diferente. Uma lista de candidatos é formada, consistindo dos n vizinhos mais próximos. A partir desta lista, um candidato é selecionado aleatoriamente, entre os α mais próximos, para ser adicionado à rota. Este procedimento foi baseado no artigo de Silva et al. (2012).

Algoritmo 1 Procedimento de Construção retirado de Silva et al. (2012)

```

1: procedimento CONSTRUÇÃO( $\alpha$ )
2:    $s \leftarrow \{0\}$ 
3:   Inicialize uma lista com os possíveis candidatos  $CL$ 
4:    $CL \leftarrow CL - \{0\}$ 
5:    $l \leftarrow 0$ 
6:   enquanto  $CL \neq \emptyset$  faça
7:     Ordene  $CL$  em ordem ascendente conforme a distância em relação a  $r$ 
8:     Escolha  $c$  aleatoriamente entre os  $\alpha\%$  melhores em  $CL$ 
9:      $CL \leftarrow CL - \{c\}$ 
10:     $s \leftarrow s \cup \{c\}$ 
11:     $l \leftarrow c$ 
12:  devolve  $s$ 

```

3.2 Busca local

O algoritmo GILS-RVND, apresentado no Algoritmo 2, combina técnicas de GRASP, ILS e RVND, formando uma busca local inspirada no trabalho de Silva et al. (2012). A operação do GILS-RVND se dá através de uma série de iterações, especificamente até que sejam alcançadas I_{MAX} iterações. Em cada iteração individual, ocorre o procedimento ILS, o qual continua até que se registre um total de I_{ILS} iterações sem qualquer melhoria.

Durante essas iterações, a solução inicial é construída com base no procedimento *Construtivo*, que usa um parâmetro de aleatoriedade α selecionado de um conjunto R . A solução resultante dessa construção é então submetida a uma busca local *RVND*, apresentado no Algoritmo 3. Se uma solução melhor é encontrada durante esta busca, a solução atual é atualizada e o contador de iterações ILS é reinicializado. Se não, uma perturbação é aplicada à solução para diversificar a busca, e o contador de iterações ILS é incrementado.

Ao final de todas as iterações, o GILS-RVND retorna a melhor solução encontrada durante sua execução. Esta solução é representada como “rota”, e a qualidade (ou custo) de cada solução é avaliada através da função objetivo f . A função objetivo difere entre as duas variantes do problema. Enquanto uma variante considera a soma dos tempos de travessia dos arcos, a outra, no TSPTW-M, leva em conta o tempo total, que inclui não apenas os tempos dos arcos, mas também os tempos de espera.

Algoritmo 2 Procedimento GILS-RVND retirado de Silva et al. (2012).

```

1: procedimento GILS-RVND( $I_{Max}$ ,  $I_{ILS}$ ,  $R$ )
2:    $f^* \leftarrow \infty$ 
3:   para  $i \leftarrow 1$  to  $I_{Max}$  faça
4:      $\alpha \leftarrow$  Escolha valor aleatório em  $R$ 
5:      $s \leftarrow$  Construção( $\alpha$ )
6:      $s' \leftarrow s$ 
7:      $iterILS \leftarrow 0$ 
8:     enquanto  $iterILS < I_{ILS}$  faça
9:        $s \leftarrow$  RVND( $s$ )
10:      se  $f(s) < f(s')$  então
11:         $s' \leftarrow s$ 
12:         $iterILS \leftarrow 0$ 
13:         $s \leftarrow$  Perturbar( $s'$ )
14:         $iterILS \leftarrow iterILS + 1$ 
15:      se  $f(s') < f^*$  então
16:         $s^* \leftarrow s'$ 
17:         $f^* \leftarrow f(s')$ 
18:   devolve  $s^*$ 

```

Algoritmo 3 Procedimento RVND retirado de Silva et al. (2012).

```

1: procedimento RVND( $s$ )
2:   Inicialize a Lista de Vizinhaça  $NL$ .
3:   Inicialize estruturas de dados de subsequências.
4:   enquanto  $NL \neq \emptyset$  faça
5:     Escolha uma vizinhaça  $N^{(\eta)}$  de  $NL$  aleatoriamente.
6:     Aplique a vizinhaça  $N^{(\eta)}$  em  $s$  encontrando um novo melhor vizinho  $s$ 
7:     se  $f(s') < f(s)$  então
8:        $s \leftarrow s'$ .
9:        $f(s) \leftarrow f(s')$ .
10:      Reinicialize  $NL$ .
11:      Atualize estruturas de dados de subsequências.
12:    senão
13:      Remova  $N^{(\eta)}$  de  $NL$ .
14:  devolve  $s$ 

```

3.2.1 Estruturas de Vizinhanças

O método *RVND* (Algoritmo 3) é a base para a realização da busca local, na qual se define t como a quantidade de estruturas de vizinhança, sendo $N = N^1, N^2, N^3, \dots, N^t$ o conjunto que as representa. Quando uma vizinhança particular de N não é capaz de melhorar a solução vigente, o *RVND* opta de forma aleatória por uma diferente vizinhança dentro do mesmo conjunto para continuar a busca. As estruturas de vizinhança implementadas para a resolução são descritas a seguir:

- **Swap** — $N^{(1)}$ — Dois clientes da solução são trocados de lugar.
- **Reinserção** — $N^{(2)}$ — Um cliente é realocado para outra posição da solução.
- **Or-opt2** — $N^{(3)}$ — Dois clientes adjacentes são realocados para outra posição da solução.
- **Or-opt3** — $N^{(4)}$ — Três clientes adjacentes são realocados para outra posição da solução.

As Figuras 1–4 ilustram as vizinhanças utilizadas.

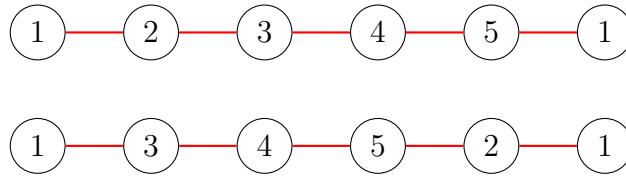


Figura 1 – Exemplo de reinserção em uma rota com 5 vértices, em que o vértice 2 é reinserido na posição 5.

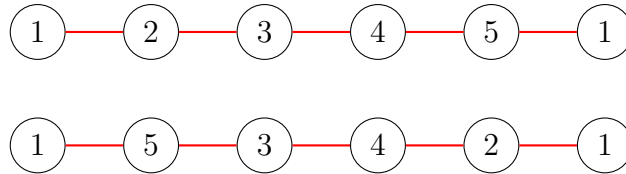


Figura 2 – Exemplo de Swap em uma rota com 5 vértices, em que o vértice 2 é trocado com o vértice 5.

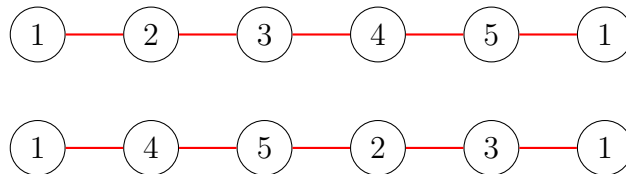


Figura 3 – Exemplo do Or-opt2 em uma rota com 5 vértices, em que os vértices 2 e 3 são reinseridos na posição 5.

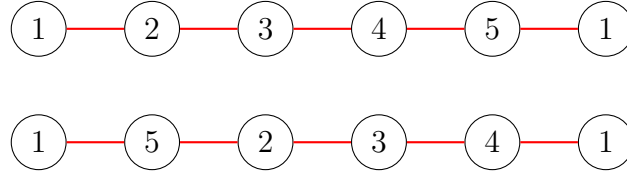


Figura 4 – Exemplo do Or-opt3 em uma rota com 5 vértice, em que os vértices 2, 3 e 4 são reinseridos na posição 5

3.2.2 Avaliação Eficiente

O algoritmo, ao verificar as restrições de janela de tempo, recorre a estruturas auxiliares baseadas em *subsequências*. Estas são formadas e renovadas em complexidade $\mathcal{O}(n^2)$. Através delas, é possível examinar tais restrições em complexidade $\mathcal{O}(1)$. Este processo de verificação e estruturas auxiliares é baseado nas estruturas propostas em Vidal et al. (2013) e Bulhões et al. (2018). Define-se:

- $C(\sigma)$: Custo para percorrer σ ;
- $TW(\sigma)$: Quantidade de tempo excedida do final da janela de tempo ou *time-warp* mínimo;
- $E(\sigma)$: Horário mais cedo de visita ao primeiro vértice tal que a duração e o *time-warp* sejam mínimos;
- $L(\sigma)$: Horário mais tarde de visita ao primeiro vértice tal que a duração e o *time-warp* sejam mínimos.

Ao analisar a inviabilidade, notamos que ela se dá pelo valor de TW. Um valor de TW igual a 0 indica que não há violação de nenhuma janela de atendimento. No entanto, quando TW é positivo, este valor é multiplicado por um grande número penalizador, o que consequentemente penaliza a função objetivo, tornando-se uma forma de indicar que a solução atual não é desejável devido à violação das janelas de tempo.

Usando o caso trivial onde a subsequência σ' é composta apenas pelo vértice i , temos que: $D(\sigma') = d_i$, $TW(\sigma') = 0$, $E(\sigma') = e_i$, $L(\sigma') = l_i$. Utilizando o operador \oplus para representar a concatenação de duas subsequências diferentes, observamos que uma dada subsequência σ , onde $|\sigma| > 1$, pode ser originada a partir de outras duas subsequências, σ_1 e σ_2 . A concatenação pode ser descrita pelas seguintes equações:

$$D(\sigma^1 \oplus \sigma^2) = D(\sigma^1) + D(\sigma^2) + t_{\sigma^1(|\sigma^1|)\sigma^2(1)} + \Delta_{WT} \quad (3.1)$$

$$E(\sigma^1 \oplus \sigma^2) = \max\{E(\sigma^2) - \Delta, E(\sigma^1)\} - \Delta_{WT} \quad (3.2)$$

$$L(\sigma^1 \oplus \sigma^2) = \min\{L(\sigma^2) - \Delta, L(\sigma^1)\} + \Delta_{TW} \quad (3.3)$$

$$TW(\sigma^1 \oplus \sigma^2) = TW(\sigma^1) + TW(\sigma^2) + \Delta_{TW}, \quad (3.4)$$

onde:

$$\Delta = D(\sigma^1) - TW(\sigma^1) + t_{\sigma^1(|\sigma^1|)\sigma^2(1)} \quad (3.5)$$

$$\Delta_{WT} = \max\{E(\sigma^2) - \Delta - L(\sigma^1), 0\} \quad (3.6)$$

$$\Delta_{TW} = \max\{E(\sigma^1) + \Delta - L(\sigma^2), 0\}. \quad (3.7)$$

3.3 Mecanismo de Perturbação

O procedimento de perturbação foi baseado no movimento *Double-Bridge* apresentado por Martin, Otto e Felten (1991). No movimento “Double-Bridge” essa estrutura da rota é desconstruída em segmentos separados em 4 subsequências aleatória. Na etapa subsequente, os segmentos separados são rearranjados de uma maneira específica, resultando em um novo padrão que destaca interações cruzadas entre os segmentos. A Figura 5 ilustra o mecanismo de perturbação utilizado.

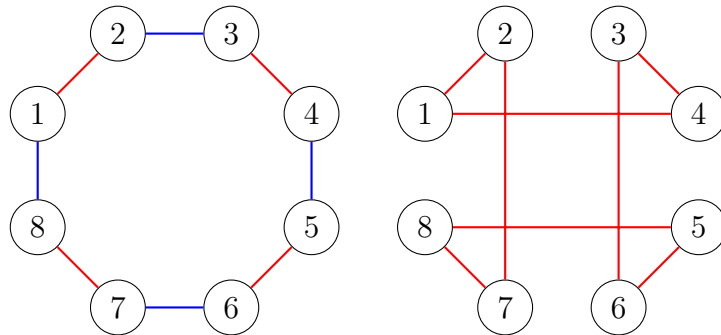


Figura 5 – Exemplo do Double-Bridge em uma rota com 8 clientes, em que arcos em azul são removidos e rearranjados

4 Experimentos Computacionais

O algoritmo proposto foi implementado em C++ (g++ 5.4.0) e todos os experimentos foram realizados em uma máquina com processador Intel i7-12700 com 4,9 GHz e 16 GB RAM e sistema operacional Linux Ubuntu 20.04.6 LTS. Durante os experimentos, utilizou-se a seguinte parametrização, baseada na proposta por Silva et al. (2012):

- $I_{Max} = 10$;
- $I_{ILS} = \min\{100, n\}$, onde n é o número de clientes;
- $R = \{0, 00, 0, 01, 0, 02, \dots, 0, 25\}$.

4.1 Resultados

As tabelas subsequentes exibem os resultados obtidos para uma variedade de instâncias. Nestas, é possível identificar a *Best Known Solution* (BKS), o custo ótimo alcançado, a média dos custos encontrados, bem como o tempo médio de execução. Os resultados destacados em negrito indicam as instâncias nas quais nossos resultados superaram os previamente estabelecidos. Os valores médios foram obtidos a partir de dez execuções distintas. Nas tabelas relacionadas ao TSPTW, além dos resultados da nossa heurística, estão inclusos também os da heurística GVNS, fornecidos para testes por Silva e Urrutia (2010). O gap é calculado usando a seguinte formula: $\text{Gap} = 100 \times (\text{Média} - \text{BKS}/\text{BKS})$

4.1.1 Resultados para o TSPTW

Os resultados apresentados nas Tabelas 2 e 4 equiparam-se ao BKS e à heurística GVNS em termos de qualidade, sem mostrar melhorias significativas tanto na média quanto nos melhores resultados. Na Tabela 3, observa-se um gap médio geralmente inferior ao do GVNS, com destaque para um resultado superior em uma das instâncias. A Tabela 5 exhibe resultados que ultrapassam o BKS e, na maioria das instâncias, apresentam uma média de qualidade superior. Por outro lado, diferentemente do GVNS, a heurística que propusemos levou mais tempo para identificar as soluções. Conforme evidenciado na Tabela 5, o tempo médio necessário foi 12 vezes maior em comparação ao tempo de execução do GILS-RVND. Enquanto isso, nas instâncias sugeridas por Gendreau et al. (1998), a diferença foi mínima de 0,2 segundos, conforme ilustrado na Tabela 3.

Tabela 2 – Resultados para TSPTW em instâncias propostas por Dumas et al. (1995)

Instância	BKS	GILS-RVND				GVNS		
		Melhor	Média	Gap (%)	Tempo (s)	Melhor	Média	Tempo (s)
n20w20	361,2	361,2	361,2	0,00	0,01	361,2	361,2	0,02
n20w40	316,0	316,0	316,0	0,00	0,01	316,0	316,0	0,03
n20w60	309,8	309,8	309,8	0,00	0,01	309,8	309,8	0,04
n20w80	311,0	311,0	311,0	0,00	0,01	311,0	311,0	0,04
n20w100	275,2	275,2	275,2	0,00	0,01	275,2	275,2	0,04
n40w20	486,6	486,6	486,6	0,00	0,06	486,6	486,6	0,06
n40w40	461,0	461,0	461,0	0,00	0,06	461,0	461,0	0,07
n40w60	416,4	416,4	416,4	0,00	0,06	416,4	416,4	0,09
n40w80	399,8	399,8	399,8	0,00	0,06	399,8	399,8	0,10
n40w100	377,0	377,0	377,0	0,00	0,07	377,0	377,0	0,12
n60w20	581,6	581,6	581,6	0,00	0,21	581,6	581,6	0,10
n60w40	590,2	590,2	590,2	0,00	0,23	590,2	590,2	0,14
n60w60	560,0	560,0	560,0	0,00	0,25	560,0	560,0	0,17
n60w80	508,0	508,0	508,0	0,00	0,26	508,0	508,0	0,23
n60w100	514,8	514,8	514,8	0,00	0,26	514,8	514,8	0,26
n80w20	676,6	676,6	676,6	0,00	0,54	676,6	676,6	0,17
n80w40	630,0	630,0	630,0	0,00	0,60	630,0	630,0	0,24
n80w60	606,4	606,4	606,4	0,00	0,64	606,4	606,4	0,35
n80w80	593,8	593,8	593,8	0,00	0,67	593,8	593,8	0,41
n100w20	757,6	757,6	757,6	0,00	1,13	757,6	757,6	0,27
n100w40	701,8	701,8	701,8	0,00	1,21	701,8	701,8	0,37
n100w60	696,6	696,6	696,6	0,00	1,27	696,6	696,6	0,53
n150w20	868,4	868,4	868,4	0,00	3,05	868,4	868,4	0,63
n150w40	834,8	834,8	834,8	0,00	3,24	834,8	834,8	1,04
n150w60	818,6	818,6	818,6	0,00	3,91	818,6	818,6	1,42
n200w20	1009,0	1009,0	1009,0	0,00	7,27	1009,0	1009,1	1,46
n200w40	984,2	984,2	984,2	0,00	8,02	984,2	984,2	2,20
Média	579,5	579,5	579,5	0,00	1,227	579,5	579,5	0,393

4.1.2 Resultados para o TSPTW-M

Para a variante TSPTW-M, a comparação foi realizada com base nos resultados apresentados por López-Ibáñez et al. (2013). No entanto, devido à inacessibilidade do código-fonte, não foi possível realizar uma comparação direta com o BEAM-ACO, como ocorreu na comparação com o GVNS para o TSPTW.

As Tabelas 6, 7, 8 e 9 demonstram resultados próximos ao BKS, sem grandes variações. Por outro lado, na Tabela 10, os resultados excedem o BKS, com um gap médio de $-0,00002$. Finalmente, na Tabela 11, embora o resultado médio tenha sido inferior, o melhor resultado obtido, em média, superou o BKS.

Finalmente, as Tabelas 12 e 13 reportam os resultados obtidos de forma agregada.

Tabela 3 – Resultados para TSPTW em instâncias propostas por Gendreau et al. (1998)

Instância	BKS	GILS-RVND				GVNS		
		Melhor	Média	Gap (%)	Tempo (s)	Melhor	Média	Tempo (s)
n20w120	265,6	265,6	265,6	0,00	0,01	265,6	265,6	0,06
n20w140	232,8	232,8	232,8	0,00	0,01	232,8	232,8	0,06
n20w160	218,2	218,2	218,2	0,00	0,01	218,2	218,2	0,06
n20w180	236,6	236,6	236,6	0,00	0,01	236,6	236,6	0,07
n20w200	241,0	241,0	241,0	0,00	0,01	241,0	241,0	0,07
n40w120	360,0	377,8	377,8	4,94	0,07	377,8	377,8	0,15
n40w140	348,4	364,4	364,4	4,59	0,07	364,4	364,4	0,17
n40w160	326,8	326,8	326,8	0,00	0,07	326,8	326,8	0,17
n40w180	326,8	330,4	330,4	1,10	0,08	330,4	330,6	0,20
n40w200	313,8	313,8	314,6	0,25	0,08	313,8	314,2	0,21
n60w120	451,0	451,0	451,0	0,00	0,26	451,0	451,0	0,29
n60w140	452,0	452,0	452,0	0,00	0,27	452,0	452,0	0,34
n60w160	448,6	464,0	464,0	3,43	0,24	464,0	464,4	0,35
n60w180	421,2	421,2	421,2	0,00	0,30	421,2	421,2	0,45
n60w200	427,4	427,4	427,4	0,00	0,27	427,4	427,4	0,49
n80w100	578,6	578,6	578,6	0,00	0,65	578,6	578,6	0,49
n80w120	541,4	541,4	541,4	0,00	0,71	541,4	541,4	0,55
n80w140	506,0	506,0	506,0	0,00	0,79	506,0	506,2	0,64
n80w160	502,8	504,8	504,8	0,40	0,72	504,8	505,2	0,68
n80w180	489,0	500,6	500,6	2,37	0,81	500,6	500,8	0,75
n80w200	481,4	481,4	481,8	0,07	0,82	481,8	481,8	0,85
n100w80	666,4	666,4	666,4	0,00	1,34	666,4	666,4	0,62
n100w100	642,0	640,6	641,3	-0,11	1,45	642,0	642,1	0,73
n100w120	597,2	597,2	597,3	0,02	1,57	597,2	597,3	0,84
n100w140	548,4	548,4	548,4	0,00	1,39	548,4	548,4	0,89
n100w160	555,0	555,0	555,0	0,00	1,55	555,0	555,0	1,03
n100w180	561,6	561,6	561,6	0,00	1,75	561,6	561,6	1,33
n100w200	550,2	550,2	551,2	0,18	1,83	550,2	550,2	1,40
Média	443,8	446,0	446,1	0,59	0,633	446,1	446,1	0,494

Tabela 4 – Resultados para TSPTW em instâncias propostas por Ohlmann e Thomas (2007)

Instância	BKS	GILS-RVND				GVNS		
		Melhor	Média	Gap (%)	Tempo (s)	Melhor	Média	Tempo (s)
n150w120	722,0	722,0	722,0	0,00	4,64	722,0	722,1	2,39
n150w140	693,8	693,8	694,1	0,04	4,70	693,8	693,9	2,63
n150w160	671,0	671,0	671,0	0,00	4,81	671,0	671,0	2,96
n200w120	803,6	803,6	803,6	0,00	9,38	803,6	803,7	4,66
n200w140	798,0	798,0	798,0	0,00	10,33	798,0	798,6	5,95
Média	737,7	737,7	737,7	0,00	6,772	737,7	737,9	3,718

Tabela 5 – Resultados para TSPTW em instâncias propostas por Silva e Urrutia (2010)

Instância	BKS	GILS-RVND				GVNS		
		Melhor	Média	Gap (%)	Tempo (s)	Melhor	Média	Tempo (s)
n200w100	10019,6	10019,6	10019,6	0,00	7,73	10019,6	10019,6	0,77
n200w200	9252,0	9252,0	9252,0	0,00	9,00	9252,0	9252,0	0,94
n200w300	8026,4	8022,8	8022,8	-0,04	12,22	8025,8	8030,8	1,20
n200w400	7067,2	7062,4	7062,78	-0,06	13,77	7068,2	7071,8	1,38
n200w500	6466,4	6466,2	6466,3	0,00	14,96	6466,4	6468,4	1,59
n250w100	12633,0	12633,0	12633,0	0,00	12,88	12633,0	12633,0	1,34
n250w200	11310,4	11310,4	11310,4	0,00	17,02	11310,4	11310,4	1,61
n250w300	10230,4	10230,4	10230,4	0,00	21,72	10230,4	10230,6	2,00
n250w400	8899,4	8896,2	8896,2	0,00	26,88	8897,8	8902,0	2,47
n250w500	8082,4	8065,8	8071,7	-0,13	29,51	8074,2	8083,4	2,67
n300w100	15041,2	15041,2	15041,2	0,00	20,51	15041,2	15041,2	2,22
n300w200	13846,8	13851,4	13851,4	0,03	27,38	13846,6	13852,1	2,58
n300w300	11477,6	11477,2	11477,3	0,00	38,45	11478,8	11483,5	4,05
n300w400	10413,0	10402,8	10402,9	-0,10	44,79	10411,0	10416,4	3,66
n300w500	9861,8	9842,2	9845,0	-0,17	49,13	9849,4	9861,6	4,06
n350w100	17494,0	17494,0	17494,0	0,00	30,04	17494,0	17494,0	3,44
n350w200	15672,0	15672,0	15672,0	0,00	44,49	15672,0	15672,0	4,20
n350w300	13650,2	13648,8	13649,3	0,00	87,26	13650,2	13651,9	5,38
n350w400	12099,0	12083,2	12086,3	-0,11	124,88	12096,2	12104,7	5,86
n350w500	11365,8	11347,8	11350,3	-0,14	78,25	11354,2	11368,8	5,67
n400w100	19454,8	19454,8	19454,8	0,00	41,79	19454,8	19454,8	5,02
n400w200	18349,9	18439,8	18439,8	0,49	58,50	18439,8	18439,8	6,01
n400w300	15873,4	15871,8	15871,8	-0,01	87,04	15872,2	15874,3	8,28
n400w400	14115,4	14079,4	14082,5	-0,23	115,69	14122,6	14129,9	8,48
n400w500	12747,6	12715,2	12721,0	-0,21	120,66	12739,4	12752,1	8,91
Média	12137,9	12135,2	12136,1	-0,01	45,38	12140,0	12143,9	3,75

Tabela 6 – Resultados para TSPTW-M em instâncias propostas por Ascheuer (1996)

Instância	BKS	Melhor	Média	Tempo (s)	Gap (%)
rbg016b	2094	2094	2094	0,01	0,00
rbg017,2	2351	2351	2351	0,00	0,00
rbg017a	4296	4296	4296	0,00	0,00
rbg019a	2694	2694	2694	0,01	0,00
rbg019d	3479	3479	3479	0,01	0,00
rbg020a	4689	4689	4689	0,01	0,00
rbg021,2	4528	4528	4528	0,01	0,00
rbg021,3	4528	4528	4528	0,01	0,00
rbg021,5	4516	4516	4516,4	0,01	0,01
rbg021,6	4492	4492	4493	0,01	0,02
rbg021,7	4481	4481	4481	0,01	0,00
rbg021,8	4481	4481	4481,6	0,01	0,01
rbg021,9	4481	4481	4481,6	0,01	0,01
rbg027a	5093	5093	5093	0,02	0,00
rbg031a	3498	3498	3498	0,02	0,00
rbg033a	3757	3757	3757	0,03	0,00
rbg034a	3314	3314	3314	0,03	0,00
rbg035a,2	3325	3325	3325	0,03	0,00
rbg035a	3388	3388	3388	0,04	0,00
rbg038a	5699	5699	5699	0,05	0,00
rbg040a	5679	5679	5679	0,06	0,00
rbg041a	3793	3793	3793	0,06	0,00
rbg042a	3260	3260	3260,5	0,08	0,02
rbg048a	9799	9799	9799	0,13	0,00
rbg049a	13257	13257	13257	0,12	0,00
rbg050b	11957	11957	11957	0,15	0,00
rbg050c	10985	10985	10985	0,12	0,00
rbg055a	6929	6929	6929	0,15	0,00
rbg067a	10331	10331	10331	0,27	0,00
rbg086a	16889	16899	16899	0,65	0,06
rbg125a	14214	14214	14214	1,71	0,00
rbg132,2	18524	18524	18524	1,69	0,00
rbg132	18524	18524	18524	1,83	0,00
rbg152	17455	17455	17455	2,63	0,00
rbg172a	17783	17783	17783,6	3,89	0,00
rbg193,2	21401	21401	21401	4,29	0,00
rbg193	21401	21401	21401	4,75	0,00
rbg201a	21380	21380	21380	5,31	0,00
rbg233,2	26143	26143	26143	6,79	0,00
rbg233	26143	26143	26143	7,63	0,00
Média	9375,8	9376,0	9376,1	1,066	0,00

Tabela 7 – Resultados para TSPTW-M em instâncias propostas por Dumas et al. (1995)

Instância	BKS	Melhor	Média	Tempo (s)	Gap (%)
n20w20	370.4	370.4	370.4	0.01	0,00
n20w40	342.8	342.8	342.8	0.01	0,00
n20w60	362.0	362.0	362.0	0.01	0,00
n40w20	521.2	521.2	521.2	0.07	0,00
n40w40	512.2	512.2	512.2	0.07	0,00
n40w60	481.4	481.4	481.4	0.07	0,00
n40w80	486.6	486.6	486.6	0.06	0,00
n40w100	463.0	463.0	463.0	0.06	0,00
n60w20	626.8	626.8	626.8	0.24	0,00
n60w60	672.8	672.8	672.8	0.24	0,00
n60w80	628.2	628.2	628.2	0.23	0,00
n60w100	620.2	620.2	620.48	0.23	0,05
n80w20	748.2	748.2	748.2	0.60	0,00
n80w60	712.6	712.6	712.6	0.56	0,00
n100w20	823.0	823.0	823.0	1.21	0,00
n100w40	821.0	821.0	821.0	1.14	0,00
n100w60	817.2	817.2	817.2	1.15	0,00
n150w20	978.4	978.4	978.4	3.13	0,00
n150w40	990.4	990.4	990.4	3.04	0,00
n150w60	981.4	988.6	988.6	3.11	0,73
n200w20	1137.8	1137.8	1137.8	6.33	0,00
n200w40	1156.0	1156.0	1156.0	6.41	0,00
Média	693.3	693.7	693.7	1.272	0,04

Tabela 8 – Resultados para TSPTW-M em instâncias propostas por Gendreau et al. (1998)

Instância	BKS	Melhor	Média	Tempo (s)	Gap (%)
n20w120	319,6	319,6	319,6	0,01	0,00
n20w140	286,2	286,2	286,2	0,01	0,00
n20w160	311,4	311,4	311,4	0,01	0,00
n20w200	281,8	281,8	281,8	0,01	0,00
n40w120	470,6	470,6	471,3	0,07	0,15
n40w140	458,2	458,2	458,2	0,07	0,00
n40w180	427,4	427,4	427,9	0,07	0,12
n60w120	573,8	573,8	573,8	0,23	0,00
n60w140	600,0	600,0	600,0	0,23	0,00
n60w160	619,6	619,6	619,6	0,23	0,00
n60w180	576,0	576,0	576,0	0,20	0,00
n60w200	570,2	570,2	570,2	0,22	0,00
n80w140	672,8	672,8	672,9	0,56	0,01
n80w160	653,6	653,6	656,1	0,60	0,38
n80w180	656,4	656,4	657,4	0,58	0,15
n80w200	646,2	646,2	646,2	0,57	0,00
n100w80	805,8	805,8	805,8	1,11	0,00
n100w100	795,8	795,8	795,8	1,09	0,00
n100w120	895,4	895,4	895,4	1,00	0,00
n100w140	906,4	906,4	906,4	0,99	0,00
n100w160	865,0	865,0	865,0	0,97	0,00
Média	590.1	590.1	590.3	0.420	0,04

Tabela 9 – Resultados para TSPTW-M em instâncias propostas por Ohlmann e Thomas (2007)

Instância	BKS	Melhor	Média	Tempo (s)	Gap (%)
n150w120.1	972	972	972	3,22	0,00
n150w120.2	917	917	917	3,04	0,00
n150w120.4	925	925	925	3,81	0,00
n150w120.5	907	907	907	3,30	0,00
n150w140.3	844	844	844	2,82	0,00
n150w140.4	898	898	898	2,84	0,00
n150w140.5	926	926	926	2,81	0,00
n150w160.1	959	959	959	2,71	0,00
n150w160.3	934	934	934	2,89	0,00
n150w160.5	920	920	920	2,74	0,00
n200w120.1	1089	1089	1089	6,75	0,00
n200w120.2	1072	1072	1072	5,37	0,00
n200w120.3	1128	1128	1128	6,46	0,00
n200w120.4	1072	1072	1072	5,71	0,00
n200w120.5	1073	1073	1073	6,28	0,00
n200w140.1	1138	1138	1138	6,79	0,00
n200w140.2	1087	1087	1087	6,09	0,00
n200w140.3	1083	1083	1083	6,51	0,00
n200w140.4	1100	1100	1100	6,66	0,00
n200w140.5	1121	1121	1121	6,67	0,00
Média	1008,3	1008,3	1008,3	4,674	0,00

Tabela 10 – Resultados para TSPTW-M em instâncias propostas por Pesant et al. (1998)

Instância	BKS	Melhor	Média	Tempo (s)	Gap (%)
rc201.0	853,71	853,70	853,70	0,02	-0,001
rc201.1	850,48	850,47	850,47	0,02	-0,001
rc201.2	883,97	883,95	883,95	0,03	-0,002
rc201.3	722,43	722,42	722,42	0,01	-0,001
rc202.0	850,48	850,47	850,47	0,01	-0,001
rc202.1	702,28	702,28	702,28	0,01	0,000
rc202.2	853,71	853,70	853,70	0,02	-0,001
rc202.3	883,97	883,95	883,95	0,02	-0,002
rc203.1	850,48	850,47	850,47	0,06	-0,001
rc204.0	839,24	839,23	839,23	0,02	-0,001
rc204.1	492,60	492,57	492,57	0,04	-0,006
rc204.2	870,52	870,51	870,51	0,14	-0,001
rc205.0	834,62	834,61	834,61	0,02	-0,001
rc205.1	899,24	899,23	899,23	0,01	-0,001
rc205.2	908,79	908,77	908,77	0,02	-0,002
rc205.3	684,21	684,19	684,19	0,02	-0,003
rc206.1	756,45	756,43	756,43	0,05	-0,003
rc206.2	776,19	776,17	776,17	0,05	-0,003
rc207.1	785,37	785,26	785,26	0,06	-0,014
rc208.0	836,04	835,92	836,30	0,14	0,031
rc208.1	615,51	615,42	615,42	0,04	-0,015
rc208.2	596,21	596,12	596,12	0,04	-0,015
Média	788,5	788,4	788,5	0,039	0,000

Tabela 11 – Resultados para TSPTW-M em instâncias propostas por Potvin e Bengio (1996)

Instância	BKS	Melhor	Média	Tempo (s)	Gap (%)
rc_201.1	592,06	592,05	592,05	0,01	0,00
rc_201.2	860,17	860,14	860,14	0,02	0,00
rc_201.4	889,18	889,16	889,16	0,02	0,00
rc_202.1	850,48	850,47	850,47	0,04	0,00
rc_202.2	338,52	338,51	338,51	0,00	0,00
rc_202.3	894,10	894,08	894,08	0,03	0,00
rc_203.1	488,42	488,41	488,41	0,01	0,00
rc_203.2	853,71	853,70	853,70	0,04	0,00
rc_203.3	921,44	921,42	921,42	0,08	0,00
rc_204.1	917,83	917,75	920,00	0,13	0,24
rc_204.3	455,03	454,95	454,95	0,02	0,00
rc_205.1	417,81	417,80	417,80	0,00	0,00
rc_205.2	820,19	820,17	820,17	0,02	0,00
rc_205.3	950,05	950,04	950,04	0,04	0,00
rc_205.4	837,71	837,68	837,68	0,02	0,00
rc_206.1	117,85	117,81	117,81	0,00	0,00
rc_206.2	870,49	870,45	871,72	0,08	0,14
rc_207.1	804,67	804,62	808,22	0,06	0,44
rc_207.2	713,90	713,84	713,84	0,05	0,00
rc_207.3	745,77	745,76	745,99	0,07	0,03
rc_207.4	133,14	133,14	133,14	0,00	0,00
rc_208.1	810,70	810,58	810,62	0,10	0,00
rc_208.2	579,51	579,45	579,45	0,04	0,00
rc_208.3	686,80	686,71	686,71	0,09	0,00
Média	689,60	689,50	689,80	0,04	0,03

Tabela 12 – Resultados agregados para TSPTW

Instâncias	BKS	GILS-RVND				GVNS		
		Melhor	Média	Gap (%)	Tempo (s)	Melhor	Média	Tempo (s)
DUMAS et al.	579,5	579,5	579,5	0,0000	1,227	579,5	579,5	0,393
GENDREAU et al.	443,8	446,0	446,1	0,5900	0,633	446,1	446,1	0,494
OHLMANN; THOMAS	737,7	737,7	737,7	0,0000	6,772	737,7	737,9	3,718
SILVA; URRUTIA	12137,9	12135,2	12136,1	-0,0148	45,38	12140,0	12143,9	3,750

Tabela 13 – Resultados agregados para TSPTW-M.

Instâncias	BKS	Melhor	Média	Tempo (s)	Gap (%)
ASCHEUER	9375,8	9376,0	9376,1	1,066	0,000
DUMAS et al.	693,3	693,7	693,7	1,272	0,040
GENDREAU et al.	590,1	590,1	590,3	0,420	0,040
OHLMANN; THOMAS	1008,3	1008,3	1008,3	4,674	0,000
PESANT et al.	788,5	788,4	788,5	0,039	0,000
POTVIN; BENGIO	689,6	689,5	689,8	0,040	0,029

5 Considerações finais

Este trabalho abordou o Problema do Caixeiro Viajante com Janelas de Tempo (TSPTW) e o Problema do Caixeiro Viajante com Janelas de Tempo e Minimização de Makespan (TSPTW-M). Foi apresentada uma abordagem heurística para a solução do problema usando a meta-heurística GILS-RVND. Para a avaliação das soluções encontradas durante a resolução do problema apresentou-se um método de avaliação eficiente.

Os resultados, testados em instâncias sugeridas por Silva e Urrutia (2010), Ascheuer (1996), Dumas et al. (1995), Gendreau et al. (1998), Ohlmann e Thomas (2007), Pesant et al. (1998) e Potvin e Bengio (1996), geralmente se equiparam ou superam as melhores soluções conhecidas e a heurística GVNS. Embora a maioria das instâncias mostre resultados próximos aos melhores já conhecidos, algumas instâncias específicas apresentam melhorias significativas, tanto em termos de média quanto nos melhores resultados alcançados.

Futuras pesquisas podem investigar o impacto da aplicação de diferentes algoritmos construtivos, bem como examinar a aplicação do método em outras variantes do problema que lidam com janelas de tempo.

Referências

- ASCHEUER, N. *Hamiltonian Path Problems in the On-line Optimization of Flexible Manufacturing Systems*. Tese (Doutorado), 1996.
- BULHÕES, T. et al. The static bike relocation problem with multiple vehicles and visits. *European Journal of Operational Research*, Elsevier, v. 264, n. 2, p. 508–523, 2018.
- DORIGO, M.; CARO, G. D.; GAMBARDELLA, L. M. Ant algorithms for discrete optimization. *Artificial Life*, v. 5, n. 2, p. 137–172, 1999.
- DUMAS, Y. et al. An optimal algorithm for the traveling salesman problem with time windows. *Operations Research*, v. 43, n. 2, p. 367–371, 1995. Disponível em: <<https://doi.org/10.1287/opre.43.2.367>>.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *J. Glob. Optim.*, Springer Science and Business Media LLC, v. 6, n. 2, p. 109–133, mar. 1995.
- GENDREAU, M. et al. A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research*, v. 46, n. 3, p. 330–335, 1998. Disponível em: <<https://doi.org/10.1287/opre.46.3.330>>.
- GENDREAU, M.; POTVIN, J.-Y. (Ed.). *Handbook of Metaheuristics*. 3. ed. Cham, Switzerland: Springer International Publishing, 2018. (International series in operations research & management science).
- GLOVER, F. Tabu search: A tutorial. *Interfaces*, Institute for Operations Research and the Management Sciences (INFORMS), v. 20, n. 4, p. 74–94, ago. 1990. ISSN 1526-551X. Disponível em: <<http://dx.doi.org/10.1287/inte.20.4.74>>.
- GUTIN, G.; PUNNEN, A. P. *The traveling salesman problem and its variations*. [S.l.]: Springer Science & Business Media, 2006. v. 12.
- HOLLAND, J. H. Genetic algorithms and adaptation. In: _____. *Adaptive Control of Ill-Defined Systems*. Boston, MA: Springer US, 1984. p. 317–333. ISBN 978-1-4684-8941-5. Disponível em: <https://doi.org/10.1007/978-1-4684-8941-5_21>.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization (pso). In: *Proc. IEEE International Conference on Neural Networks, Perth, Australia*. [S.l.: s.n.], 1942. p. 1942–1948.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983. Disponível em: <<https://www.science.org/doi/abs/10.1126/science.220.4598.671>>.
- LÓPEZ-IBÁÑEZ, M.; BLUM, C. Beam-aco for the travelling salesman problem with time windows. *Computers & Operations Research*, v. 37, n. 9, p. 1570–1583, 2010. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054809003116>>.

- LÓPEZ-IBÁÑEZ, M. et al. The travelling salesman problem with time windows: Adapting algorithms from travel-time to makespan optimization. *Applied Soft Computing*, v. 13, n. 9, p. 3806–3815, 2013. ISSN 1568-4946. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1568494613001658>>.
- LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search: Framework and applications. In: _____. *Handbook of Metaheuristics*. Cham: Springer International Publishing, 2019. p. 129–168. ISBN 978-3-319-91086-4. Disponível em: <https://doi.org/10.1007/978-3-319-91086-4_5>.
- MARTIN, O.; OTTO, S. W.; FELTEN, E. W. *Large-step Markov chains for the traveling salesman problem*. [S.l.]: Citeseer, 1991.
- MLADENović, N.; HANSEN, P. Variable neighborhood search. *Computers & Operations Research*, v. 24, n. 11, p. 1097–1100, 1997. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054897000312>>.
- OHLMANN, J. W.; THOMAS, B. W. A compressed-annealing heuristic for the traveling salesman problem with time windows. *INFORMS Journal on Computing*, v. 19, n. 1, p. 80–90, 2007. Disponível em: <<https://doi.org/10.1287/ijoc.1050.0145>>.
- PESANT, G. et al. An exact constraint logic programming algorithm for the traveling salesman problem with time windows. *Transportation Science*, v. 32, n. 1, p. 12–29, 1998. Disponível em: <<https://doi.org/10.1287/trsc.32.1.12>>.
- POTVIN, J.-Y.; BENGIO, S. The vehicle routing problem with time windows part ii: Genetic search. *INFORMS Journal on Computing*, v. 8, n. 2, p. 165–172, 1996. Disponível em: <<https://doi.org/10.1287/ijoc.8.2.165>>.
- SAVELSBERGH, M. W. P. The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, v. 4, n. 2, p. 146–154, 1992. Disponível em: <<https://doi.org/10.1287/ijoc.4.2.146>>.
- SILVA, M. M. et al. A simple and effective metaheuristic for the minimum latency problem. *European Journal of Operational Research*, v. 221, n. 3, p. 513–520, 2012. ISSN 0377-2217. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S037722171200269X>>.
- SILVA, R. F. da; URRUTIA, S. A general vns heuristic for the traveling salesman problem with time windows. *Discrete Optimization*, v. 7, n. 4, p. 203–211, 2010. ISSN 1572-5286. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1572528610000289>>.
- SUBRAMANIAN, A. et al. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, v. 37, n. 11, p. 1899–1911, 2010. ISSN 0305-0548. Metaheuristics for Logistics and Vehicle Routing. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054809002779>>.
- SUBRAMANIAN, A. et al. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, v. 37, n. 11, p. 1899–1911, 2010. ISSN 0305-0548. Metaheuristics for Logistics and Vehicle Routing. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054809002779>>.

VIDAL, T. et al. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, v. 40, n. 1, p. 475–489, 2013. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054812001645>>.