



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ANNE CAROLINE DA SILVA ROCHA

**APLICAÇÃO DO MAX-SAT PARCIAL PONDERADO PARA OTIMIZAÇÃO DO
PROBLEMA DO CAIXEIRO-VIAJANTE**

RUSSAS

2023

ANNE CAROLINE DA SILVA ROCHA

APLICAÇÃO DO MAX-SAT PARCIAL PONDERADO PARA OTIMIZAÇÃO DO
PROBLEMA DO CAIXEIRO-VIAJANTE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Alexandre Matos Arruda.

Coorientador: Prof. Dr. Dmontier Pinheiro Aragão Junior.

RUSSAS

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R571a Rocha, Anne Caroline da Silva.
Aplicação do Max-SAT Parcial Ponderado para Otimização do Problema do Caixeiro-Viajante / Anne Caroline da Silva Rocha. – 2023.
49 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Ciência da Computação, Russas, 2023.
Orientação: Prof. Dr. Alexandre Matos Arruda.
Coorientação: Prof. Dr. Dmontier Pinheiro Aragão Junior.

1. Otimização Combinatória. 2. Problema do Caixeiro-Viajante. 3. Max-SAT. I. Título.

CDD 005

ANNE CAROLINE DA SILVA ROCHA

APLICAÇÃO DO MAX-SAT PARCIAL PONDERADO PARA OTIMIZAÇÃO DO
PROBLEMA DO CAIXEIRO-VIAJANTE

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Russas da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em: 19/12/2023.

BANCA EXAMINADORA

Prof. Dr. Alexandre Matos Arruda (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Dmontier Pinheiro Aragão
Junior (Coorientador)
Universidade Federal do Ceará (UFC))

Prof. Dr. Bonfim Amaro Júnior
Universidade Estadual do Ceará (UECE)

AGRADECIMENTOS

À minha mãe, que por diversas vezes sozinha batalhou arduamente para que eu tivesse uma boa qualidade de vida apesar das dificuldades, que vendo fagulhas do que poderia se transformar em um grande potencial, investiu na minha educação e sempre que possível me presenteava com livros. Ao meu padrasto por ajudá-la a carregar a carga da casa e dos filhos. Ao meu irmão, Yuri, por me deixar sempre irritá-lo e ser implicante para que sempre pudesse me divertir um pouco, também agradeço pelos abraços excessivos, mesmo que por várias vezes eu não queria tanto grude. À pequena Sophia, por sempre me livrar do estresse com suas risadas de menina danada e pelo seu cheiro que me traz tanto conforto, por me curar sempre com seu amor.

Ao tio e padrinho, Naldo, pelo carinho e amor demonstrado, por assumir quando necessário o papel de figura paterna e ajudar minha mãe, você é uma pessoa de coração grande.

Aos meus amigos que tornaram cada dia dessa jornada mais suportável com o companheirismo, risadas, pelos bons e maus momentos juntos, por me fazerem sentir parte verdadeiramente de um grupo e por todo amor. Agradeço a Hanna, Dhioleno, Milene, Adryan, Natalia, João Victor, Lucas, Dárcio e Wilkinis por tudo. E ao meu colega, Thiago, porque segundo ele amigo é pai e mãe, por tanto somos apenas colegas já que não sou seu pai e nem sua mãe, mas ainda assim também sentirei sua falta. Também não posso deixar de destacar as contribuições de Juan Carlo, que durante as fases iniciais da graduação foi um grande amigo e apesar de não estarmos mais juntos na mesma universidade o carinho se mantém.

Ao meu companheiro de pesquisa João Pedro, pela paciência, por aguentar ir até tarde da noite programando ou escrevendo. Por sempre estar torcendo e acreditando em mim, pela incrível habilidade em me fazer rir nos momentos mais estressantes, pela companhia nas viagens apesar dos caminhos errados que pegamos e das velhas de bicicletas passando na frente do carro. Não tenho palavras suficientes para descrever meu carinho por você e nossa conexão.

Ao meu professor, orientador e acima de tudo grande amigo, Alexandre Arruda. Você vem me guiando na graduação muito antes de assumir o papel de orientador, esteve presente durante diferentes fases da minha vida aconselhando e apoiando. Pessoas como você são raras e sentirei muito a sua falta, contudo entendo que preciso de novos desafios e nem neles todos o terei como figura orientadora. Você me inspirou e continua me inspirando todos os santos dias.

Ao professor Dmontier Jr., um homem de imensa virtude e conhecimentos, sempre me impressiono com sua capacidade em trabalhar com diferentes temas e por vezes relacioná-los, além do entusiasmo contagiante. Reconheço cada um dos momentos que mesmo estando bastante

ocupado tirava parte do seu tempo para dedicar-se a este trabalho e ao artigo que produzimos juntos ao esclarecer cada dúvida e por cada sugestão.

Agradeço ao professor Bonfim Amaro por fazer parte desta banca e que apesar de não estar mais fazendo parte do campus de Russas, se dispôs a prestigiar o meu trabalho e fornecer suas contribuições. Também reconheço o seu papel na minha formação durante o período em que era meu professor, seu esforço e dedicação a docência são notáveis.

Ao Jonata, pela amizade e por ser um dos meus maiores apoiadores, acreditando em mim quando nem eu mesma acreditava. Agradeço pela sua sinceridade, paciência e o carinho.

À professora Rossana M. C. Andrade e Evilásio pelos ensinamentos que me foram repassados no período que estive trabalhando com ambos no GREat, parte do que vocês me ensinaram ou do caminho que me foi mostrado contribuíram para o desenvolvimento das habilidades de escrita que me foram muito úteis na execução deste trabalho. Também agradeço aos demais colegas que trabalhei conjuntamente no laboratório.

Por último, expresso minha gratidão aos demais que tiveram uma contribuição direta ou indireta na minha formação acadêmica.

"Você deve saber com quantos homens
incompetentes eu tive que competir - em vão."
(INGE LEHMANN.)

RESUMO

A Satisfação Booleana (*Boolean Satisfiability (SAT)*) é um problema com relevância tanto em termos teóricos quanto práticos. Os solucionadores SAT podem resolver uma variedade de problemas combinatórios de forma bem-sucedida na ciência da computação e inteligência artificial. Com os avanços recentes nas heurísticas dos solucionadores, muitos problemas industriais podem ser resolvidos de forma rápida e eficiente com a aplicação de solucionadores SAT, particularmente, o Max-SAT parcial ponderado desempenha um papel fundamental na solução de problemas da classe de otimização combinatória. O Problema do Caixeiro Viajante (*Travelling Salesman Problem (TSP)*) está entre os principais problemas estudados de otimização combinatória, em decorrência da sua aplicação em cenários reais que abrangem diferentes campos do conhecimento, como pesquisa operacional, eletrônica, matemática, engenharia, genética e ciência da computação. O objetivo é determinar o caminho de custo mínimo que visite todos os vértices e retornar ao ponto de partida. Deste modo, esta pesquisa objetiva propor um procedimento de pós-melhoria para o problema do caixeiro-viajante usando uma redução do problema para um Max-SAT parcial ponderado. Uma adaptação da heurística de Christofides e do algoritmo 2-Opt foram implementados para gerar um caminho inicial e melhorar o caminho inicial, a modelagem proposta é aplicada como um processo de aprimoramento posterior a melhoria já realizada pelo algoritmo 2-Opt. Um *benchmarking* com diferentes instâncias da biblioteca TSPLIB foram realizados para analisar os impactos da abordagem na qualidade da solução. Os resultados obtidos sugerem uma melhoria na qualidade da solução por um custo computacional razoável.

Palavras-chave: otimização combinatória; problema do caixeiro-viajante; Max-SAT.

ABSTRACT

Boolean Satisfiability (SAT) is a problem of great relevance in both theoretical and practical terms. SAT solvers can solve various combinatorial problems successfully in computer science and artificial intelligence. With recent advances in solver heuristics, many industrial problems can be solved quickly and efficiently with the application of SAT solvers, in particular, the weighted partial Maximum Satisfiability Problem (Max-SAT) plays a key role in solving problems of the combinatorial optimization class. The Traveling Salesman Problem (TSP) is among the main combinatorial optimization problems studied, due to its application in real-life scenarios covering different fields of knowledge, such as operations research, electronics, mathematics, engineering, genetics, and computer science. The aim is to determine the minimum cost route that visits all the vertices and returns to the starting point. Thus, this research aims to propose a post-improvement procedure for the traveling salesman problem using a reduction of the problem to a weighted partial Max-SAT. Adapting the Christofides solution and improving the initial route, the proposed modeling is applied as a post-improvement process to the improvement already carried out by the 2-Opt algorithm. Benchmarking with different instances of the TSPLIB library was carried out to analyze the approach's impact on the solution's quality. The results suggest an improvement in the quality of the solution at a reasonable computational cost.

Keywords: combinatorial optimization; travelling salesman problem; Max-SAT.

LISTA DE FIGURAS

Figura 1 – Percurso de Lincoln.	14
Figura 2 – Algoritmos adotados no TSP.	19
Figura 3 – Fluxograma do Algoritmo dos Vizinhos mais Próximos.	23
Figura 4 – Movimento 2-Opt.	25
Figura 5 – Procedimento de Pós-melhoria.	35
Figura 6 – Comparação do Custo para p01.	41
Figura 7 – Comparação do Custo para berlin52	41
Figura 8 – Comparação do Custo para gr21.	42
Figura 9 – Comparação do Custo para fri26.	42
Figura 10 – Comparação do Custo para att48.	43
Figura 11 – Comparação do Custo para pr76.	43

LISTA DE TABELAS

Tabela 1 – Especificações de Hardware	40
Tabela 2 – Resultados	44

LISTA DE QUADROS

Quadro 1 – Notação dos conectivos lógicos	26
Quadro 2 – Trabalhos Relacionados.	33

LISTA DE ABREVIATURAS E SIGLAS

aTSP	Asymmetric Travelling Salesman Problem
CDCL	Conflict-Oriented Clause Learning
CNF	Conjunctive Normal Form
CVRP-TW	Capacitated Vehicle Routing Problem with Time Windows
DFS	Depth-First Search
Max-SAT	Maximum Satisfiability Problem
MST	Minimum Spanning Trees
mTSP	Multi Travelling Salesman Problem
MWM	Minimum-Weight Matching
NN	Nearest Neighbor
RL	Reinforcement Learning
SAT	Boolean Satisfiability
sTSP	Symmetric Travelling Salesman Problem
TSP	Travelling Salesman Problem
VRP	Vehicle Routing Problem

SUMÁRIO

1	INTRODUÇÃO	14
2	OBJETIVOS	16
2.1	Objetivo geral	16
2.2	Objetivos específicos	16
3	FUNDAMENTAÇÃO TEÓRICA	17
3.1	Problema do Caixeiro-Viajante	17
3.2	Métodos Aproximativos	19
3.2.1	<i>Árvore Geradora Mínima</i>	20
3.2.1.1	<i>Heurística de Christofides</i>	21
3.3	Métodos Construtivos	21
3.3.1	<i>Algoritmo do Vizinhos Mais Próximos</i>	22
3.3.2	<i>Busca Gulosa</i>	22
3.4	Métodos de Melhoria	23
3.5	Método Composto	25
3.6	Problema da Satisfatibilidade Máxima	25
3.6.1	<i>Max-SAT Parcial Ponderado</i>	27
4	TRABALHOS RELACIONADOS	29
4.1	<i>Redução do SAT ao Problema do Caixeiro-Viajante</i>	29
4.2	<i>Redução do Problema do Roteamento de Veículos ao Max-SAT</i>	30
4.3	<i>Procedimento de melhoria com o algoritmo K-Opt</i>	32
4.4	<i>Análise dos trabalhos relacionados e o proposto</i>	33
5	METODOLOGIA	35
5.1	Conjunto de Dados	35
5.2	Construção da solução inicial	36
5.3	Procedimento de Melhoria com 2-Opt	36
5.4	Aumento do Espaço de Solução	36
5.5	Modelagem Lógica	37
6	RESULTADOS	40
6.1	Análise da Abordagem	40
7	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	45

REFERÊNCIAS 46

1 INTRODUÇÃO

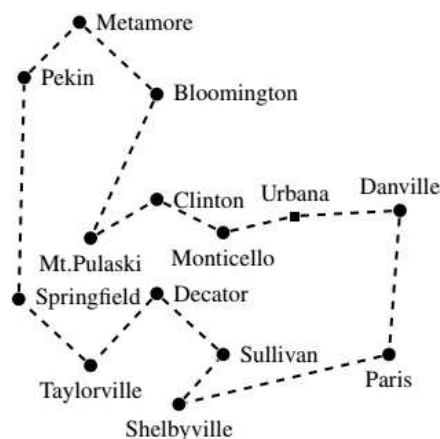
Diversos problemas de otimização podem ser formulados como problemas combinatórios, que buscam encontrar a melhor solução em um conjunto finito de soluções viáveis (SEMBA; MUJUNI, 2019). Os problemas de otimização combinatória surgem em diversas aplicações do mundo real e embora a maioria desses problemas sejam NP-difícil, precisam ser resolvidos considerando dois objetivos conflitantes: (i) uma solução de boa qualidade e (ii) encontrar a solução em um tempo razoável (HUERTA *et al.*, 2022).

Estudado desde do século XVIII (MATAI *et al.*, 2010), o Problema do Caixeiro-Viajante (*Traveling Salesman Problem (TSP)*) é descrito como um problema de permutação para encontrar o caminho de custo mínimo de modo que um vendedor visite todos os pontos de destino uma única vez (SATHYA; MUTHUKUMARAVEL, 2015; KOMARUDIN; PARHUSIP, 2019). A figura 1 apresenta um exemplo clássico do TSP conhecido como o percurso de Lincoln em 1850 (MATAI *et al.*, 2010; HAR-PELED, 2011).

De acordo com Komarudin e Parhusip (2019, p.87):

“O TSP tem muitas aplicações na área industrial, como manufatura, computação de dados, cadeia de suprimentos e logística. Por ser classificado como um problema NP-difícil, a solução de um TSP requer algoritmos complexos, pois o espaço de busca de soluções aumentará exponencialmente com base no número de cidades a serem abordadas.”

Figura 1 – Percurso de Lincoln.



Fonte: Har-Peled (2011).

Apesar de não existir uma relação direta entre os problemas do caixeiro-viajante e o da satisfazibilidade booleana, existem reduções que podem ser realizadas entre os problemas (BRONTS, 2014). E embora a redução do TSP para o SAT não ser empregada comumente na

literatura, conforme Reinelt (1991) realizar a redução de problemas para a satisfação booleana possibilita o aprimoramento do procedimento de solução dos problemas.

A Satisfação Booleana (*Boolean Satisfiability (SAT)*) é um problema significativo tanto na teoria quanto nas aplicações práticas (SOHANGHPURWALA *et al.*, 2017). A redução de problemas para o SAT é uma abordagem muito bem-sucedida para resolver problemas combinatórios difíceis em inteligência artificial e ciência da computação (RINTANEN, 2012). Os solucionadores SAT permitem a solução rápida e eficiente de problemas industriais, possuindo a satisfação booleana relevância para a ciência da computação, pois problemas podem ser representados na forma da satisfatibilidade ou em algumas de suas extensões (BATTITI, 2009; SOHANGHPURWALA *et al.*, 2017; HOSSEN; POLASH, 2021).

Como consequência do interesse teórico e prático, a extensão Max-SAT parcial ponderado desempenha um papel fundamental para os problemas de otimização combinatória (BATTITI, 2009; HOSSEN; POLASH, 2021; LARROSA *et al.*, 2008; ROCHA *et al.*, 2023), no qual problemas como roteamento (XU *et al.*, 2003), podem ser reduzidos ao Max-SAT ponderado para obter uma solução mais eficiente (LARROSA *et al.*, 2008).

Desta forma, o presente trabalho visa propor um método de otimização para o problema do caixeiro-viajante usando a redução do problema para o Max-SAT parcial ponderado, que atuará realizando pós-melhoria na solução melhorada.

Uma solução inicial será estabelecida com a utilização da árvore geradora mínima e aperfeiçoada pelo algoritmo 2-Opt. Considerando a rota inicial gerada, os K vizinhos mais próximos dos nós do caminho ou obtidos da matriz de distância serão incluídos ao percurso com a finalidade de aumentar o espaço de solução a ser explorado pelo solucionador SAT, que irá realizar a otimização final da rota. Instâncias da biblioteca TSPLIB (REINELT, 1991) e Burkardt (2019) foram utilizadas para averiguar o desempenho da modelagem proposta.

Os próximos capítulos serão organizados da seguinte forma: no capítulo 2 os objetivos gerais e específicos serão apresentando, no capítulo 3 fundamentação teórica é exposta, os trabalhos relacionados a esta pesquisa são apresentados e discutidos no capítulo 4, o detalhamento da metodologia adaptada será abordado no capítulo 5. Enquanto no capítulo 6 os resultados obtidos são descritos e no capítulo 7 as conclusões e trabalhos futuros são discutidos.

2 OBJETIVOS

2.1 Objetivo geral

Propor um procedimento de pós-melhoria para o Problema do Caixeiro-Viajante utilizando o Max-SAT Parcial Ponderado.

2.2 Objetivos específicos

1. Adaptar a heurística de Christofides (1976) para construção da solução inicial;
2. Elaborar uma modelagem lógica para pós-melhoria da solução produzida pela adaptação da heurística de Christofides (1976) e melhorada pelo 2-Opt.
3. Identificar *benchmarks* para avaliar o desempenho da abordagem proposta.
4. Analisar o impacto da abordagem proposta no custo da solução final.

3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão abordados os principais conceitos que nortearam o desenvolvimento desta pesquisa e permitem sua compreensão. Apresentando a definição do Problema do Caixeiro-Viajante, sua contribuição para diferentes problemas na ciência da computação e as principais estratégias algorítmicas apresentadas na literatura. Os conceitos de Problema da Satisfatibilidade Booleana, Max-SAT e Max-SAT Parcial Ponderado também serão explorados devido à relevância para este trabalho.

3.1 Problema do Caixeiro-Viajante

Os problemas de roteamento estão entre os paradigmas mais estudados nas atuais comunidades de pesquisa operacional e otimização (OSABA *et al.*, 2020), para os autores:

Dois motivos principais que tornam esse tópico primordial na área são (i) sua natureza prática inerente e seu interesse social, que permitem que os problemas de roteamento sejam aplicáveis não apenas em cenários de lazer ou turismo, mas também em situações relacionadas à logística e aos negócios, e (ii) sua complexidade, o que torna esses problemas muito difíceis de serem resolvidos de forma otimizada, mesmo para conjuntos de dados de tamanho médio. (OSABA *et al.*, 2020, p.136).

Dentre os problemas de roteamento, o Problema do Caixeiro-Viajante (*Traveling Salesman Problem (TSP)*) está entre os problemas de otimização combinatória mais estudados (OSABA *et al.*, 2020; FEILLET *et al.*, 2005). De acordo com Voudouris e Tsang (1999, p.469), "[...] o TSP desempenha um papel muito importante no desenvolvimento, teste e demonstração de novas técnicas de otimização", sendo possível identificar o progresso da otimização combinatória acompanhando as pesquisas associadas ao TSP (HELSGAUN, 2000).

O problema do caixeiro-viajante visa determinar o caminho de custo mínimo, dado um conjunto de cidades, no qual: (i) cada cidade é visitada apenas uma única vez e (ii) para cada par de cidades, a distância entre elas é conhecida (MATAI *et al.*, 2010; SATHYA; MUTHUKUMARAVEL, 2015; OUAARAB, 2020).

Formalmente, o problema, pode ser definido como um conjunto $C = (c_1, c_2, \dots, c_n)$ de cidades distintas, representadas como coordenadas (c_{ix}, c_{iy}) , o conjunto de arcos entre cada par de cidades $E = \{(c_i, c_j) : i, j \in \{1, 2, \dots, n\}\}$ e o custo d_{cicj} associado a cada arco (ZELINKA *et al.*, 2010; OUAARAB, 2020). Para cada cidade c_i , o percurso pode ser representado como uma permutação circular $\pi = (\pi(1), \pi(2), \dots, \pi(n))$, expressando a função objetivo 3.1 a minimização

do percurso almejada.

$$f(n) = \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(i)\pi(1)} \quad (3.1)$$

Na literatura, o custo a ser minimizado no problema do caixeiro-viajante é a distância determinada pelo cálculo da distância euclidiana (KIZILATEŞ; NURIYEVA, 2013). Embora a escolha de um caminho mais curto também proporcione ganhos de tempo para o vendedor, esse não é o objetivo principal desta versão problema do caixeiro-viajante (KIZILATEŞ; NURIYEVA, 2013). Neste trabalho, o custo do percurso também irá se referir a distância euclidiana.

O TSP possui diversas aplicações em cenários reais do cotidiano, aparecendo como subproblema ou redução de problemas nos mais variados setores. Tipicamente, aplicações de logística, rotas para transporte e fabricação de microchips costumam se beneficiar das contribuições elaboradas para o problema (FEILLET *et al.*, 2005; HUERTA *et al.*, 2022; ILAVARASI; JOSEPH, 2014).

Contudo, em seu formato tradicional, o TSP não é capaz de fornecer soluções viáveis para todas as aplicações do mundo real, ocasionando surgimento de novas variações do problema para corresponder às demandas e aos desafios enfrentados de cada cenário (ILAVARASI; JOSEPH, 2014). As variantes básicas tratam-se do problema do caixeiro-viajante simétrico (*Symmetric Travelling Salesman Problem (sTSP)*), assimétrico (*Asymmetric Travelling Salesman Problem (aTSP)*) e múltiplo (*Multi Travelling Salesman Problem (mTSP)*).

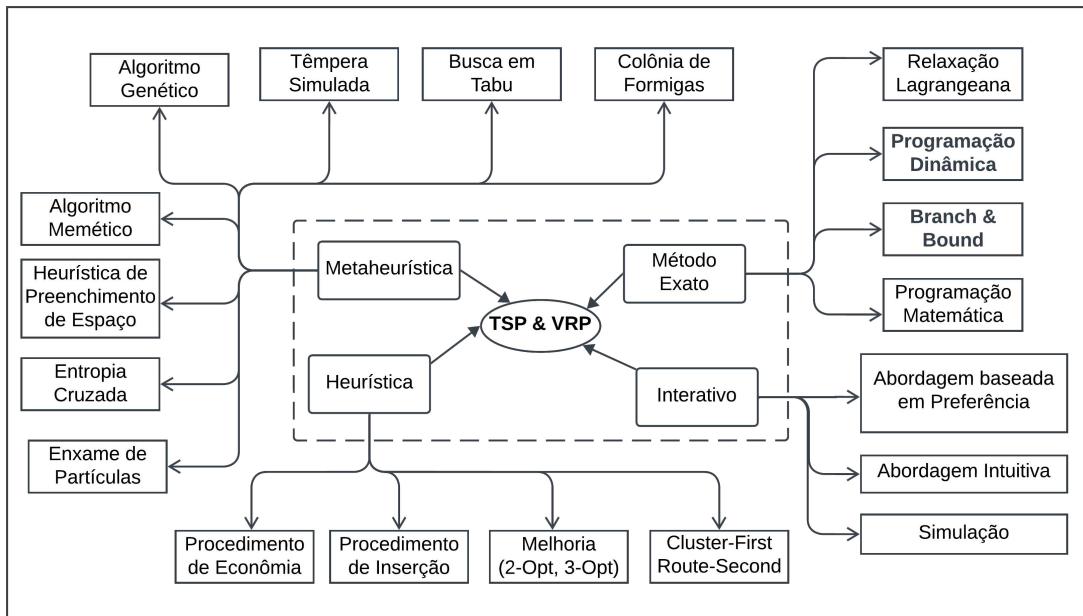
O problema do caixeiro-viajante simétrico busca encontrar um percurso de comprimento mínimo que visite cada cidade uma vez e retorne à cidade de origem, no qual o custo de viajar da cidade i para a cidade j e o custo da cidade j para a cidade i , são o mesmo em ambas as direções. Enquanto no problema do caixeiro-viajante assimétrico, os custos entre duas cidades pode variar conforme o ponto de partida (MATAI *et al.*, 2010; ARTHANARI, 2019; KOMARUDIN; PARHUSIP, 2019).

Para o problema do caixeiro-viajante múltiplo, um grupo m de caixeiros parte de depósito e deve visitar uma série de cidades intermediárias uma única vez, por tanto, o mTSP tem como objetivo elaborar roteiros para todos os m vendedores, iniciando e finalizando no depósito, de forma que cada nó intermediário seja visitado uma única vez e o custo total de visitar todos os nós, seja reduzido (MATAI *et al.*, 2010).

Como consequência da relevância prática dos problemas de otimização combinatória, diversos algoritmos foram propostas para solucioná-los. No geral, as abordagens dividem-se entre métodos exatos e aproximativos (SEMBA; MUJUNI, 2019).

Conforme Matai *et al.* (2010), dentre as principais abordagens para solucionar qualquer instância do TSP de forma otimizada, estão os métodos exatos. Contudo, como a complexidade aumenta exponencialmente com o número de cidades, utilizar métodos exatos para otimização do problema mesmo para instâncias de tamanho moderado requer um custo computacional alto (KONSTANTAKOPOULOS *et al.*, 2022).

Figura 2 – Algoritmos adotados no TSP.



Fonte: Adaptado de Anbuudayasankar *et al.* (2014, p.13).

Dada as limitações dos métodos exatos, as pesquisas voltaram-se para investigar as contribuições que a utilização de algoritmos de natureza aproximativa e heurística poderia trazer. De modo que houvesse na literatura a lacuna para a aplicação e o desenvolvimento de algoritmos que exploram diferentes abordagens (OUAARAB, 2020; MATAI *et al.*, 2010). A figura 2 exhibe os principais algoritmos adotados para resolver o TSP.

Nas próximas seções, diferentes abordagens heurísticas e aproximativas aplicadas na literatura serão apresentadas, contudo, para uma revisão da literatura mais abrangente verifique os trabalhos de Ouarab (2020), Anbuudayasankar *et al.* (2014), Matai *et al.* (2010) e Sathya e Muthukumaravel (2015).

3.2 Métodos Aproximativos

Nos métodos exatos existe a garantia de que uma solução ótima seja encontrada, contudo, para a classe de problemas NP-difícil o custo computacional requerido é muito alto,

justificando a baixa adesão de algoritmos exatos (HELSSGAUN, 2000; SEMBA; MUJUNI, 2019; KONSTANTAKOPOULOS *et al.*, 2022). Aplicado ao TSP, os métodos exatos possuem uma resolução computacionalmente complexa, limitados a pequenas instâncias do problema (ANBUUDAYASANKAR *et al.*, 2014; SEMBA; MUJUNI, 2019).

“Em contraste, os algoritmos aproximados obtêm boas soluções, mas não garantem que soluções ótimas serão encontradas. Esses algoritmos geralmente são muito simples e têm tempos de execução (relativamente) curtos” (HELSSGAUN, 2000, p.108). O objetivo é produzir soluções com equilíbrio entre a qualidade da solução e o custo computacional necessário para obtê-la e apesar de não conseguirem encontrar o ótimo, esse tipo de abordagem costuma ser apenas alguns por cento diferente da solução ótima (HELSSGAUN, 2000; MATAI *et al.*, 2010).

Os algoritmos aproximativos são adequados para situações em que valores próximos ao ótimo são aceitos, pois uma pequena melhoria na qualidade da solução ou redução do tempo de execução poderá resultar na economia de milhões de dólares, ou um aumento significativo na produtividade (HELSSGAUN, 2000; MATAI *et al.*, 2010; HUERTA *et al.*, 2022).

3.2.1 *Árvore Geradora Mínima*

O Problema da Árvore Geradora Mínima (*Minimum Spanning Trees (MST)*) consiste em encontrar uma árvore com peso total mínimo que abranja todos os vértices de um grafo G , considerando uma rede conectada com n nós e os custos associados a todas as arestas. No qual $G = (V, E, w)$ é um grafo conectado, ponderado, não direcionado composto por um conjunto V de vértices, E de arestas e a distância euclidiana w (GASS; FU, 2013; PETTIE, 2008).

Conforme Graham e Hell (1985), o MST é considerado um dos problemas mais conhecidos da otimização combinatória e desempenha um papel importante para o desenvolvimento de algoritmos dessas áreas. Utilizado com frequência em algoritmos aproximativos aplicados ao problema do caixeiro-viajante (GRAHAM; HELL, 1985), Helssgaun (2000) e Christofides (1976) implementaram a árvore geradora mínima como parte do método algorítmico usado em seus trabalhos.

Em Helssgaun (2000) a heurística de Lin-Kernighan (LIN; KERNIGHAN, 1973) foi modificada para o sTSP. A adaptação realizada pelo autor utiliza a análise de sensibilidade com intuito de direcionar e restringir as buscas, para melhorar as chances de que a ligação entre determinados nós faça parte do caminho ótimo, denominando-se de medida de proximidade (HELSSGAUN, 2000). “A medida de proximidade baseia-se na análise de sensibilidade usando

árvore geradora mínima” Helsgaun (2000, p.111).

Em consequência da relevância que o algoritmo de Christofides (1976) possui para o desenvolvimento desta pesquisa, uma subseção 3.2.1.1 será destinada à descrição detalhada do método aproximativo desenvolvido em Christofides (1976).

3.2.1.1 Heurística de Christofides

A maioria das heurísticas visa assegurar uma solução viável ou próxima do ótimo, contudo Christofides conseguiu ampliar a abordagem das heurísticas com taxas polinomiais de crescimento de variáveis (MATAI *et al.*, 2010), ao realizar o cálculo da árvore geradora mínima do grafo G (utilizado para representar o TSP) e a descoberta de uma correspondência com custo mínimo ótimo de um subgrafo de G (CHRISTOFIDES, 1976).

Matai *et al.* (2010) descreve as etapas do procedimento heurístico de Christofides em:

1. Construir uma árvore de abrangência geradora mínima (*MST*) a partir do conjunto de todas as cidades;
2. Criar uma correspondência de peso mínimo (*Minimum-Weight Matching (MWM)*) no conjunto de nós de grau ímpar. Adicione a *MST* junto com a *MWM*;
3. Criar um ciclo de Euler a partir do grafo combinado e percorrê-lo usando atalhos para evitar os nós já visitados.

3.3 Métodos Construtivos

Os métodos construtivos fazem com que passeios sejam criados gradualmente, inserindo uma cidade nova a cada etapa (HELSGAUN, 2000). Neste tipo de procedimento as rotas são geradas considerando as informações da matriz de distância, de modo que percurso final seja uma aproximação do custo ótimo (ANBUUDAYASANKAR *et al.*, 2014).

No contexto do TSP, os algoritmos construtivos atuam buscando iterativamente a cidade mais próxima que ainda não pertence ao percurso construído ou por meio de uma seleção, também iterativa, dos arcos mais curtos para a solução atual sem que o percurso seja fechado (MATAI *et al.*, 2010; OUAARAB, 2020). Nesse tipo de abordagem a solução é encontrada, o algoritmo é encerrado e nenhum tipo de melhoria é aplicada a solução (OUAARAB, 2020).

De acordo com Ouaarab (2020, p.89):

“[...] o principal objetivo desses métodos não é buscar a melhor solução, mas o cálculo reduzido, mesmo que a qualidade da solução seja moderada. Apesar de sua qualidade de solução gerada, os algoritmos de construção permanecem amplamente utilizados, mas como auxiliares em outros métodos.”

O algoritmo dos vizinhos mais próximo de Rosenkrantz *et al.* (1977), a busca gulosa e procedimento de economia de Clarke e Wright (1964) são os principais procedimentos construtivos empregados no problema do caixeiro-viajante.

3.3.1 Algoritmo do Vizinhos Mais Próximos

Considerado o algoritmo mais simples de construção de roteiros, o algoritmo do vizinho mais próximo (*Nearest Neighbor (NN)*) foi um dos primeiros métodos construtivos usados para encontrar uma solução para o problema do caixeiro-viajante (ALSALIBI *et al.*, 2012; HUERTA *et al.*, 2022). O algoritmo inicia em uma cidade escolhida aleatoriamente, em seguida a cidade mais próxima não visitada é adicionada ao percurso. O processo é repetido até que todas as cidades tenham sido visitadas e seja possível retornar à cidade de partida (ALSALIBI *et al.*, 2012; KIZILATEŞ; NURIYEVA, 2013).

Huerta *et al.* (2022) e Kizilateş e Nuriyeva (2013) discutem que apesar da facilidade de implementação e a rapidez do NN, as rotas encontradas, em geral, estão muito distantes do valor ideal, devido sua natureza gulosa. Conforme os autores, como o algoritmo do vizinho mais próximo apresenta o comportamento de que as primeiras cidades de um percurso possuam uma proximidade entre si, enquanto as demais encontram-se mais afastas, o algoritmo acaba deixando de construir rotas de custos menores. A figura 3 emprega a notação de fluxograma para evidenciar o comportamento do algoritmo.

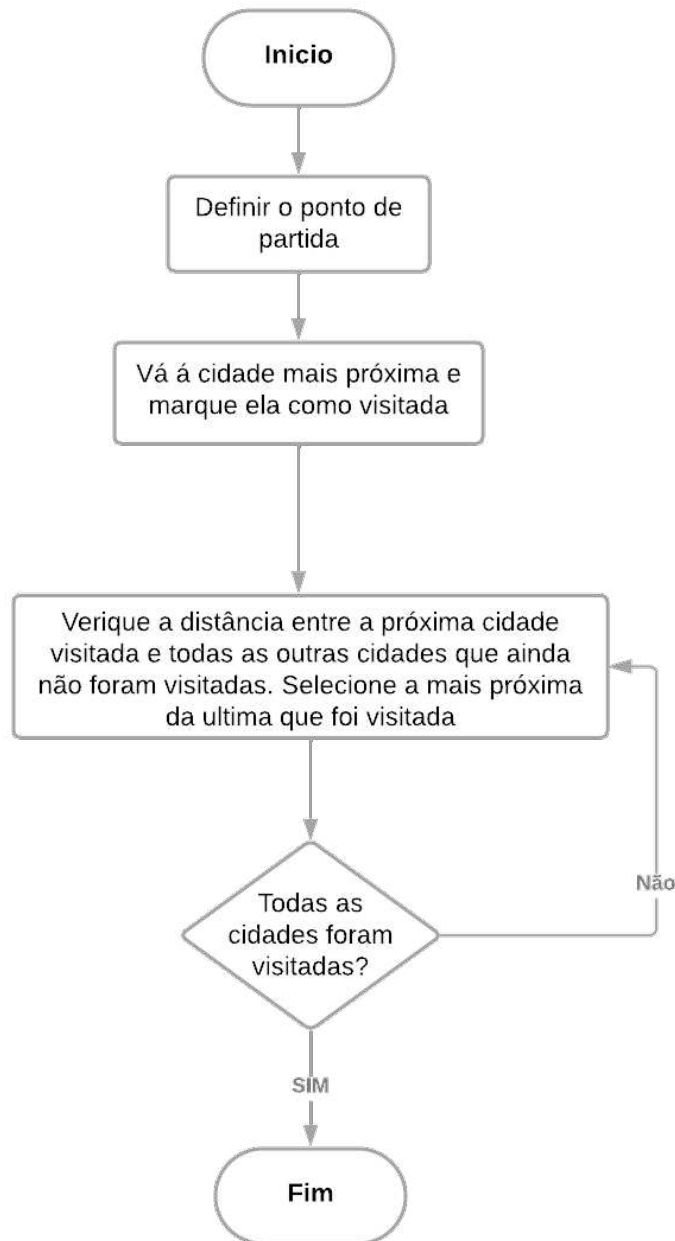
3.3.2 Busca Gulosa

A busca gulosa constrói um caminho selecionando gradualmente a aresta mais curta e adicionando-a ao percurso repetidamente, desde que não seja criado um ciclo com menos de n arestas ou aumente o grau de qualquer nó para mais de 2. Além disso, uma mesma borda não poderá ser utilizado para construção do percurso mais do que uma única vez (MATAI *et al.*, 2010).

De acordo com Matai *et al.* (2010), as etapas da busca gulosa são:

1. Classificar todas as bordas;
2. Selecionar a borda mais curta e adicioná-la ao tour, se não houver quebra das restrições;

Figura 3 – Fluxograma do Algoritmo dos Vizinhos mais Próximos.



Fonte: Adaptado de Alsalibi *et al.* (2012, p.235).

3. O percurso possui n arestas? Se não, repita a etapa 2.

3.4 Métodos de Melhoria

“As heurísticas de aprimoramento de pesquisa local são procedimentos de pesquisa iterativa que, partindo de uma solução inicial viável (geralmente o resultado de uma heurística construtiva), melhoram progressivamente a qualidade da solução aplicando uma série de modificações locais, chamadas de movimentos” (TARANTILIS *et al.*, 2005, p.457).

Iniciando com uma pequena rota inicial, escolhida de forma aleatória ou por um método construtivo, o algoritmo analisa todas as rotas vizinhas e busca encontrar uma rota que seja mais curta que a original, caso não exista, o procedimento é encerrado (ANBUUDAYASANKAR *et al.*, 2014). De forma que o percurso seja construído passo a passo, porém com a viabilidade da solução assegurada (ANBUUDAYASANKAR *et al.*, 2014).

Ouarab (2020) afirma que para os procedimentos de construção de roteiro, o problema do caixeiro-viajante é visto como um conjunto de cidades que devem ser empilhadas para formar um ciclo, enquanto para os métodos de melhoria, o TSP trata-se de um conjunto de rotas a serem “perturbados” para que uma configuração ideal seja estabelecida.

Os algoritmos 2-Opt (LIN, 1965), 3-Opt (ALFA *et al.*, 1991), Lin-Kernighan (LI; CHOW, 2021) e Or-Opt (OR, 1976; TARANTILIS *et al.*, 2005) são exemplos dos principais algoritmos de melhoria adotados na literatura.

Extensão do k-Opt, o algoritmo 2-Opt procura repetidamente por falhas na rota e executa as operações de inversão adequadas para removê-las (APPLEGATE *et al.*, 2003). O procedimento remove aleatoriamente duas arestas do percurso já gerado e reconecta os dois novos caminhos criados, realizando movimentos de duas opções (MATAI *et al.*, 2010). No qual:

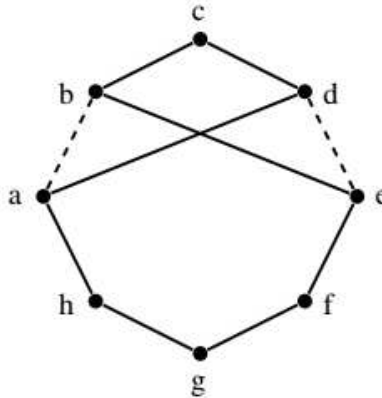
“Dado um passeio T e duas arestas $\{a, b\}$ e $\{x, y\}$ em T , existem duas possibilidades de substituir essas duas arestas por outras duas arestas. Podemos escolher o par $\{a, x\}$ e $\{b, y\}$ ou podemos escolher o par $\{a, y\}$ e $\{b, x\}$. Exatamente um desses dois pares resultará em um passeio novamente.” (BRODOWSKY; HOUGARDY, 2020, p.3).

O procedimento de troca ramal é realizado apenas se o novo passeio for mais curto que o percurso anterior, ocorrendo repetidamente até que nenhuma melhoria adicional seja possível, resultando em um percurso dois ótimo (MATAI *et al.*, 2010).

Prodhon e Prins (2016) justificam a maior adesão ao algoritmo 2-Opt como consequência de que para uma troca k de arestas, todos os movimentos possíveis serão testados $O(n^k)$ para n clientes, por tanto troca ramais de duas arestas mantém a complexidade baixa em relação aos demais procedimentos de melhorias.

Matai *et al.* (2010) destacam que o desempenho do 2-Opt está relacionado a qualidade do percurso produzido pelo procedimento construtivo adotado para geração do percurso. A figura 4 demonstra um exemplo de movimentação no procedimento 2-Opt.

Figura 4 – Movimento 2-Opt.



Fonte: Adaptado de Ouaarab (2020, p.93).

3.5 Método Composto

Um algoritmo composto trata-se de um modelo de otimização heurística que combina recursos de duas classes diferentes de algoritmos: os métodos construtivos e os de melhoria (KOMARUDIN; PARHUSIP, 2019; HELSGAUN, 2000). As rotas neste tipo de procedimento são construídas utilizando-se de algoritmos de construção conhecidos da literatura e são aprimoradas pelo uso de um ou mais procedimentos de melhoria com o objetivo de elevar a qualidade da solução gerada (ANBUUDAYASANKAR *et al.*, 2014; BARAGLIA *et al.*, 2000; CHIANG; RUSSELL, 1997).

3.6 Problema da Satisfatibilidade Máxima

Problema de satisfação de restrição, a Satisfação Booleana (*Boolean Satisfiability SAT*) é um problema que consiste em determinar se uma fórmula proposicional pode ser satisfeita por uma atribuição lógica de variáveis (ROIG, 2013; HOSSEN; POLASH, 2021).

No qual uma fórmula proposicional φ , é composta pelo conjunto $V = \{x_1, x_2, \dots, x_n\}$ de literais que podem assumir os valores verdadeiro ou falso e conectivos lógicos (ver tabela 1) (VIZEL *et al.*, 2015). Na satisfatibilidade booleana as fórmulas proposicionais são representadas na Forma Normal Conjuntiva (*Conjunctive Normal Form (CNF)*). Correspondendo a uma conjunção de um conjunto de cláusulas $\{C_1, C_2, \dots, C_n\}$, em que cada uma dessas cláusulas são disjunções de literais (SOHANGHPURWALA *et al.*, 2017; HOSSEN; POLASH, 2021), ou seja:

$$F = C_1 \vee C_2 \vee \dots \vee C_n \quad (3.2)$$

Apesar de ser um problema NP-completo, os progressos nos métodos de solução

Quadro 1 – Notação dos conectivos lógicos

Conectivos	Descrição
\wedge	Conjunção
\vee	Disjunção
\Rightarrow	Implicação
\Leftrightarrow	Bi-implicação
\neg	Negação

Fonte: Elaborado pelo autor.

e heurísticas permitem que problemas industriais pudessem ser resolvidos de forma rápida e eficiente (SOHANGHPURWALA *et al.*, 2017; ROIG, 2013). De forma que “uma ampla gama de problemas em diferentes áreas da ciência da computação e inteligência artificial pode ser resolvida usando solucionadores SAT” (HOSEN; POLASH, 2021, p.2).

A formulação matemática do SAT é composta por um número n de variáveis, a quantidade m de cláusulas e o número total de literais $|X_i|$ para a cláusula X_i , tendo como literal l_{ki} (BATTITI, 2009), de modo que a fórmula seja:

$$\bigwedge_{1 \leq i \leq m} \left(\bigvee_{1 \leq k \leq |X_i|} l_{ik} \right) \quad (3.3)$$

De acordo com Roig (2013), o desenvolvimento dos procedimentos do solucionador SAT proporcionou o surgimento de novas áreas de pesquisas, entre elas o problema da satisfatibilidade máxima. O Problema da satisfatibilidade Máxima (*Maximum Satisfiability Problem Max-SAT*) possui a mesma formulação em CNF que o SAT, contudo, o objetivo neste caso é encontrar uma atribuição de valores verdadeiros para as variáveis que satisfaça o número máximo de cláusulas (BATTITI; PROTASI, 1998; BATTITI, 2009). Operando os solucionadores do Max-SAT com blocos de cláusulas em alternativa a cláusulas individuais que exploram as novas estruturas das codificações (ARGELICH; MANYÀ, 2006).

Ambos os problemas, Max-SAT e SAT, são considerados interessantes e de extrema relevância tanto em termos teóricos e práticos (SOHANGHPURWALA *et al.*, 2017; BATTITI, 2009; HOSEN; POLASH, 2021). E apesar de serem NP-completo, os progressos nos métodos de solução e heurísticas permitiram que solucionadores de softwares modernos se aproveitarem desses recursos para solucionar problemas industriais. Recentemente, a tecnologia SAT foi aplicada para resolver combinatórios industriais (SOHANGHPURWALA *et al.*, 2017; ROIG, 2013).

“[...] muitos problemas em lógica matemática e inteligência artificial podem ser

expressos na forma de satisfatibilidade ou em algumas de suas extensões, como a satisfação de restrições ” (BATTITI, 2009, p.2036). Pois, existem algumas técnicas básicas e métodos que podem ser aplicados também ao Max-SAT (BATTITI; PROTASI, 1998), correspondendo “ a redução ao SAT é uma abordagem muito bem-sucedida para resolver problemas combinatórios difíceis em Inteligência Artificial e ciência da computação em geral” (RINTANEN, 2012, p.1).

3.6.1 Max-SAT Parcial Ponderado

Versão otimizada do problema da satisfatibilidade, o Max-SAT ponderado é uma extensão do Max-SAT que visa encontrar uma atribuição completa com custo mínimo, dada uma fórmula ponderada (LARROSA *et al.*, 2008). Diversos problemas acadêmicos e de aplicações práticas podem ser expressos como um Max-SAT ponderado (LARROSA *et al.*, 2008), entre eles: Max-Cut, Max-Clique, roteamento (XU *et al.*, 2003), bioinformática (STRICKLAND *et al.*, 2005), programação (VASQUEZ; HAO, 2001) e raciocínio probabilístico (PARK, 2002).

O Max-SAT ponderado ocorre quando um peso w_i é associado a cada cláusula C_i , onde C é o conjunto de cláusulas e w o número natural que designa o custo da falsificação da cláusula (BATTITI, 2009; LARROSA *et al.*, 2008). Uma cláusula é *hard* se o peso correspondente a ela for infinito; caso contrário, diz-se que a cláusula é considerada *soft* (LARROSA *et al.*, 2008).

Uma fórmula ponderada é um conjunto múltiplo de cláusulas ponderadas $\phi = \{(C_1, w_1), \dots, (C_m, w_m), (C_{m+1}, \infty), \dots, (C_{m+m'}, \infty)\}$, entre as quais as primeiras m cláusulas são *soft* e as últimas m' cláusulas são *hard*, correspondendo uma cláusula ponderada ao par (w_i, C_i) (ANSOTEGUI *et al.*, 2010).

Seja ϕ um multiconjunto ponderado de cláusulas, o problema Max-SAT parcial ponderado consiste em encontrar uma atribuição ótima para as variáveis de ϕ que minimize o custo da atribuição em ϕ (ANSOTEGUI *et al.*, 2010). De acordo com Zha *et al.* (2020, p.5):

“No Max-SAT parcial ponderado, a instância do problema é normalmente expressa como um conjunto de cláusulas rígidas e flexíveis, em que cada cláusula flexível tem um peso numérico positivo limitado que indica o custo de falsificar a cláusula flexível. O problema é encontrar um modelo que satisfaça todas as cláusulas rígidas e minimize o custo total (ou seja, maximizar a soma dos pesos das cláusulas flexíveis satisfeitas).”

Caso o custo seja infinito, uma cláusula *hard* deve ser falsificada e o conjunto múltiplo de cláusulas é insatisfazível. O Max-SAT parcial ponderado é um problema combinatório que pode ser utilizado em diversas aplicações, sejam problemas de programação, roteamento e

instalação de pacotes de software (ANSOTEGUI *et al.*, 2010). De acordo com Fu e Malik (2006) o Max-SAT parcial ponderado, possui um poder de expressividade maior que o Max-SAT.

Os solucionadores atuais empregam técnicas avançadas de heurísticas de ramificação, algoritmos de aprendizagem e implementações otimizadas, de forma que muitas instâncias de problemas difíceis possam ser resolvidas em quantidades razoáveis de tempo (SOHANGH-PURWALA *et al.*, 2017). Os avanços dos SAT solucionadores são apontados por Roig (2013) relacionados a aplicação bem-sucedida da satisfatibilidade proposicional em problemas industriais. Entre esses principais avanços estão backtracking não cronológico, aprendizado de cláusula orientado por conflito e propagação de restrições booleanas eficiente (FU; MALIK, 2006).

4 TRABALHOS RELACIONADOS

Nesse capítulo, será analisado os principais trabalhos associados a esta pesquisa. Na seção 4.1, é apresentada uma prova matemática reduzindo o problema da satisfazibilidade ao problema do caixeiro-viajante; na seção 4.2 uma abordagem híbrida utilizando o Max-SAT como técnica de pré-processamento da solução é aplicada ao problema do caixeiro-viajante é detalhada. Enquanto em 4.3 um método composto baseado na heurística de Clarke e Wright (1964) é explorado. Na seção 4.4 uma visão geral dos trabalhos relacionados e proposto é exibida.

4.1 *Redução do SAT ao Problema do Caixeiro-Viajante*

No seu trabalho Bronts (2014) buscou analisar o processo de redução do problema da satisfazibilidade booleana para uma versão decidível do problema do caixeiro-viajante. Investigando inicialmente uma redução direta entre os problemas, visto que se dois problemas forem pertencentes a classe NP-Completo, existirá uma redução direta entre eles (SIPSER, 2005).

Em Bronts (2014) a versão decidível do TSP consiste em descobrir se existe um percurso em que cada cidade é visitada exatamente uma vez e sem exceder o custo. E apesar de não existir uma relação direta entre o SAT e TSP, existem reduções que podem ser realizadas entre os problemas (BRONTS, 2014).

Karp (1972) forneceu transformações de instâncias para diferentes problemas NP-completos (KARP, 1972; BRONTS, 2014), que foram usadas como base por Bronts (2014) para o fornecimento de provas complementares as existentes em Karp (1972) para averiguar a possibilidade de uma redução direta entre os problemas. Contudo, a redução direta mostrou-se difícil de realizar e o autor decidiu reduzir do SAT para o CLIQUE (KARP, 1972), do CLIQUE para a cobertura de vértices (KARP, 1972), da cobertura de vértices para o circuito hamiltoniano e por fim do circuito hamiltoniano para o TSP (KARP, 1972).

Bronts (2014) relata que após finalizar as reduções pode compreender as razões pelas quais não conseguiu estabelecer uma redução direta do SAT para TSP, pois o grafo resultante torna-se complicado para apenas uma fórmula booleana simples e qual fórmula original do grafo. Além disso, do CLIQUE para a cobertura de vértices se tem o grafo complementar e de cobertura de vértices para o circuito hamiltoniano as bordas passam a ser consideradas e não os nós, de modo que a redução direta seja um processo complicado e que exija mais tempo do que o autor disponha para elaboração da sua tese (BRONTS, 2014).

4.2 *Redução do Problema do Roteamento de Veículos ao Max-SAT*

Khadilkar (2022) propôs o uso das técnicas de aprendizado por reforço, implementação de políticas e o uso do Max-SAT para o problema do roteamento de veículos com janelas de tempo (*Capacitated Vehicle Routing Problem with Time Windows (CVRP-TW)*) como uma abordagem híbrida. O autor afirma que a metodologia desenvolvida pode ser aplicável a outros problemas de otimização combinatória, em consequência do seu carácter generalizável.

Os métodos híbridos costumam se adotados, pois permitem a combinação das vantagens dos diferentes tipos de algoritmos ao usarem heurísticas de construção e melhoria ao mesmo tempo, de modo que cada parte pode ser implementada usando diferentes algoritmos e estruturas de dados. No caso de Khadilkar (2022) a combinação dos benefícios das formulações exatas, aprendizado e metaheurísticas foram buscadas em seu trabalho.

Primeiro um pré-processamento é realizado no conjunto de dados para caracterização dos dados, em seguida um algoritmo de agrupamento é aplicado para separar todos os clientes em clusters, agrupamento de Khadilkar (2022) funciona sem necessitar predeterminar o número de clusters. Características como tempo médio de viagem entre todos os pares de clientes, a janela de tempo máxima e a maior distância entre clientes no conjunto de dados são utilizadas para calcular os recursos que são usados como entradas para a política de aprendizado por reforço.

O aprendizado por reforço (*Reinforcement Learning (RL)*) é um paradigma do aprendizado de máquina, no qual o novo conhecimento é adquirido de forma dinâmica mediante um processo de tentativa e erro (VIZEL *et al.*, 2015). Diversos trabalhos propuseram o uso do RL para o Vehicle Routing Problem (VRP) (ver Vizel *et al.* (2015)) e foram usados de inspiração pelo autor para elaboração do procedimento híbrido.

A abordagem desenvolvida por Khadilkar (2022) utiliza a implementação política para determinar os κ pares mais viáveis de veículos e clientes são selecionados pela política de aprendizado por reforço durante a fase de *rollout*. O RL determina em cada etapa um par veículo-cliente viável conseguem atender um cliente respeitando os limites de capacidade e a janela de tempo, por meio de redes neurais que estabelecem um valor para cada par viável (KHADILKAR, 2022).

O Max-SAT é aplicado em dois momentos distintos na abordagem proposta, primeiramente é usado enquanto as rotas estão sendo calculadas, a fim de otimizar os resultados da distribuição, chamo pelo autor de SAT avançado, e depois para melhoria da solução final, denominado de SAT de restrição. No SAT avançado existe o sub-caminho R gerado pela imple-

mentação das políticas e o conjunto A clientes adicionais que estão ativos e que são vizinhos do cluster dos membros de R . O objetivo é incluir na rota do sub-caminho R alguns clientes de A , desde que as restrições de capacidade e tempo do veículo sejam respeitadas (KHADILKAR, 2022).

Enquanto o SAT de restrição é aplicado para explorar oportunidades de otimização potencialmente maiores realizando pequenas alterações. A principal diferença entre SAT avançado e o SAT de restrição, é que no SAT de restrição apenas a rota R é considerada, pois o objetivo é encontrar uma ordem ideal dos clientes (KHADILKAR, 2022). A formulação Max-SAT é ativada nos 2.000 épocas (*epoch*) finais, enquanto os primeiros 4.000 episódios são realizados apenas com o aprendizado por reforço e implementações de políticas.

O conjunto de dados de Sintef (2008) foi utilizado por Khadilkar (2022) para o treinamento do RL com 56 instâncias de 25 clientes. Enquanto o modelo elaborado foi submetido a testes usando o mesmo conjunto de treinamento e outras duas bases, uma de 50 e outra 100 clientes.

Uma comparação da distância média e do tempo é realizada entre a abordagem híbrida proposta, o aprendizado por reforço proposto em Sultana *et al.* (2021), o Q -network proposto por Gupta *et al.* (2022) e desempenho de um algoritmo genético computado pelos mesmo pesquisadores do Q -network. Khadilkar (2022) também verifica o impacto do uso do Max-SAT registrando o valor obtido apenas pelo RL e implementação de políticas, e depois com adição do solver.

O comparativo dos algoritmos evidencia uma qualidade de solução do algoritmo genético mais aproximada ao melhor valor da distância conhecido, contudo o método de Khadilkar (2022) apresenta um tempo computacional reduzido em relação ao algoritmo genético ao mesmo tempo que obtém uma qualidade de solução superior aos demais algoritmos (em apenas quatro situações o Q -network conseguiu uma qualidade superior).

Khadilkar (2022) destaca que conforme o número de *rollouts* aumentam, o tempo de computação sofre um aumento, ao mesmo tempo que a qualidade da solução melhora. Havendo, desta forma, a flexibilidade da abordagem para se adaptar a diferentes estratégias de solução conforme o valor de κ (KHADILKAR, 2022).

4.3 *Procedimento de melhoria com o algoritmo K-Opt*

Komarudin e Parhusip (2019) adotaram a combinação da heurística de Clarke e Wright (1964) e a busca local K-opt, buscando melhorar a aproximação das soluções geradas enquanto mantém o tempo de computação relativamente curto. A utilização do método composto é justificada pela combinação das vantagens dos procedimentos construtivos e de melhoria para resolver o TSP (CLARKE; WRIGHT, 1964).

O algoritmo de Clarke e Wright (1964) é usado para construção da solução inicial, operando conectando os nós com base na maior economia. As economias são calculadas a partir da soma das distâncias entre dois nós e da soma das distâncias entre o depósito e cada nó. A economia de cada nó será calculada a partir das coordenadas x e y para criação de uma tabela contém a lista de economias dos 100 nós mais próximos entre si (KOMARUDIN; PARHUSIP, 2019; CLARKE; WRIGHT, 1964).

A tabela é utilizado para elaborar a solução inicial, a construção do percurso será realizada conectando os nós com as maiores economias até que todos estejam conectados. Para assegurar que a solução seja válida para o contexto do problema do caixeiro-viajante, cada etapa da heurística elimina rotas que estejam ligadas na etapa anterior (KOMARUDIN; PARHUSIP, 2019).

Komarudin e Parhusip (2019) utilizaram o método K-Opt para melhorar a solução inicial gerada pela de economia de Clarke e Wright (1964). O valor de K determina o número de bordas, que são removidas e reconectadas em cada iteração. No procedimento composto proposto pelos autores, o valor de K irá variar entre dois, três, quatro e cinco operadores opt na mesma iteração, para explorar diferentes vizinhanças do espaço de solução e encontrar soluções melhores (KOMARUDIN; PARHUSIP, 2019).

Um limite de tempo de no máximo 24 horas de iteração e o valor máximo de 999 de iterações é estabelecido para atuar no procedimento de melhoria para que algoritmo termine em um tempo razoável, mas que ainda assim uma boa solução seja gerada. Deste modo, o procedimento é encerrado caso nenhuma melhoria adicional seja atingida ou caso o limite de tempo ou a iteração máxima ocorram (KOMARUDIN; PARHUSIP, 2019).

O conjunto de dados VLSI fornecidos por Andre Rohe e estudados no Forschungsinstitut für Diskrete Mathematik — Universität Bonn, foram utilizados por Komarudin e Parhusip (2019) para averiguar o comportamento da abordagem proposta, o conjunto de dados usado apresenta suporte ao formato da biblioteca TSPLIB (REINELT, 1991). Os autores selecionaram

30 das 102 instâncias disponíveis do VLSI com uma variação do número de nós de 131 a 85.900 nós nas instâncias escolhidas.

Komarudin e Parhusip (2019) relataram que para instância variando entre 100 e 2.000 nós, o método composto desenvolvido por eles obtém soluções que diferem cerca de 0-3% das melhores soluções já conhecidas. De modo geral, a solução média obtida pelo procedimento de Komarudin e Parhusip (2019) está a uma diferença de cerca 2,55% da melhor solução já encontrada. Mostrando-se um resultado competitivo, dado o grande número de nós e a complexidade de resolver o problema do caixeiro-viajante (KOMARUDIN; PARHUSIP, 2019).

4.4 *Análise dos trabalhos relacionados e o proposto*

O quadro 2 apresenta os resumo dos trabalhos relacionados, permitindo uma comparação entre a pesquisa elaborada neste trabalho e os que foram selecionados considerando o objetivo a ser atingido, o uso de métodos construtivos, de melhoria, compostos ou híbridos e a adoção de solucionadores SAT.

Quadro 2 – Trabalhos Relacionados.

Trabalho	Objetivo	Uso de técnicas para melhorias adicionais na solução	Adoção de solucionadores SAT
Khadilkar (2022)	Minimizar a distância das rotas	Sim	Max-SAT
Komarudin e Parhusip (2019)	Minimizar a distância das rotas	Não se aplica	Não se aplica
Autor	Minimizar a distância das rotas	Sim	Max-SAT Parcial Ponderado

Fonte: Elaborado pelo autor.

Nesta pesquisa, é proposto o uso do Max-SAT Parcial Ponderado para realizar uma pós-melhoria da solução inicial, correspondendo ao uso de um método exato para realizar melhorias na solução, similarmente ao que é proposto por Khadilkar (2022). A principal diferença entre este trabalho e o desenvolvido por Khadilkar (2022) consiste no uso de diferentes solucionadores SAT. Outra distinção entre os trabalhos está relacionada que abordagem proposta neste trabalho, assim como em Komarudin e Parhusip (2019) trata-se de um método composto, Khadilkar (2022) refere-se a um método híbrido.

Outra similaridade entre este trabalho e do Komarudin e Parhusip (2019) consiste na adoção de um limite de tempo para que a otimização da rota ocorra, contudo, um limite

de tempo maior é usado por Komarudin e Parhusip (2019) em relação ao estabelecido neste trabalho. Apesar de ambos serem procedimentos compostos, Komarudin e Parhusip (2019) usa uma abordagem algorítmica diferente do escolhida nesta pesquisa para construção do caminho. Na realização de melhorias do percurso há uma variação dos operadores opt em Komarudin e Parhusip (2019), enquanto o 2-Opt é fixo neste trabalho.

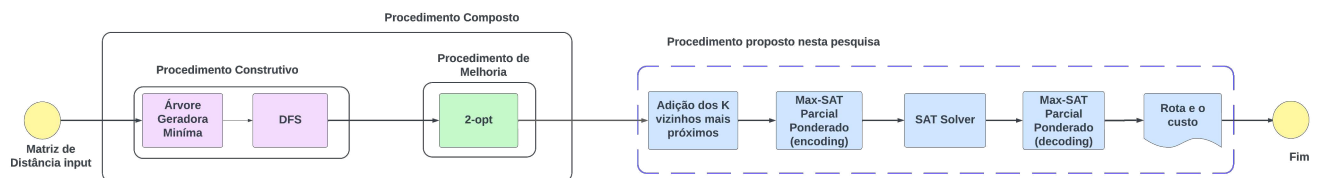
A relevância da pesquisa de Bronts (2014) para o escopo deste estudo, trata-se do estabelecimento de uma relação entre os problemas da satisfazibilidade booleana e o problema do caixeiro-viajante, embora nesta pesquisa seja realizada a redução do TSP para o SAT.

5 METODOLOGIA

Neste capítulo, serão descritos os procedimentos metodológicos realizados para elaboração da abordagem proposta. Na seção 5.1 será informado o conjunto de dados utilizados, em 5.2 é descrito a heurística utilizada para construção da solução inicial, enquanto na 5.3 o método de melhoria. A abordagem é detalhada na seção 5.5.

Todos os algoritmos implementados nesta pesquisa foram desenvolvidos utilizando a linguagem de programação Python (versão 3.11.5). A figura 5 demonstra o fluxo do funcionamento do procedimento resultante da aplicação da modelagem como procedimento de pós-melhoria sobre a solução obtida pelo algoritmo construtivo da árvore geradora mínima e a melhoria realizada pelo 2-Opt.

Figura 5 – Procedimento de Pós-melhoria.



Fonte: Elaborado pelo autor.

5.1 Conjunto de Dados

Para o desenvolvimento deste trabalho, foram identificados benchmarks das instâncias da biblioteca TSPLIB (REINELT, 1991) e a instância p01 disponibilizada por Burkardt (2019). A TSPLIB trata-se de uma famosa biblioteca na literatura que contém exemplos de instâncias para o problema do caixeiro-viajante e outros problemas associados ao TSP (REINELT, 1991). As instâncias selecionadas para averiguar a melhoria de custo e tempo computacional da abordagem proposta foram: p01, gr21, fri26, att48, berlin52 e pr76, contendo respectivamente conjunto de 15, 21, 26, 48, 52 e 76 nós. A instância att48 corresponde as 48 capitais dos Estados Unidos e berlin52 a 52 pontos em Berlim na Alemanha.

As instâncias p01, gr17, fri26, e att48 já estavam com a matriz de distância euclidiana calculadas, prontas para o uso. Contudo, para as instâncias gr21, berlin52 e pr76 foi necessário calcular a distância euclidiana entre cada cidade e construir uma matriz de distância.

Para realizar o cálculo da matriz de distância, a biblioteca NumPy foi empregada para criação de uma matriz $n \times n$ com todos os elementos sendo 0, enquanto o método *get_weight()*

da TSPLIB é usada para preencher a matriz com a distância entre cada nó. O cálculo da matriz de distância é importante, pois como visto na figura 5 a entrada para o procedimento proposto consiste em uma matriz de distância.

5.2 Construção da solução inicial

Para construção da solução inicial, uma versão simplificada do algoritmo de Christofides (1976) foi desenvolvida utilizando recursos da biblioteca Python NetworkX (HAGBERG *et al.*, 2008). Nesta versão, a correspondência de peso mínimo elaborada a partir da árvore de geradora mínima não é adotada, porém, o ciclo de Euler é construído usando uma busca de profundidade (*Depth-First Search (DFS)*).

Dada a matriz de distância fornecida como parâmetro, a NetworkX é utilizada para transformar a matriz de distância em um objeto do tipo grafo, para que então o método *minimum_spanning_tree* seja adotado para implementação da MST, considerando as menores arestas.

“O DFS é um tipo de algoritmo de busca que usa pilhas para localizar todos os nós. O método do DFS para localizar nós não visitados simplifica a descoberta das melhores rotas” (UDDIN; UDDIN, 2020, p.417). Por este motivo foi adotado neste trabalho para criar o caminho Euleriano para evitar os nós visitados anteriormente, desse modo resultado na finalização do procedimento construtivo e retornando uma rota inicial.

5.3 Procedimento de Melhoria com 2-Opt

Para realizar a melhoria da solução inicial, optou-se pelo 2-Opt, pois este algoritmo mantém as trocas ramais com uma complexidade menor em relação aos demais procedimentos de melhorias (PRODHON; PRINS, 2016). Por tanto, após a criação da rota inicial, aplica-se a técnica 2-Opt para aprimorar o caminho gerado pela heurística adaptada de Christofides (1976). Nessa etapa, o 2-Opt realizará inversões de duas arestas no percurso até que o procedimento não seja capaz de identificar novas possíveis melhorias.

5.4 Aumento do Espaço de Solução

Em seguida, os K vizinhos mais próximos são adicionados a cada vértice do caminho resultante do aprimoramento pelo algoritmo 2-Opt, considerando a proximidade dos vértices na

árvore de geradora mínima. Caso um nó não possua os K vizinhos necessários na configuração da árvore, a matriz de distância será considerada para a escolha dos vizinhos.

O valor de K testado foram para um, dois e os três vizinhos mais próximos, com a finalidade de averiguar o impacto de diferentes escolhas de vizinhos e da ampliação do espaço de busca. Realizada a adição do K vizinho mais próximo, a matriz de distância é simplificada ao considerar apenas aos custos das arestas que compõe o caminho inicial, bem como as arestas que ligam cada nó do caminho inicial aos seus K vizinhos mais próximos. Por convenção, as arestas removidas foram declaradas com custo -1 para representar o custo infinito. A matriz simplificada é empregada como parâmetro de entrada na fase seguinte em que a modelagem é utilizada.

5.5 Modelagem Lógica

Para este trabalho, optou-se pela utilização do solucionador NaPS (SAKAI; NABESHIMA, 2015) devido à sua implementação eficiente e otimizada baseado de modo subjacente no solucionador MiniSat, ao suporte para solucionar instâncias do problema Max-SAT Parcial Ponderado e por permitir limitar o tempo de otimização. O NaPS é usado nesta pesquisa para resolver o modelo passado por parâmetro como uma instância do Max-SAT parcial ponderado.

Baseado nas dimensões da matriz de distância simplificada, o número de cidades é definido e o mapeamento de cada aresta W que conecta uma cidade i a uma cidade j como cláusulas no formato w_{ij} é realizado, onde $i, j \in 1, 2, 3, \dots, n$, de modo que n é o número de cidade e $i \neq j$. Cada cláusula pode assumir um valor verdadeiro ($w_{ij} = 1$) ou falso ($w_{ij} = 0$). As cláusulas são mapeadas como números inteiros para construção do modelo que será utilizado como entrada para o solucionador SAT.

Logo após ao mapeamento das cláusulas, as restrições do problema são especificadas para que o solucionador SAT possa ser executado. As restrições 5.2 e 5.3 garantem que apenas uma aresta entre e saia do vértice zero, ao passo que as restrições 5.4 e 5.5 asseguram que apenas uma aresta entre e saia de cada vértice. A restrição 5.6 estabelece que nenhum vértice seja visitado mais de uma vez, enquanto 5.7 e 5.8 fornecem a base e a indução do caminho para evitar a formação de sub-rotas na solução. A função 5.1 trata-se da função de minimização da rota.

$$\sum_{i=0}^n \sum_{j=0}^n w_{ij} \times d_{ij} \tag{5.1}$$

$$\sum_{j=0}^n w_{ij} = 1, \quad i = 0 \quad (5.2)$$

$$\sum_{i=0}^n w_{ij} = 1, \quad j = 0 \quad (5.3)$$

$$\sum_{j=0}^n w_{ij} = 1, \quad \forall i \in \{1, 2, 3, \dots, n\}, \quad i \neq j \quad (5.4)$$

$$\sum_{i=0}^n w_{ij} = 1, \quad \forall j \in \{1, 2, 3, \dots, n\}, \quad i \neq j \quad (5.5)$$

$$\sum_{i=0}^n \sum_{j=i+1}^n -w_{ij} - w_{ji} \geq 1 \quad (5.6)$$

$$\sum_{i=1}^n \sum_{j=1}^n -w_{ij} + c_{ji} \geq 1, \quad i \neq j \quad (5.7)$$

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n -w_{ij} - c_{jk} + c_{ik} \geq 1, \quad i \neq j \quad (5.8)$$

O aprendizado de cláusula orientada por conflito (*Conflict-Oriented Clause Learning (CDCL)*) é usada pelo solucionador para melhorar o desempenho da otimização da solução, ao realizar a propagação da cláusula unitária ou selecionando cláusulas não assinadas quando não houver mais cláusulas unitárias. No CDCL, novas cláusulas são adicionadas baseado no surgimento de conflitos nas atribuições parciais na resolução do problema, a verificação da causa do conflito é realizada entre as cláusulas para adicionar uma nova restrição ao problema que evita a repetição insatisfatória da sub-árvore (GOCHT, 2017; SOHANGHPURWALA *et al.*, 2017).

Um tempo limite de 20 minutos foi estabelecido para que o solucionador SAT realize a otimização. Conseqüentemente, os seguintes cenários poderão ocorrer: o ótimo para o modelo passado por parâmetro é encontrado antes que o limite de tempo seja atingido, encerrando a execução do algoritmo e o caminho é exibido; ou em 20 minutos, o ótimo para o modelo não é

encontrado e, assim que o tempo é alcançado, o algoritmo retorna o custo obtidos até o momento da finalização do algoritmo e o caminho.

Ao encerrar a execução do SAT solver, ocorre o mapeamento reverso de inteiros para cláusulas, para que seja possível exibir o caminho obtido e o custo associado à solução. Neste trabalho, o solucionador SAT é utilizado para realizar um pós-melhoria da solução, isto é, realizar melhorias adicionais na solução final.

6 RESULTADOS

Neste capítulo, serão apresentados os resultados alcançados pela aplicação do Max-SAT parcial ponderado para pós-melhoria da solução. Com a finalidade de compreender os impactos do uso do solucionador SAT na solução final, verificou-se o custo da rota antes e após a aplicação da modelagem desenvolvida. A métrica de custo neste trabalho refere-se a distância.

Foram verificados a aproximação da distância obtida pela solução antes e depois do uso da modelagem com o custo ótimo de cada instância, o tempo médio de dez execuções das instâncias e a variação do número de vizinhos (K) para um, dois e três vizinhos mais próximos. Como descrito na seção 5.1 os testes foram realizados em cinco *benchmarkings* da biblioteca TSPLIB e um de Burkardt (2019), estando disponível para consulta os valores do custo ótimo das instâncias. As especificações técnicas dos componentes de hardware utilizado para execução dos testes são descritos na tabela 1.

Tabela 1 – Especificações de Hardware

CPU	Memória RAM	Espaço de Armazenamento
Intel Core i5-10210U @ 1.60 GHz	8GB	128GB SSD

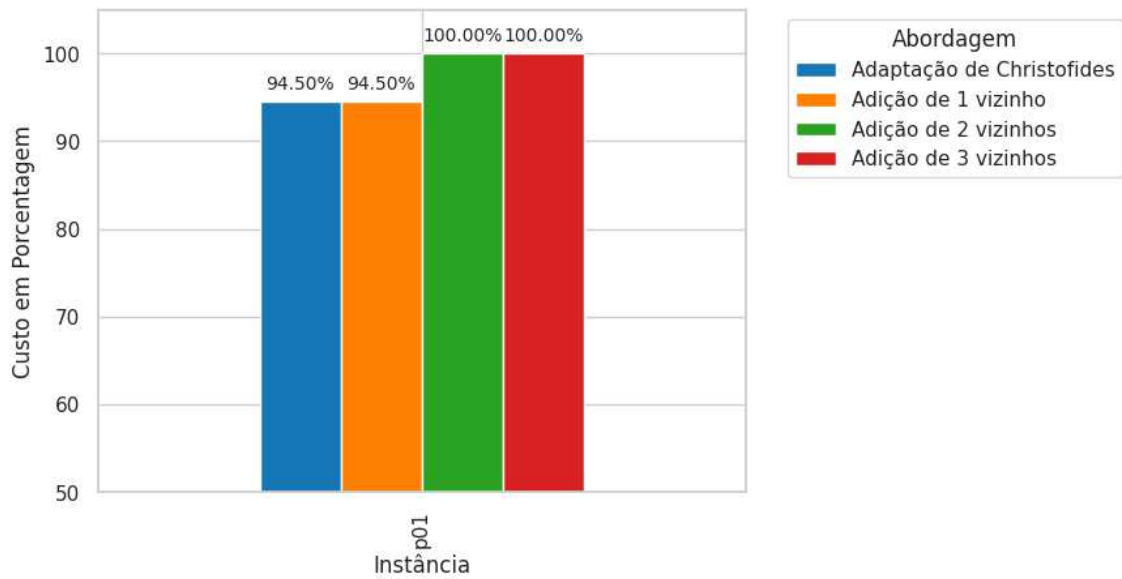
Fonte: Elaborado pelo autor.

6.1 Análise da Abordagem

Conforme descrito na metodologia (ver 5.4), uma análise do impacto da escolha de um, dois e os três vizinhos mais próximos na qualidade da solução e no custo computacional em função do tempo foi realizada. Em todas as instâncias utilizadas para teste, a escolha de um vizinho mais próximo a cada nó conjuntamente com a otimização do solucionador SAT obtiveram o mesmo custo que a versão modificada de heurística de Christofides (1976) e a melhoria do 2-Opt.

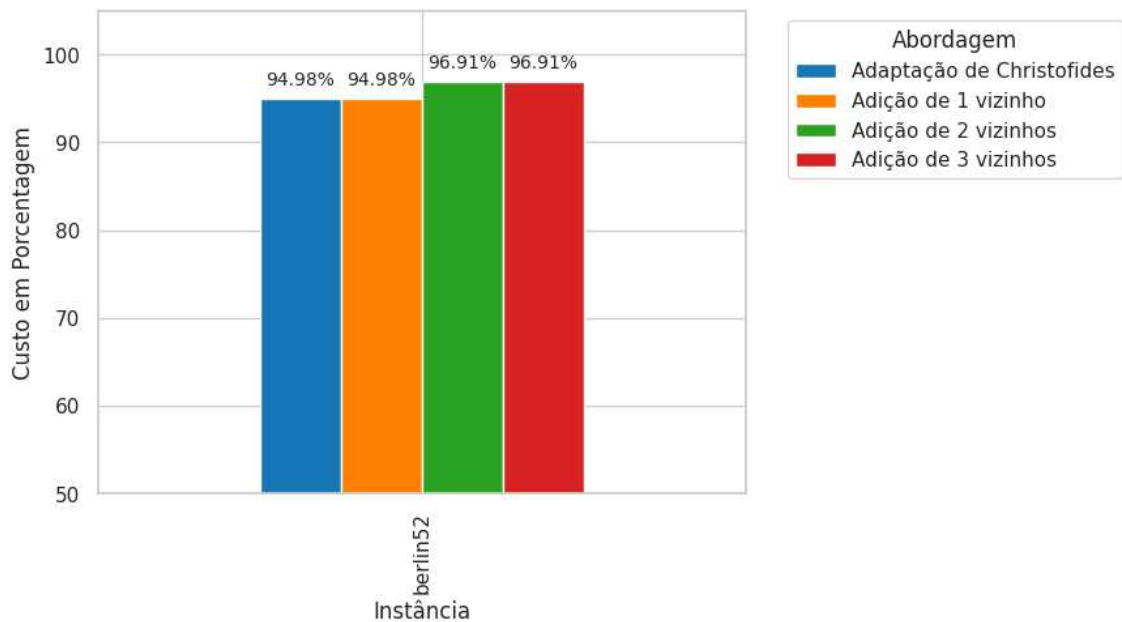
Para as instâncias p01 e berlin52, a partir da adição dos dois vizinhos e permanecendo a mesma para os três vizinhos mais próximos, pode-se observar a melhoria do custo em relação à adaptação de Christofides (1976) e ao 2-Opt de 5,5% e 1,93%, respectivamente. Nas figuras 6 e 7 é possível observar o desempenho da abordagem proposta nesta pesquisa variando os números de vizinhos e comparada com a versão modificada de Christofides (1976) e a melhoria do 2-Opt em termos de porcentagem para as instâncias p01 e berlin52, correspondendo 100% em ambas as visualizações o valor ótimo de cada instância.

Figura 6 – Comparação do Custo para p01.



Fonte: Elaborado pelo autor.

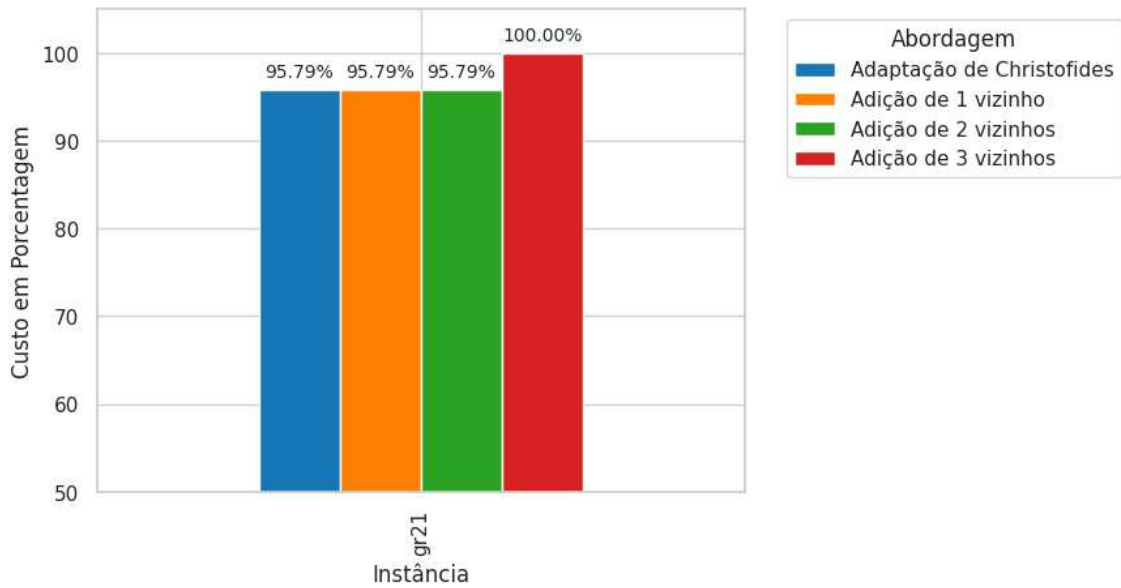
Figura 7 – Comparação do Custo para berlin52



Fonte: Elaborado pelo autor.

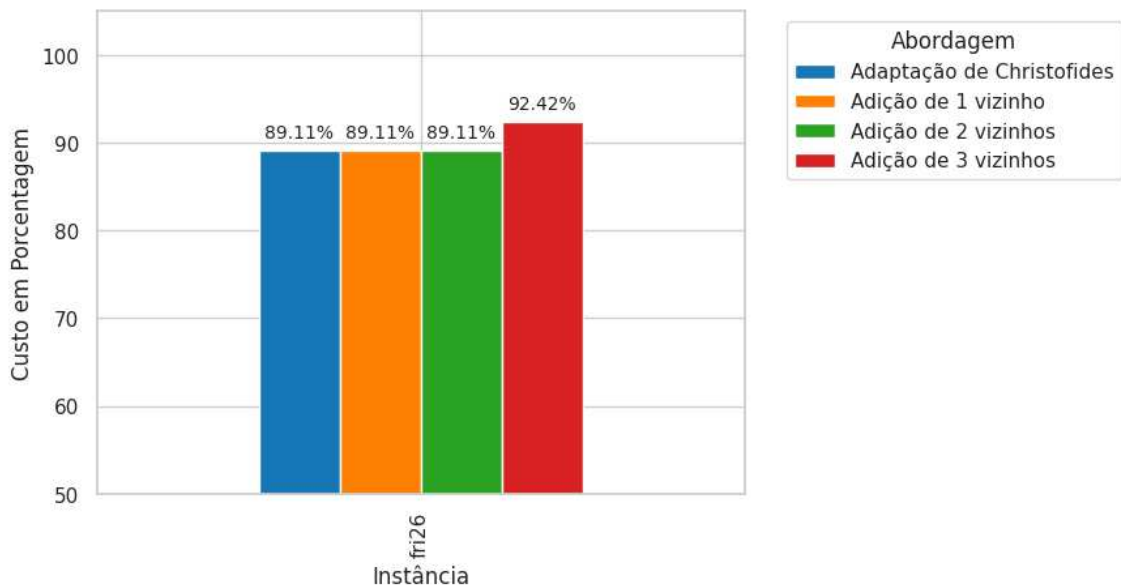
Para as instâncias gr21, fri26 e att48 a melhoria obtida pelo uso da modelagem só ocorre para os três vizinhos mais próximos (ver 8, 9 e 10), ocasionada pelo aumento do espaço de solução que pode ser explorado pelo solucionador SAT. A melhoria em relação à solução obtida pela MST e melhorada pelo 2-Opt consiste em um refinamento da solução pela modelagem em cerca de 4,21%, 3,31 % e 4,29%, respectivamente para as instâncias gr21, fri26 e att48.

Figura 8 – Comparação do Custo para gr21.



Fonte: Elaborado pelo autor.

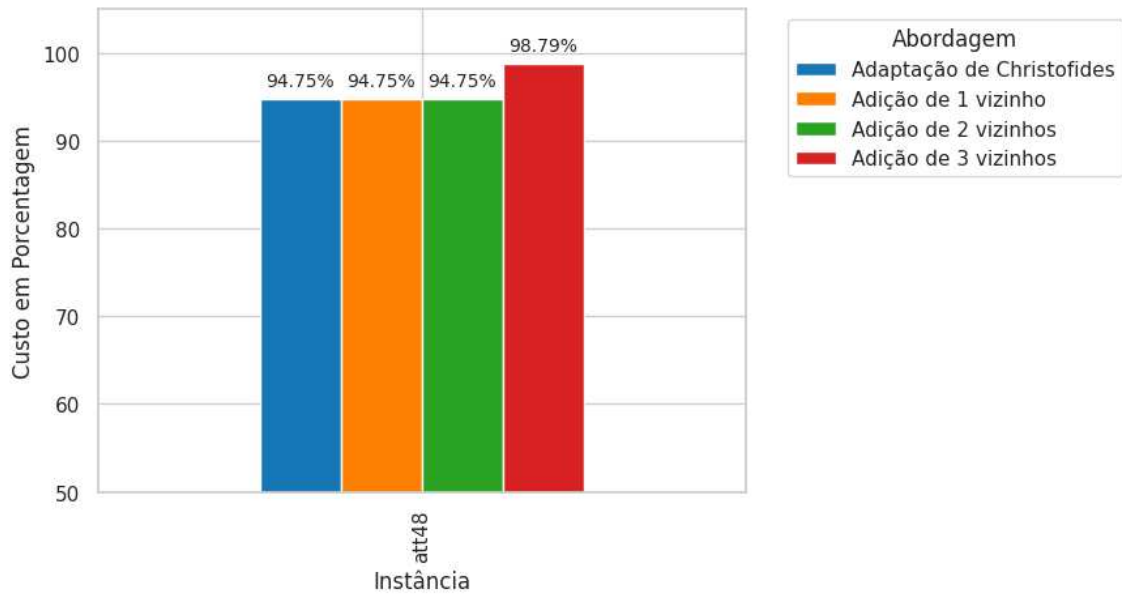
Figura 9 – Comparação do Custo para fri26.



Fonte: Elaborado pelo autor.

Entretanto, para a instância pr76 observa-se que para os dois vizinhos mais próximos o custo se mantém o mesmo que para MST com o 2-Opt e para modelagem com a adição de um vizinho mais próximo. Enquanto na adição dos três vizinhos mais próximos ocorre um aumento do custo da solução obtida pelo procedimento proposto nesta pesquisa como pode ser observado na figura 11. Este comportamento é explicado pela limitação de tempo para realização da otimização, pois com a escolha dos três vizinhos mais próximos e a aumento do espaço de solução, para 76 nós, 20 minutos de otimização não foi tempo suficiente para que o solucionador realizasse a melhoria na solução.

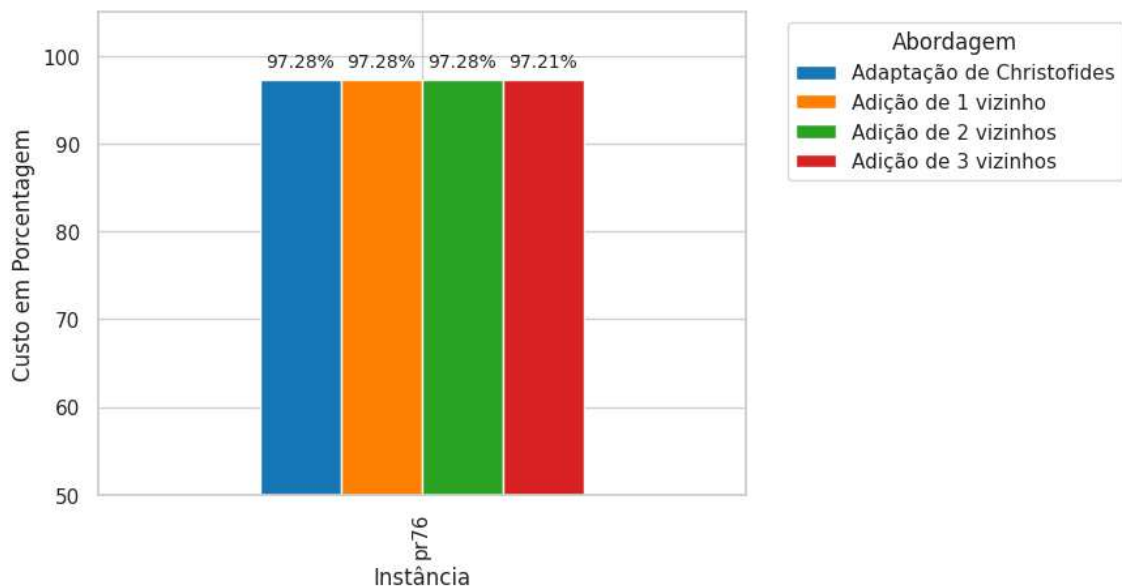
Figura 10 – Comparação do Custo para att48.



Fonte: Elaborado pelo autor.

Consequentemente, apesar das instâncias p01 e berlin52 obtenham excelentes resultados com a adição de dois vizinhos mais próximos, os três vizinhos mais próximos garantem a maior aproximação do custo ótimo com uma variação mínima do tempo requerido em relação aos dois vizinhos mais próximos em uma porcentagem significativa das instâncias, conforme sugerem os resultados.

Figura 11 – Comparação do Custo para pr76.



Fonte: Elaborado pelo autor.

Adotando o aumento do espaço de solução para os três vizinhos mais próximos de cada nó da rota inicial construída e melhorada pelo procedimento 2-Opt, a pós-melhoria da

solução realizada pela abordagem proposta aplicando o Max-SAT parcial ponderado referente ao percurso de menor distância conhecida e fornecido como valor ótimo pela TSPLIB, a modelagem lógica consegue obter excelentes aproximações com variações de tempo mínima, conforme os resultados apresentados na tabela 2.

Tabela 2 – Resultados

Instâncias			Abordagem			
			Christofides adaptado + 2-opt	Modelagem lógica		
Nome	Tamanho	ótimo	Custo	Custo	K	Tempo Médio (S)
p01	15	291	307	307	1	0.143
				291	2	0.154
				291	3	0.204
gr21	21	2707	2816	2816	1	0.331
				2816	2	0.336
				2707	3	0.690
fri26	26	937	1039	1039	1	0.487
				1039	2	0.568
				1008	3	2.855
att48	48	33523	35281	35281	1	2.752
				35281	2	3.553
				33927	3	14.94
berlin52	52	7542	7920	7920	1	2.999
				7775	2	3.121
				7775	3	20.251
pr76	76	108159	111100	111100	1	1203.341
				111100	2	1202.185
				111173	3	1202.366

Fonte: Elaborado pelo autor.

A tabela 2 evidencia o grau de proximidade entre o custo ótimo da instância e o custo encontrado pela modelagem, que para as instâncias p01 e gr21 o valor ótimo é encontrado, enquanto para fri26, att48 e berlin52 ocorre uma aproximação de cerca de 7,58%, 1,21% e 3,09% do ótimo, respectivamente. Indicando que a abordagem proposta nesta pesquisa demonstra ser mais eficiente do que os métodos construtivos na obtenção de soluções próximas ao ótimo ou ótimas a custos computacionais aceitáveis para instâncias pequenas.

7 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Na literatura, algoritmos de natureza heurística e metaheurística, como 2-Opt e têmpera simulada (MATAI *et al.*, 2010), são empregados para realizarem melhorias adicionais nas rotas finas do problema do caixeiro-viajante com o intuito de aprimorar a solução final. Contudo, nesta pesquisa um método de otimização de exato é utilizado para a otimização para o TSP usando a redução do problema para Max-SAT parcial ponderado para executar um procedimento de pós-melhoria na solução inicial obtida usando o solucionador SAT.

Os resultados apresentados indicam que a modelagem lógica adotada com a aumento do espaço da solução para os três vizinhos mais próximos obtém soluções próximas ao ótimo ou ótimas a custos computacionais aceitáveis para instâncias pequenas. Mostrando-se uma abordagem apropriada para contextos em que aproximações do valor ótimo são aceitas, pois a melhoria alcançada na qualidade da solução ou no custo computacional necessário para obter a rota, por menor que seja, ainda resultará em uma economia significativa de custos para as empresas (HUERTA *et al.*, 2022; HELSGAUN, 2000).

Como os solucionadores SAT estão em constante evolução graças às competições anuais de SAT, que não só permitem o aprimoramento dos solucionadores devido às licenças de código aberto, mas também são um espaço para o surgimento de novas abordagens e solucionadores mais rápidos (SOHANGHPURWALA *et al.*, 2017; MICHEL; HENTENRYCK, 2000).

Os solucionadores SAT estão em constante evolução graças às competições anuais de SAT, que não só permitem o aprimoramento dos solucionadores devido às licenças de código aberto, mas também são um espaço para o surgimento de novas abordagens e solucionadores mais rápidos (SOHANGHPURWALA *et al.*, 2017). Esses solucionadores possuem a facilidade de incorporar novas heurísticas em sua operação, como a heurística de ramificação e a heurística de decisão aprimorada (SOHANGHPURWALA *et al.*, 2017). De modo que em trabalhos futuros pretende-se explorar o impacto que diferentes abordagens heurísticas podem ter na qualidade da solução e no tempo de execução.

Critérios mais sofisticados para a escolha de K vizinhos e o uso de agrupamento, também devem ser exploradas em trabalhos futuros, uma vez que abordagens eficientes para instâncias pequenas são uma boa contribuição para as estratégias de agrupamento. Além da pretensão de explorar variações do TSP em trabalhos futuros.

REFERÊNCIAS

- ALFA, A. S.; HERAGU, S. S.; CHEN, M. A 3-opt based simulated annealing algorithm for vehicle routing problems. **Computers Industrial Engineering**, v. 21, n. 1, p. 635–639, 1991. ISSN 0360-8352.
- ALSALIBI, B.; BABAEIANJELODAR, M.; VENKAT, I. A comparative study between the nearest neighbor and genetic algorithms: A revisit to the traveling salesman problem. **International Journal of Computer Science and Electronics Engineering**, v. 1, p. 34–38, 12 2012.
- ANBUUDAYASANKAR, S. P.; GANESH, K.; MOHAPATRA, S. Survey of methodologies for tsp and vrp. In: _____. **Models for Practical Routing Problems in Logistics: Design and Practices**. Cham: Springer International Publishing, 2014. p. 11–42. ISBN 978-3-319-05035-5.
- ANSOTEGUI, C.; BONET, M. L.; LEVY, J. A new algorithm for weighted partial maxsat. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 24, n. 1, Jul. 2010.
- APPLEGATE, D.; COOK, W.; ROHE, A. Chained lin-kernighan for large traveling salesman problems. **INFORMS Journal on Computing**, v. 15, n. 1, p. 82–92, 2003.
- ARGELICH, J.; MANYÀ, F. Exact max-sat solvers for over-constrained problems. **Journal of Heuristics**, v. 12, n. 4, p. 375–392, Sep 2006. ISSN 1572-9397.
- ARTHANARI, T. S. Symmetric traveling salesman problem and flows in hypergraphs: New algorithmic possibilities. **Atti della Accademia Peloritana dei Pericolanti : Classe di Scienze Fisiche, Matematiche e Naturali**, v. 97, p. 1, 2019.
- BARAGLIA, R.; LAFORENZA, D.; LAGANÀ, A. A web-based metacomputing problem-solving environment for complex applications. In: BUYYA, R.; BAKER, M. (Ed.). **Grid Computing — GRID 2000**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. p. 111–122. ISBN 978-3-540-44444-2.
- BATTITI, R. Maximum satisfiability problem. In: _____. **Encyclopedia of Optimization**. Boston, MA: Springer US, 2009. p. 2035–2041. ISBN 978-0-387-74759-0.
- BATTITI, R.; PROTASI, M. Approximate algorithms and heuristics for max-sat. In: _____. **Handbook of Combinatorial Optimization: Volume1–3**. Boston, MA: Springer US, 1998. p. 77–148. ISBN 978-1-4613-0303-9.
- BRODOWSKY, U. A.; HOUGARDY, S. The approximation ratio of the 2-opt heuristic for the euclidean traveling salesman problem. arXiv, 2020.
- BRONTS, M. M. **Giving a step-by-step reduction from SAT to TSP and giving some remarks on Neil Tennant’s ‘Changes of Mind’**. Faculty of Mathematics and Natural Sciences: Bachelor’s Thesis, 2014.
- BURKARDT, J. **TSP - Data for the Traveling Salesperson Problem**. 2019. <https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html>. (Acessado em 27/03/2023).
- CHIANG, W.-C.; RUSSELL, R. A. A reactive tabu search metaheuristic for the vehicle routing problem with time windows. **INFORMS Journal on Computing**, v. 9, n. 4, p. 417–430, 1997.

CHRISTOFIDES, N. Worst-case analysis of a new heuristic for the travelling salesman problem. **Operations Research Forum**, v. 3, n. 1, p. 20, Mar 1976. ISSN 2662-2556.

CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, v. 12, n. 4, p. 568–581, 1964.

FEILLET, D.; DEJAX, P.; GENDREAU, M. Traveling salesman problems with profits. **Transportation Science**, INFORMS, Linthicum, MD, USA, v. 39, n. 2, p. 188–205, may 2005. ISSN 1526-5447.

FU, Z.; MALIK, S. On solving the partial max-sat problem. In: BIERE, A.; GOMES, C. P. (Ed.). **Theory and Applications of Satisfiability Testing - SAT 2006**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 252–265. ISBN 978-3-540-37207-3.

Minimum spanning tree problem. In: GASS, S. I.; FU, M. C. (Ed.). **Encyclopedia of Operations Research and Management Science**. Boston, MA: Springer US, 2013. p. 981–982. ISBN 978-1-4419-1153-7.

GOCHT, S. **Incremental SAT Solving for SAT Based Planning**. 1–43 p. Tese (Doutorado) – Karlsruher Institut für Technologie, 2017.

GRAHAM, R.; HELL, P. On the history of the minimum spanning tree problem. **Annals of the History of Computing**, v. 7, n. 1, p. 43–57, 1985.

GUPTA, A.; GHOSH, S.; DHARA, A. Deep reinforcement learning algorithm for fast solutions to vehicle routing problem with time-windows. In: **5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)**. New York, NY, USA: Association for Computing Machinery, 2022. (CODS-COMAD 2022), p. 236–240. ISBN 9781450385824.

HAGBERG, A.; SWART, P.; CHULT, D. S. Exploring network structure, dynamics, and function using networkx. In: . United States: [S. n.], 2008. Research Org.: Los Alamos National Lab. (LANL), Los Alamos, NM (United States).

HAR-PELED, S. Approximating the euclidean traveling salesman problem (tsp). In: _____. American Mathematical Society: Geometric Approximation Algorithms, 2011. p. 177–190. ISBN 9780821849118.

HELGAUN, K. An effective implementation of the lin–kernighan traveling salesman heuristic. **European Journal of Operational Research**, v. 126, n. 1, p. 106–130, 2000. ISSN 0377-2217.

HOSSEN, M. S.; POLASH, M. M. A. An efficient local search sat solver with effective preprocessing for structured instances. **SN Computer Science**, v. 2, n. 2, p. 105, Feb 2021. ISSN 2661-8907.

HUERTA, I. I.; NEIRA, D. A.; ORTEGA, D. A.; VARAS, V.; GODOY, J.; ASÍN-ACHÁ, R. Improving the state-of-the-art in the traveling salesman problem: An anytime automatic algorithm selection. **Expert Systems with Applications**, v. 187, p. 115948, 2022. ISSN 0957-4174.

ILAVARASI, K.; JOSEPH, K. S. Variants of travelling salesman problem: A survey. In: . Piscataway, New Jersey: International Conference on Information Communication and Embedded Systems (ICICES2014), 2014. p. 1–7.

KARP, R. M. Reducibility among combinatorial problems. In: _____. **Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department.** Boston, MA: Springer US, 1972. p. 85–103. ISBN 978-1-4684-2001-2.

KHADILKAR, H. **Solving the capacitated vehicle routing problem with timing windows using rollouts and MAX-SAT.** 2022.

KIZILATEŞ, G.; NURIYEVA, F. On the nearest neighbor algorithms for the traveling salesman problem. In: NAGAMALAI, D.; KUMAR, A.; ANNAMALAI, A. (Ed.). **Advances in Computational Science, Engineering and Information Technology.** Heidelberg: Springer International Publishing, 2013. p. 111–118. ISBN 978-3-319-00951-3.

KOMARUDIN; PARHUSIP, S. F. Composite algorithm based on clarke - wright and local search for the traveling salesman problem. In: **Proceedings of the 5th International Conference on Industrial and Business Engineering.** New York, NY, USA: Association for Computing Machinery, 2019. (ICIBE '19), p. 87–90. ISBN 9781450376532.

KONSTANTAKOPOULOS, G. D.; GAYIALIS, S. P.; KECHAGIAS, E. P. Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. **Operational Research**, v. 22, n. 3, p. 2033–2062, jul. 2022.

LARROSA, J.; HERAS, F.; de Givry, S. A logical approach to efficient max-sat solving. **Artificial Intelligence**, v. 172, n. 2, p. 204–233, 2008. ISSN 0004-3702.

LI, M.; CHOW, J. School bus routing problem with a mixed ride, mixed load, and heterogeneous fleet. **Transportation Research Record Journal of the Transportation Research Board**, v. 2675, p. 467–479, Apr 2021.

LIN, S. Computer solutions of the traveling salesman problem. **Bell System Technical Journal**, v. 44, n. 10, p. 2245–2269, 1965.

LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. **Operations Research, INFORMS**, v. 21, n. 2, p. 498–516, Apr 1973. ISSN 0030-364X.

MATAI, R.; SINGH, S.; MITTAL, M. L. Traveling salesman problem: an overview of applications, formulations, and solution approaches. In: DAVENDRA, D. (Ed.). **Traveling Salesman Problem.** Rijeka: IntechOpen, 2010. cap. 1.

MICHEL, L.; HENTENRYCK, P. V. Localizer. **Constraints**, Kluwer Academic Publishers, USA, v. 5, n. 1/2, p. 43–84, jan 2000. ISSN 1383-7133.

OR, I. **TRAVELING SALESMAN TYPE COMBINATORIAL PROBLEMS AND THEIR RELATION TO THE LOGISTICS OF REGIONAL BLOOD BANKING.** Ann Arbor, Michigan: Northwestern University, 1976.

OSABA, E.; YANG, X.-S.; Del Ser, J. Chapter 9 - traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics. In: YANG, X.-S. (Ed.). **Nature-Inspired Computation and Swarm Intelligence.** Academic Press: Academic Press, 2020. p. 135–164. ISBN 978-0-12-819714-1.

OUAARAB, A. Review of tour generation for solving traveling salesman problems. In: _____. **Nature-Inspired Computation in Navigation and Routing Problems: Algorithms, Methods and Applications**. Singapore: Springer Singapore, 2020. p. 85–102. ISBN 978-981-15-1842-3.

PARK, J. D. Using weighted max-sat engines to solve mpe. In: **Eighteenth National Conference on Artificial Intelligence**. USA: American Association for Artificial Intelligence, 2002. p. 682–687. ISBN 0262511290.

PETTIE, S. Minimum spanning trees. In: _____. **Encyclopedia of Algorithms**. Boston, MA: Springer US, 2008. p. 541–544. ISBN 978-0-387-30162-4.

PRODHON, C.; PRINS, C. Metaheuristics for vehicle routing problems. In: _____. **Metaheuristics**. Cham: Springer International Publishing, 2016. p. 407–437. ISBN 978-3-319-45403-0.

REINELT, G. Tsplib— a traveling salesman problem library. **ORSA Journal on Computing**, ORSA, v. 3, n. 4, p. 376–384, Nov 1991. ISSN 0899-1499.

RINTANEN, J. Planning as satisfiability: Heuristics. **Artificial Intelligence**, v. 193, p. 45–86, 2012. ISSN 0004-3702.

ROCHA, A. C.; LIMA, J. P.; ARRUDA, A.; JR., D. Using a max-sat solver as a post-improvement move for traveling salesman problem. In: **The 50th International Conference on Computers and Industrial Engineering**. New York, NY, USA: Curran Associates, 2023. (CIE 50).

ROHE, A. **VLSI Data Sets**. 2013. <https://www.math.uwaterloo.ca/tsp/vlsi/>. (Acessado em 11/10/2023).

ROIG, I. A. Solving hard industrial combinatorial problems with sat. **TDX (Tesis Doctorals en Xarxa)**, Universitat Politècnica de Catalunya, 5 2013.

ROSENKRANTZ, D.; STEARNS, R.; II, P. An analysis of several heuristics for the traveling salesman problem. **SIAM Journal on Computing**, v. 6, p. 563–581, 09 1977.

SAKAI, M.; NABESHIMA, H. Construction of an robdd for a pb-constraint in band form and related techniques for pb-solvers. **IEICE Transactions on Information and Systems**, E98.D, n. 6, p. 1121–1127, 2015.

SATHYA, N.; MUTHUKUMARAVEL, A. A review of the optimization algorithms on traveling salesman problem. **Indian journal of science and technology**, v. 8, 2015.

SEMBA, S.; MUJUNI, E. An empirical performance comparison of meta-heuristic algorithms for school bus routing problem. **Tanzania Journal of Science**, v. 45, p. 81–92, 2019. ISSN 2507-7961.

SINTEF. **Solomon benchmark**. 2008. (Acessado em 11/10/2023). Disponível em: <https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/>.

SIPSER, M. **Introdução á teoria da computação**. Cengage: Cengage Learning, 2005. ISBN 9788522104994.

SOHANGHPURWALA, A. A.; HASSAN, M. W.; ATHANAS, P. Hardware accelerated sat solvers—a survey. **Journal of Parallel and Distributed Computing**, v. 106, p. 170–184, 2017. ISSN 0743-7315.

STRICKLAND, D. M.; BARNES, E.; SOKOL, J. S. Optimal protein structure alignment using maximum cliques. **Operations Research**, v. 53, n. 3, p. 389–402, 2005.

SULTANA, N. N.; BANIWAL, V.; BASUMATARY, A.; MITTAL, P.; GHOSH, S.; KHADILKAR, H. **Fast Approximate Solutions using Reinforcement Learning for Dynamic Capacitated Vehicle Routing with Time Windows**. 2021.

TARANTILIS, C.; IOANNOU, G.; PRASTACOS, G. Advanced vehicle routing algorithms for complex operations management problems. **Journal of Food Engineering**, v. 70, n. 3, p. 455–471, 2005. ISSN 0260-8774. Operational Research and Food Logistics.

UDDIN, M. B.; UDDIN, T. Solving constraint satisfaction problem in tsp using ga and dfs algorithms. v. 07, p. 417–421, 11 2020.

VASQUEZ, M.; HAO, J.-K. A “logic-constrained” knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. **Computational Optimization and Applications**, v. 20, n. 2, p. 137–157, Nov 2001. ISSN 1573-2894.

VIZEL, Y.; WEISSENBACHER, G.; MALIK, S. Boolean satisfiability solvers and their applications in model checking. **Proceedings of the IEEE**, v. 103, n. 11, p. 2021–2035, 2015.

VOUDOURIS, C.; TSANG, E. Guided local search and its application to the traveling salesman problem. **European Journal of Operational Research**, v. 113, n. 2, p. 469–499, 1999. ISSN 0377-2217.

XU, H.; RUTENBAR, R.; SAKALLAH, K. sub-sat: a formulation for relaxed boolean satisfiability with applications in routing. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 22, n. 6, p. 814–820, 2003.

ZELINKA, I.; SENKERIK, R.; BIALIC-DAVENDRA, M.; DAVENDRA, D. Chaos driven evolutionary algorithm for the traveling salesman problem. In: DAVENDRA, D. (Ed.). **Traveling Salesman Problem**. Rijeka: IntechOpen, 2010. cap. 4.

ZHA, A.; GAO, R.; CHANG, Q.; KOSHIMURA, M.; NODA, I. Cnf encodings for the min-max multiple traveling salesmen problem. In: . Baltimore, Maryland: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), 2020. p. 285–292.