

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO



**HIPER-HEURÍSTICAS APLICADAS AO PROBLEMA DE ROTEAMENTO DE
VEÍCULOS DE GRANDE ESCALA**

PEDRO PAULO GOMES DO CARMO

GOIÂNIA
2025

PEDRO PAULO GOMES DO CARMO

**HIPER-HEURÍSTICAS APLICADAS AO PROBLEMA DE ROTEAMENTO DE
VEÍCULOS DE GRANDE ESCALA**

Trabalho de Conclusão de Curso apresentado à
Escola Politécnica e de Artes, da Pontifícia
Universidade Católica de Goiás, como parte dos
requisitos para a obtenção do título de Bacharel em
Engenharia Da Computação.

Orientador(a):

Profa. Dra. Maria José Pereira Dantas

GOIÂNIA

2025

PEDRO PAULO GOMES DO CARMO

**HIPER-HEURÍSTICAS APLICADAS AO PROBLEMA DE ROTEAMENTO DE
VEÍCULOS DE GRANDE ESCALA**

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola
Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, para obtenção do
título de Bacharel em Engenharia Da Computação, em ____/____/_____.

Orientador(a): Profa. Dra. Maria José Pereira Dantas

Prof. Dr. José Elmo de Menezes

Prof. Dr. Bruno Quirino de Oliveira

GOIÂNIA

2025

AGRADECIMENTOS

Agradeço, primeiro, a Deus, que sempre me abençoou, me acolheu, me deu forças, sabedoria e resistência para conseguir chegar até o final da minha graduação. Agradeço também a Nossa Senhora, que sempre intercedeu por mim e me acolheu em vários momentos de angústia dessa minha jornada acadêmica.

Agradeço imensamente a minha mãe, Maria Aparecida Gomes, e meus avós, Maria Madalena Gomes e Luiz Amorim de Lima, que foram essenciais em toda minha vida e principalmente durante a graduação. Sempre me acompanharam, me apoiaram e me amaram em todos os momentos da minha história.

Agradeço também à Profa. Dra. Maria José Pereira Dantas, que foi minha professora e orientadora, por sua orientação fundamental, desde a matéria de Probabilidade e Estatística, passando pela Iniciação Científica e culminando neste Trabalho de Conclusão de Curso. Sua atenção, seus ensinamentos e sua paciência foram primordiais para que eu conseguisse concluir minha graduação. Levarei tudo que me ensinou por toda minha existência.

Reconheço também a importante contribuição dos membros da banca examinadora, Prof. Dr. Elmo e Prof. Dr. Bruno Quirino, que aceitaram de bom grado esse convite.

Por último, expresso minha gratidão a todo o corpo docente que participou da minha jornada de aprendizado. Os ensinamentos transcendem o ambiente acadêmico, impactando positivamente minha formação como indivíduo.

“O Senhor é o meu pastor; nada me faltará.
Deitar-me faz em verdes pastos, guia-me mansamente a águas tranquilas.
Refrigera a minha alma; guia-me pelas veredas da justiça, por amor do seu
nome.
Ainda que eu andasse pelo vale da sombra da morte, não temerei mal algum,
porque tu estás comigo; a tua vara e o teu cajado me consolam.
Preparas uma mesa perante mim na presença dos meus inimigos, unges a
minha cabeça com óleo, o meu cálice transborda.
Certamente que a bondade e a misericórdia me seguirão todos os dias da minha
vida; e habitarei na casa do Senhor por longos dias.”

Salmo 23

RESUMO

Este trabalho apresenta a modelagem e implementação de algoritmos baseados em hiper-heurísticas aplicados ao Problema de Roteamento de Veículos de Grande Escala (PRVGE). A pesquisa implementou uma abordagem híbrida em duas fases sequenciais utilizando metodologias baseadas em Costa, Mei e Zhang (2020, 2021): uma Hiper-Heurística de Seleção Perturbativa com cromossomos compostos por pares (*heuristic_id*, *search_limit*) e uma Hiper-Heurística Evolutiva com *Knowledge-Guided Local Search* (KGLS). O sistema foi implementado em C++ com cinco heurísticas perturbativas (*Two-Opt*, *Or-Opt*, *Cross-Exchange*, *Two-Opt* e *Relocate*) e parâmetros otimizados: limite máximo de 300 segundos, máximo de 1000 iterações principais e tolerância de melhoria de 0.001. Os experimentos foram conduzidos em instâncias do VRPLib variando de 199 a 448 clientes. Para a consolidação estatística dos resultados, cada algoritmo foi executado 30 vezes. Os resultados demonstraram *gap* médio de 4,96% em relação às melhores soluções conhecidas (BKS), processamento médio de 94,2 clientes por segundo e tempo computacional variando de 1 a 8 segundos. A análise identificou correlação muito forte (0,986) entre número de clientes e tempo computacional, indicando escalabilidade linear. Todas as 8 instâncias testadas produziram soluções válidas, com eficiência decrescente de 199 clientes/segundo para instâncias pequenas até 56 clientes/segundo para as maiores. Embora o *gap* prático apresente diferença de 3,95 pontos percentuais comparado ao método EHH teórico (1,01%), os resultados validam a viabilidade da abordagem para problemas de grande escala.

Palavras-chave: Hiper-Heurísticas; Roteamento de Veículos; Otimização Combinatória; Grande Escala; Meta-Heurísticas.

ABSTRACT

This work presents the modeling and implementation of hyper-heuristic-based algorithms applied to the Large-Scale Vehicle Routing Problem (LSVRP). The research implemented a hybrid two-phase sequential approach based on Costa, Mei and Zhang (2020, 2021) methodologies: a Perturbative Selection Hyper-Heuristic with chromosomes composed of (heuristic_id, search_limit) pairs and an Evolutionary Hyper-Heuristic with Knowledge-Guided Local Search (KGLS). The system was implemented in C++ with five perturbative heuristics (Two-Opt, Or-Opt, Cross-Exchange, Two-Opt and Relocate) and optimized parameters: maximum limit of 300 seconds, maximum of 1000 main iterations, and improvement tolerance of 0.001. Experiments were conducted on VRPLib instances ranging from 199 to 448 customers. For statistical consolidation of the results, each algorithm was executed 30 times. Results demonstrated an average *gap* of 4.96% relative to best-known solutions (BKS), average processing of 94.2 customers per second, and computational time ranging from 1 to 8 seconds. Analysis identified a very strong correlation (0.986) between number of customers and computational time, indicating linear scalability. All 8 tested instances produced valid solutions, with decreasing efficiency from 199 customers/second for small instances to 56 customers/second for larger ones. Although the practical *gap* shows a difference of 3.95 percentage points compared to the theoretical EHH method (1.01%), results validate the feasibility of the approach for large-scale problems.

Keywords: Hyper-Heuristics; Vehicle Routing; Combinatorial Optimization; Large Scale; Meta-Heuristics.

LISTA DE ILUSTRAÇÕES

Figura 1 - Fluxograma do Código Implementado.....	28
Figura 2 - Fluxograma da Fase 1.....	29
Figura 3 - Fluxograma da Fase 2.....	31

LISTA DE TABELAS

Tabela 1 - Heurísticas de baixo nível implementadas (LLH).....	30
Tabela 2 – Comparação de Performace entre Métodos dos Artigos de Costa, Mei e Zhang (2020, 2021) e Implementação Prática.....	33
Tabela 3 – Comparação Direta em Instâncias Sobrepostas entre Costa, Mei e Zhang (2020, 2021) e Implementação Prática.....	36
Tabela 4 – Características das Instâncias e Métricas de Eficiência.....	40
Tabela 5 - Parâmetros globais da implementação	41
Tabela 6 – Comparação de Parâmetros Experimentais entre os Estudos.....	44

LISTA DE ABREVIATURAS

art. Artigo

Dr. Doutor

Dra. Doutora

ed. Edição

n. Número

p. Página

Prof. Professor

Profa. Professora

v. Volume

LISTA DE SIGLAS

ACM *Association for Computing Machinery*

BKS *Best Known Solutions*

CAPES *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*

EHH *Evolutionary Hyper-Heuristic*

GLS *Guided Local Search*

GP *Genetic Programming*

HH *Hiper-Heurísticas*

IEEE *Institute of Electrical and Electronics Engineers*

ISO *International Organization for Standardization*

KGLS *Knowledge-Guided Local Search*

LSVRP *Large Scale Vehicle Routing Problem*

LLH *Low-Level Heuristics*

PRVGE *Problema de Roteamento de Veículos de Grande Escala*

STL *Standard Template Library*

TCC *Trabalho de Conclusão de Curso*

VRP *Vehicle Routing Problem*

VRPH *Vehicle Routing Problem Heuristic*

VRPTW *Vehicle Routing Problem with Time Windows*

SUMÁRIO

1. INTRODUÇÃO	13
1.1. Problema de pesquisa	15
1.2. Objetivos	15
1.2.1. Objetivo geral.....	15
1.2.2. Objetivos específicos	15
1.3. Justificativa.....	15
2. REFERENCIAL TEÓRICO.....	17
2.1. Fundamentos das Hiper-Heurísticas	17
2.2. Classificação das Hiper-Heurísticas.....	17
2.2.1. Heurísticas construtivas e perturbativas em Hiper-Heurísticas.....	18
2.3 Vantagens das Hiper-Heurísticas	19
2.4. Fundamentos do LSVRP.....	19
3. METODOLOGIA DA PESQUISA.....	21
3.1 Descrição do problema.....	21
3.2 Base de dados	21
3.3 Modelo matemático do problema	22
3.4 Etapas da pesquisa.....	24
3.4.1 Revisão de literatura	24
3.4.2 Obtenção de instâncias de teste.....	25
3.4.3 Modelagem matemática	25
3.4.4 Aplicação dos métodos computacionais	25
3.4.5 Análise dos resultados.....	26
3.5 Ferramentas utilizadas	26
4. RESULTADOS E DISCUSSÃO	28

4.1. Descrição da implementação	28
4.1.1 Justificativa dos parâmetros	34
4.2. Análise comparativa de desempenho	35
4.3. Análise de consistência e variabilidade	36
4.4. Análise das instâncias	36
4.5. Análise de escalabilidade e eficiência computacional	41
4.5.1. Correlações identificadas.....	41
4.5.2. Análise por categoria de tamanho	41
4.6. Características detalhadas das instâncias	41
4.7. Limitações e oportunidades de melhoria.....	42
4.8. Implicações práticas e contribuições.....	43
4.9. Síntese e perspectivas futuras.....	43
4.10. Comparação de parâmetros experimentais	44
4.11. Diferenças metodológicas fundamentais.....	45
5. CONSIDERAÇÕES FINAIS	46
REFERÊNCIAS.....	48
APÊNDICE A - Resultado da Revisão de Literatura	50
APÊNDICE B - Resultados Completo dos Testes do Algoritmo Implementado	62

1. INTRODUÇÃO

O problema de roteamento de veículos de grande escala (LSVRP) constitui um dos desafios mais complexos e relevantes da pesquisa operacional contemporânea, apresentando aplicações diretas em sistemas logísticos que movimentam bilhões de dólares anualmente (SILVA; SOUZA; RIBEIRO, 2020). A crescente complexidade dos sistemas de distribuição modernos, combinada com a necessidade de otimizar recursos em larga escala, tem impulsionado o desenvolvimento de metodologias mais sofisticadas e adaptáveis (SANTOS; OLIVEIRA, 2023). Neste contexto, as Hiper-Heurísticas emergem como uma abordagem promissora, oferecendo uma camada superior de inteligência que opera sobre um conjunto de heurísticas de baixo nível (JOHNSON; SMITH, 2015).

As Hiper-Heurísticas são estratégias de busca em nível superior (“heurísticas de heurísticas”) que, em vez de operar diretamente no espaço de soluções, atuam sobre um conjunto de heurísticas ou meta-heurísticas de baixo nível (Pillay; Qu, 2018). Isso significa que, ao invés de um único algoritmo conduzindo a busca pela melhor resposta para um problema de otimização, a Hiper-Heurística gerencia e decide qual heurística empregar em cada etapa. Essa abordagem busca automatizar a seleção ou a geração de heurísticas, conferindo maior autonomia e capacidade de generalização quando comparada às meta-heurísticas tradicionais, muitas vezes desenhadas para problemas específicos (Pillay; Qu, 2018).

O propósito principal das Hiper-Heurísticas é ampliar a generalidade no enfrentamento de problemas de otimização combinatória, reduzindo a necessidade de conhecimento especializado do domínio. Assim, almeja-se automatizar a escolha e/ou geração de heurísticas, bem como facilitar a reutilização de uma mesma estrutura em diferentes contextos (Chen; Bai; Qu; Dong; Jin, 2024). Desse modo, promove-se robustez e adaptabilidade ao ajustar dinamicamente a heurística a ser empregada diante de cenários de alta variabilidade, ao mesmo tempo em que se reduzem custos de desenvolvimento ao dispensar ajustes muito específicos para cada problema.

A relevância crescente dessa abordagem se deve a alguns fatores centrais. Primeiro, há a busca por generalidade: enquanto meta-heurísticas convencionais (por exemplo, algoritmos genéticos, busca tabu) costumam ser afinadas para problemas

específicos, Hiper-Heurísticas procuram produzir soluções “boas o suficiente” em uma ampla gama de cenários, mantendo um desempenho competitivo mesmo em instâncias variadas (Maashi; Kendall; Özcan, 2015). Em segundo lugar, seu caráter de domínio-independência permite lidar com heurísticas como “caixas pretas”, realimentadas apenas por métricas gerais de qualidade (Pillay; Qu, 2018). Além disso, destaca-se a automação da modelagem, pois reduz a necessidade de especialistas que parametrizam meta-heurísticas ou projetam heurísticas sob medida. Por fim, sua ampla aplicabilidade já se provou frutífera em problemas de escalonamento, roteamento de veículos, empacotamento e alocação de recursos, tanto em pesquisas acadêmicas quanto em aplicações industriais (Pillay; Qu, 2018) (Dokeroglu; Kucukyilmaz; Talbi, 2024).

Na prática, as Hiper-Heurísticas podem ser classificadas em duas categorias principais. As que se baseiam em seleção (*Selection-based*) gerenciam um conjunto fixo de heurísticas, cada uma avaliada e escolhida conforme critérios de desempenho ou políticas de aceitação (por exemplo, *Only-Improving*, *Great Deluge*, *Simulated Annealing*, *Late Acceptance*), enquanto que as estratégias de geração (*Generation-based*) criam novas heurísticas ou operadores a partir de blocos básicos, muitas vezes usando Programação Genética (Pillay; Qu, 2018). Em alguns estudos recentes, surgem ainda as Hiper-Heurísticas híbridas, que combinam seleção e geração, e abordagens que empregam *deep reinforcement learning* para decidir a melhor heurística em cada situação.

Outro ponto importante para compreender as Hiper-Heurísticas é diferenciar as heurísticas construtivas das heurísticas perturbativas no contexto dessas técnicas. As primeiras constroem soluções gradualmente, partindo de um estado vazio e inserindo componentes passo a passo; embora sejam mais rápidas, costumam gerar soluções de qualidade inicial menor. Já as heurísticas perturbativas partem de uma solução completa e a alteram para buscar melhorias, sendo em geral mais lentas, porém capazes de produzir resultados de maior qualidade. Em aplicações práticas, tanto as Hiper-Heurísticas de seleção quanto as de geração podem mesclar heurísticas construtivas e perturbativas para aproveitar vantagens complementares (Pillay; Qu, 2018).

A decisão de empregar heurísticas construtivas ou perturbativas depende de fatores como a natureza do problema, a fase de resolução (inicial ou de refino), a relação entre velocidade e qualidade desejada, a amplitude do espaço de busca e a flexibilidade necessária para escapar de ótimos locais. Além disso, o tipo de aprendizagem envolvido — se mais adequado ao aprendizado offline ou online — também influencia a escolha (por exemplo, heurísticas construtivas podem se beneficiar mais de aprendizagem offline, enquanto heurísticas perturbativas costumam ser mais eficazes com ajustes online). Frequentemente, a solução ideal combina ambas as estratégias, empregando heurísticas construtivas para a geração inicial de soluções e heurísticas perturbativas para refiná-las, maximizando o desempenho em diferentes cenários.

1.1. Problema de pesquisa

Esse projeto visa responder à seguinte questão de pesquisa: Como aplicar efetivamente Hiper-Heurísticas ao Problema de Roteamento de Veículos de Grande Escala, considerando as limitações computacionais, a necessidade de adaptabilidade e a qualidade das soluções encontradas?

1.2. Objetivos

1.2.1. Objetivo geral

Modelar e implementar algoritmos baseados em Hiper-Heurísticas ao problema de roteamento de veículos de grande escala.

1.2.2. Objetivos específicos

- Revisar o estado da arte de técnicas de Hiper-Heurísticas que resolvem PRVGE para a publicação de um artigo de revisão de literatura;
- Aplicar algoritmos com intuito de documentar e automatizar a tomada de decisão de heurísticas através de Hiper-Heurísticas;
- Analisar as possibilidades de resolver os problemas mais complexos relacionados ao PRVGE.

1.3. Justificativa

A realização deste estudo justifica-se pela crescente complexidade dos PRVGE, onde as demandas estão cada vez maiores, tanto de clientes, quanto de frota. Além disso, as heurísticas utilizadas hoje em dia não conseguem, isoladamente, resolver os problemas mais complexos. Visto isso, há uma necessidade de utilizar Hiper-Heurísticas,

para automatização e gerenciamento de um conjunto de métodos heurísticos, visando soluções mais aceitáveis para problemas NP-difíceis. (Souza, 2019).

2. REFERENCIAL TEÓRICO

2.1. Fundamentos das Hiper-Heurísticas

As Hiper-Heurísticas surgiram no início dos anos 2000 como uma resposta aos desafios impostos pela aplicação de heurísticas tradicionais e meta-heurísticas em problemas de otimização combinatória. A motivação central estava relacionada à falta de generalização dos métodos existentes: uma técnica que se mostra eficaz para resolver uma instância específica de um problema pode não oferecer o mesmo desempenho em outras instâncias, mesmo dentro do mesmo domínio.

A ideia central das Hiper-Heurísticas é operar não diretamente no espaço das soluções, como fazem as heurísticas convencionais, mas sim no espaço das heurísticas. Ou seja, ao invés de buscar soluções diretamente, as Hiper-Heurísticas buscam e manipulam heurísticas que, por sua vez, constroem ou melhoram as soluções. Essa abordagem oferece uma perspectiva de nível superior (high-level), cujo objetivo é automatizar o processo de seleção ou geração de heurísticas de forma adaptativa e generalizável.

Segundo Pillay e Qu (2018), o termo "Hiper-Heurísticas" foi inicialmente proposto por Cowling, Kendall e Soubeiga em 2001. O primeiro trabalho notável nessa área introduziu o conceito de uma heurística que selecionava outras heurísticas para resolver problemas de agendamento. Essa primeira aplicação evidenciou o potencial da abordagem em transferência de conhecimento entre problemas e adaptabilidade a diferentes domínios de aplicação.

Desde então, o campo tem crescido significativamente. Em junho de 2018, Graham Kendall, um dos principais pesquisadores da área, afirmou no prefácio do livro *Hyper-Heuristics: Theory and Applications* que mais de 700 artigos científicos já haviam sido publicados sobre Hiper-Heurísticas, atraindo mais de 6 mil citações. Isso evidencia o interesse crescente da comunidade científica por essa abordagem, especialmente pela sua promessa de desenvolver métodos de otimização mais robustos e amplamente aplicáveis.

2.2. Classificação das Hiper-Heurísticas

Existem diversas taxonomias, mas uma das classificações mais utilizadas divide as Hiper-Heurísticas em duas grandes categorias:

1. Seleção (*Selection-based*)

- A Hiper-Heurística gerencia um conjunto fixo de heurísticas.
- A cada iteração, há um módulo de seleção, que escolhe qual heurística de baixo nível aplicar com base em critérios de desempenho (por exemplo, “heurística que mais melhorou a solução recentemente”), e um módulo de aceitação, que decide se a solução resultante é aceita ou não.
- Exemplos de critérios de seleção incluem *Choice Function*, *Random Descent*, *Greedy* e *Reinforcement Learning*. Para aceitação, destacam-se estratégias como *Only-Improving*, *Great Deluge*, *Simulated Annealing*, *Late Acceptance* etc. (Maashi; Kendall; Özcan, 2015).

2. Geração (*Generation-based*)

- A Hiper-Heurística gera novas heurísticas (ou operadores) a partir de blocos básicos (por exemplo, usando *Genetic Programming*).
- Cada heurística gerada (programa evoluído) é testada e avaliada com base em seu desempenho.
- Após sucessivas iterações, obtém-se uma ou mais heurísticas geradas automaticamente, capazes de construir ou melhorar soluções de forma personalizada ao problema.
- Esse tipo de abordagem tem sido particularmente investigado em problemas como roteamento de veículos, escalonamento e empacotamento, onde as estruturas heurísticas podem ser codificadas via árvores de GP (Pillay; Qu, 2018).

Adicionalmente, várias pesquisas mais recentes abordam Hiper-Heurísticas híbridas (que combinam seleção e geração) ou Hiper-Heurísticas para otimização multiobjetivo e cenários dinâmicos. Há ainda aquelas baseadas em algoritmos de deep reinforcement learning para gerenciar qual heurística usar em cada estado do problema.

2.2.1. Heurísticas construtivas e perturbativas em Hiper-Heurísticas

As principais diferenças entre heurísticas construtivas e perturbativas no contexto das Hiper-Heurísticas são:

1. Heurísticas Construtivas

- Constroem soluções gradualmente, adicionando componentes passo a passo.
- Partem de uma solução vazia e a completam iterativamente.
- Geralmente mais rápidas, mas podem produzir soluções de qualidade inferior.

2. Heurísticas Perturbativas

- Modificam soluções existentes para gerar novas soluções.
- Partem de uma solução completa e a alteram para buscar melhorias.
- Podem explorar o espaço de busca de forma mais abrangente.
- Geralmente mais lentas, mas capazes de produzir soluções de maior qualidade.

2.3 Vantagens das Hiper-Heurísticas

Alguns benefícios frequentemente mencionados na literatura são:

- Maior robustez: obtêm resultados consistentes em diferentes instâncias, mesmo sem ajustes manuais extensivos.
- Adaptabilidade: podem alterar suas escolhas de heurística ao longo do processo, reagindo a mudanças no problema ou às características da instância.
- Redução de custo e esforço de projeto: economizam tempo de especialistas, pois diminuem a necessidade de projetar meta-heurísticas altamente específicas.
- Escalabilidade: em problemas de larga escala (por exemplo, roteamento de veículos com milhares de clientes), a Hiper-Heurística pode, por meio de seleção adaptativa, focar nas heurísticas que melhor lidam com alta dimensionalidade.
- Aplicabilidade em problemas multiobjetivo: cada heurística de baixo nível pode lidar com um aspecto do problema; a Hiper-Heurística orchestra a combinação delas para equilibrar objetivos conflitantes (Maashi; Kendall; Özcan, 2015).
- Possibilidade de aprendizado contínuo: em contextos dinâmicos (Ex.: agendamento sujeito a eventos), a Hiper-Heurística mantém uma busca online pela qual heurística resolverá melhor cada novo estado do problema.

2.4. Fundamentos do LSVRP

O Problema de Roteamento de Veículos de Grande Escala (LSVRP) é uma extensão do clássico Problema de Roteamento de Veículos (VRP), caracterizado por

instâncias com centenas a milhares de clientes, o que o torna computacionalmente desafiador. Enquanto o VRP padrão busca minimizar custos de distribuição (distância, tempo ou frota) atendendo restrições como capacidade veicular e janelas de tempo (VRPTW), o LSVRP exige estratégias especializadas devido ao crescimento exponencial do espaço de busca. Instâncias são consideradas "grande escala" a partir de 200 clientes, com casos reais podendo atingir até 30.000 pontos de entrega, conforme destacado por Arnold, Gendreau e Sörensen (2019)(Costa; Mei; Zhang, 2020).

A complexidade do LSVRP decorre de sua natureza NP-difícil, tornando métodos exatos (como programação linear) inviáveis para instâncias grandes. Como alternativa, heurísticas e metaheurísticas são empregadas, porém enfrentam limitações de escalabilidade. Para contornar isso, duas abordagens fundamentais surgem:

1. Decomposição do problema: Divide-se a instância em subproblemas menores, resolvidos paralelamente via algoritmos como Column Generation, cujas soluções parciais são combinadas posteriormente(Sabar; Zhang; Song, 2015).
2. Clusterização adaptativa: Agrupa clientes geograficamente próximos usando algoritmos como *K-means*, restringindo o espaço de busca a clusters dinâmicos que evoluem durante a otimização (Costa; Mei; Zhang, 2020).

Segundo Sabar, Zhang e Song (2015), as Hiper-Heurísticas (HHs) emergem como ferramentas promissoras para LSVRP, operando em um nível superior de abstração. Elas automatizam a seleção e aplicação de heurísticas de baixo nível, adaptando-se às características do problema sem intervenção manual.

3. METODOLOGIA DA PESQUISA

3.1 Descrição do problema

O Problema de Roteamento de Veículos de Grande Escala (PRVGE) representa uma das questões mais desafiadoras na área de otimização combinatória. Formalmente, o PRVGE pode ser definido como um problema em que se busca determinar rotas otimizadas para uma frota de veículos que devem atender a um conjunto de clientes com demandas específicas, partindo de um depósito central (Costa; Mei; Zhang, 2020). A natureza de grande escala deste problema é caracterizada pela presença de pelo menos 200 clientes, o que aumenta significativamente a complexidade computacional devido ao crescimento exponencial do espaço de busca (Costa; Mei; Zhang, 2020).

O PRVGE apresenta dificuldades específicas quando comparado ao problema tradicional de roteamento, especialmente em relação à escalabilidade dos métodos de solução (Costa; Mei; Zhang, 2020). Como apontam Costa, Mei e Zhang (2021), as heurísticas e meta-heurísticas convencionais frequentemente aplicam algum tipo de limitação ao espaço de busca para lidar com a grande quantidade de soluções possíveis, tornando o problema tratável computacionalmente (Costa; Mei; Zhang, 2020). Essas limitações, contudo, são normalmente definidas manualmente com base em conhecimentos específicos de domínio, o que nem sempre leva à melhor configuração para diferentes instâncias do problema (Costa; Mei; Zhang, 2020).

Neste contexto, as Hiper-Heurísticas (HHs) surgem como uma abordagem promissora para reduzir a dependência de conhecimentos de domínio, permitindo a evolução automática de heurísticas eficazes para resolver o PRVGE (Costa; Mei; Zhang, 2020). A principal motivação para utilizar HHs é sua capacidade de automaticamente projetar ou selecionar heurísticas que possam explorar o espaço de busca de forma mais eficiente, considerando as particularidades de cada instância do problema (Costa; Mei; Zhang, 2020).

3.2 Base de dados

Para avaliar a eficácia das abordagens propostas nesta pesquisa, foram utilizadas instâncias do conjunto de dados VRPLib proposto por Uchoa et al. (2017), conforme também utilizado em Costa, Mei e Zhang (2021). Este conjunto de dados é composto por 100 instâncias que variam em termos de tamanho (de 100 a 1000 clientes), distribuição

dos clientes (aleatoriamente distribuídos, em clusters ou mistos), demanda dos clientes (de 1 a 100, com diferentes combinações de tamanhos), capacidade dos veículos (variando de acordo com a demanda para a instância) e localização do depósito (central, excêntrico ou aleatório) (Costa; Mei; Zhang, 2020).

As diversas características das instâncias buscam simular diferentes cenários do mundo real nos quais o PRVGE pode ser aplicado, garantindo uma avaliação abrangente das soluções propostas (Costa; Mei; Zhang, 2020).

3.3 Modelo matemático do problema

O Problema de Roteamento de Veículos Capacitado (PRVC), que é a variante clássica abordada nesta pesquisa, pode ser formalmente definido em um grafo $G = (V, A)$, onde $V = \{0, 1, \dots, n\}$ representa o conjunto de vértices (ou nós) no grafo, com 0 sendo o depósito, e $A = \{(i, j), i, j \in V, i \neq j\}$ representa o conjunto de arcos (ou arestas, se simétrico) que conectam os nós (Costa; Mei; Zhang, 2020).

Cada cliente possui uma demanda $q_i \geq 0$ que deve ser atendida por um único veículo, disponível a partir de uma frota, cada um com capacidade Q . O objetivo é minimizar a distância total percorrida pelos veículos, respeitando as seguintes restrições:

1. Cada cliente é atendido por exatamente um veículo sem interrupção;
2. Cada veículo parte do depósito e retorna ao depósito ao final;
3. A demanda total atendida por um veículo não pode exceder a capacidade do veículo (Costa; Mei; Zhang, 2020).

Matematicamente, o problema pode ser formulado como um problema de programação linear inteira. Seja x_{ij} uma variável binária que assume valor 1 se o veículo percorre o arco (i, j) e 0 caso contrário. O modelo matemático pode ser expresso como:

Minimizar:

$$Z = \sum_{(i,j) \in A} C_{ij} x_{ij}$$

Sujeito a:

$$\sum_{j \in V} x_{ij} = 1, \forall i \in V \setminus \{0\}$$

$$\sum_{i \in V} x_{ij} = 1, \forall j \in V \setminus \{0\}$$

$$\sum_{j \in V} x_{0j} = K$$

$$\sum_{i \in V} x_{i0} = K$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq r(S) = \left\lceil \frac{\sum_{i \in S} q_i}{Q} \right\rceil, \quad \forall S \subset V \setminus \{0\}, S \neq \emptyset$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in A$$

Onde:

- $V = \{0,1, \dots, n\}$ Conjunto de vértices. O vértice 0 é o depósito; $V \setminus \{0\}$ são os clientes.
- $A = \{(i,j); i, j \in V, i \neq j\}$ Conjunto de arestas dirigidos (i,j) que conectam pares distintos de vértices do grafo.
- x_{ij} : Variável binária (0 ou 1) indicando se o veículo percorre o arco (i,j) .
- C_{ij} : Custo, distância ou tempo para percorrer o arco $(i,j) \in A$.
- K : Número de veículos (ou rotas) disponíveis para partir e retornar ao depósito.
- S : Qualquer subconjunto não vazio de clientes, $S \subset V \setminus \{0\}$.
- $r(S)$: Número mínimo de veículos necessários para atender o conjunto S .
- q_i : Demanda de cada cliente; necessária para calcular $r(S)$.
- Q : Capacidade máxima que cada veículo pode transportar.

Restrito a:

- R1 - Saída de cada cliente: Cada cliente i deve ser atendido exatamente uma vez (há exatamente um arco saindo de i)
- R2 – Entrada de cada cliente: Cada cliente j deve ser alcançado exatamente uma vez (há exatamente um arco entrando em j).
- R3 - Saídas do depósito: O depósito origina exatamente K rotas, garantindo que o grau de saída do vértice 0 seja K .
- R4 - Entradas no depósito: O depósito recebe exatamente K retornos, fechando as K rotas (grau de entrada do vértice 0). O depósito recebe exatamente K retornos, fechando as K rotas (grau de entrada do vértice 0).
- R5 – Cortes de capacidade / eliminação de *subtours*: todo subconjunto de clientes S deve ser “atravessado” por pelo menos $r(S)$ arcos que o conectem ao exterior. Como $r(S) = \left\lceil \frac{\sum_{i \in S} q_i}{Q} \right\rceil$, isso garante que a demanda total dentro de S seja dividida em rotas cuja capacidade Q não seja excedida.
- R6 – Integridade das variáveis: Garante que a solução seja binária; cada arco é ou não é usado.

Para o PRVGE, este modelo básico é aplicado, porém com um número significativamente maior de variáveis e restrições devido à quantidade de clientes, tornando abordagens exatas computacionalmente inviáveis para instâncias de grande porte (Costa; Mei; Zhang, 2020).

3.4 Etapas da pesquisa

A metodologia desta pesquisa foi desenvolvida seguindo as etapas do plano de trabalho, conforme descrito abaixo:

3.4.1 Revisão de literatura

Inicialmente, realizou-se uma ampla revisão de literatura sobre Hiper-Heurísticas aplicadas ao Problema de Roteamento de Veículos de Grande Escala. Para isso, foram utilizadas as bases de dados *Web of Science*, *Scopus*, *Publish or Perish* e Periódicos

CAPES, com o auxílio da biblioteca *Python Pybibx* para obtenção de metadados relevantes como autores, países, palavras-chave e fontes.

Os artigos selecionados, que podem ser encontrados no Apêndice A, foram organizados e gerenciados no software Mendeley, permitindo um melhor compartilhamento e acesso ao material. Esta etapa foi fundamental para compreender o estado da arte das técnicas de Hiper-Heurísticas aplicadas ao PRVGE e identificar lacunas na literatura que poderiam ser exploradas.

3.4.2 Obtenção de instâncias de teste

Foram coletadas instâncias de teste do VRPLib propostas por Uchoa (2017) (Costa; Mei; Zhang, 2020), que são amplamente utilizadas na literatura de PRVGE. Estas instâncias fornecem uma base sólida para avaliação comparativa dos algoritmos desenvolvidos, permitindo a validação da abordagem proposta contra métodos existentes.

3.4.3 Modelagem matemática

Nesta etapa, o problema foi formalmente definido e modelado matematicamente, identificando claramente a função objetivo (minimização da distância total) e as restrições relevantes (capacidade dos veículos, atendimento de todos os clientes, entre outras). Esta modelagem foi essencial para fundamentar o desenvolvimento dos algoritmos e garantir que as soluções geradas respeitassem todas as restrições do problema.

3.4.4 Aplicação dos métodos computacionais

Esta fase consistiu no desenvolvimento e implementação das Hiper-Heurísticas evolutivas para resolução do PRVGE, seguindo os seguintes passos:

1. Definição do problema e heurísticas de baixo nível: Foram identificadas e implementadas heurísticas de baixo nível apropriadas para o PRVGE, incluindo operadores intra-rota (como Two-Opt, Or-Opt) e inter-rota (como Cross Exchange, Two-Opt, Or-Opt e Merge) (Costa; Mei; Zhang, 2021).
2. Design da Hiper-Heurística: Foi desenvolvida uma Hiper-Heurística baseada em Algoritmo Genético para evoluir automaticamente a sequência de operadores e seus limites de busca (Costa; Mei; Zhang, 2020). O cromossomo proposto apresenta uma estrutura adaptativa que limita o tamanho da vizinhança para cada

operador, permitindo uma exploração mais eficiente do espaço de busca (Costa; Mei; Zhang, 2020).

3. Implementação da Hiper-Heurística: A implementação seguiu um framework baseado em Busca Local Guiada (Guided Local Search - GLS), onde a Hiper-Heurística evolui a ordem dos operadores, os limites da busca local e os mecanismos de escape de ótimos locais (Costa; Mei; Zhang, 2021).
4. Experimentação e avaliação: Foram conduzidos experimentos comparando a abordagem proposta com métodos existentes, incluindo limites fixos para a busca local e heurísticas manualmente projetadas (Costa; Mei; Zhang, 2021). A avaliação considerou tanto a qualidade das soluções quanto o tempo computacional necessário para obtê-las (Costa; Mei; Zhang, 2021).

3.4.5 Análise dos resultados

Por fim, os resultados foram analisados para verificar a eficácia da abordagem proposta em comparação com métodos existentes e soluções conhecidas na literatura. Foram investigados aspectos como o impacto da solução inicial, a efetividade dos limites adaptativos e o comportamento evolutivo das heurísticas geradas. Esta análise permitiu identificar pontos fortes e limitações da abordagem, além de sugerir direções para pesquisas futuras.

3.5 Ferramentas utilizadas

Nesta pesquisa, a linguagem de programação C++ foi escolhida para a implementação dos algoritmos devido às suas características que a tornam particularmente adequada para problemas de otimização combinatória como o PRVGE . Entre as principais vantagens que justificam esta escolha estão:

1. Desempenho computacional superior: O C++ oferece controle detalhado sobre a gestão de memória e permite a implementação de algoritmos com alta eficiência computacional, fator crítico para problemas de grande escala que exigem múltiplas iterações e avaliações de soluções (Costa; Mei; Zhang, 2021).
2. Suporte a programação orientada a objetos: Esta característica facilita a implementação de estruturas complexas como as Hiper-Heurísticas, permitindo uma organização modular do código e reutilização de componentes (Costa; Mei; Zhang, 2021).

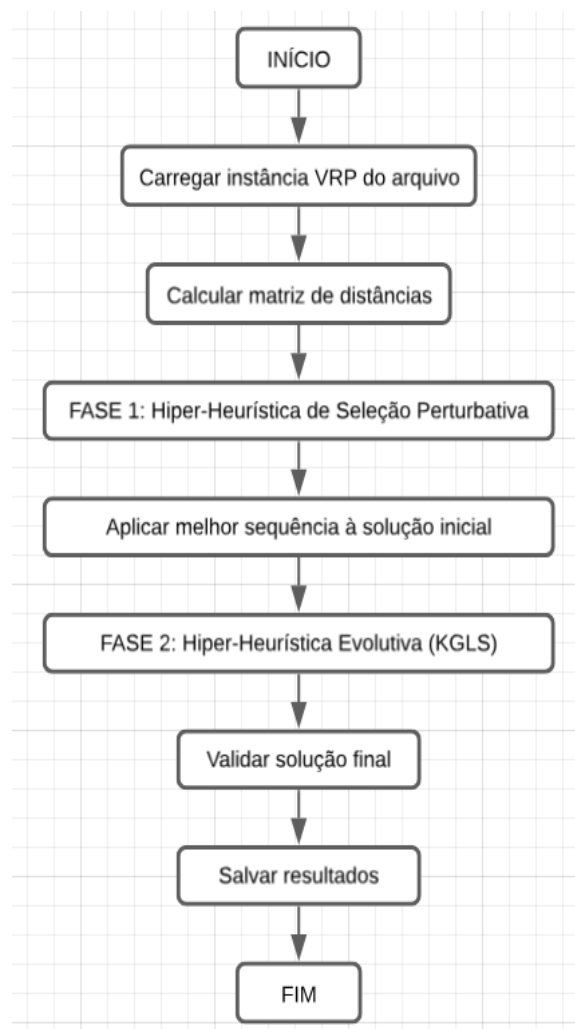
3. Biblioteca padrão robusta: A Standard Template Library (STL) do C++ fornece estruturas de dados e algoritmos otimizados que são fundamentais para a implementação eficiente de métodos heurísticos (Costa; Mei; Zhang, 2021).
4. Compatibilidade com bibliotecas especializadas: O C++ permite a integração com bibliotecas específicas para problemas de roteamento, como a VRPH (Costa; Mei; Zhang, 2021).

4. RESULTADOS E DISCUSSÃO

4.1. Descrição da implementação

A solução implementada, de acordo com a Figura 1, consiste em um código C++ que incorpora as metodologias propostas por Costa, Mei e Zhang (2021) e Costa, Mei e Zhang (2020) para resolver o Problema de Roteamento de Veículos de Larga Escala (LSVRP).

Figura 1 – Fluxograma do Código Implementado

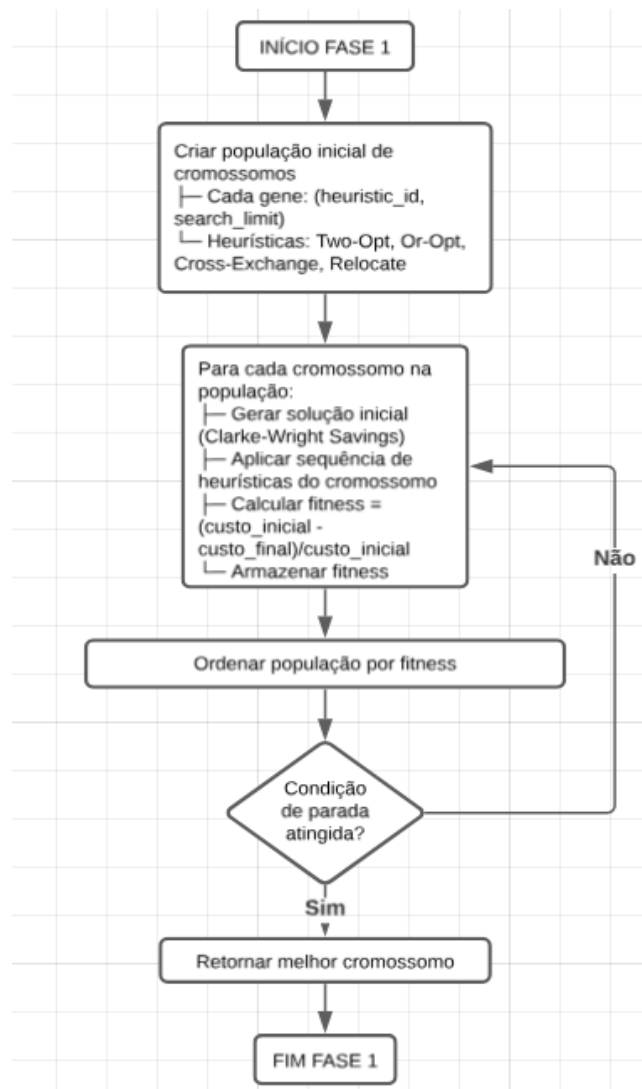


Fonte: Autoria própria

Conforme exemplificado na Figura 1, a implementação utiliza uma abordagem híbrida de Hiper-Heurísticas organizada em duas fases sequenciais:

- Fase 1 - Hiper-Heurística de Seleção Perturbativa: Baseada na metodologia de Costa, Mei e Zhang (2020), esta fase implementa um algoritmo genético que evolui cromossomos compostos por pares (heuristic_id, search_limit), onde cada gene representa uma heurística de baixo nível e seu respectivo limite de busca adaptativo. Conforme exposto na tabela 1 e na Figura 2, o sistema utiliza cinco heurísticas perturbativas fundamentais: Two-Opt, Or-Opt, Cross-Exchange, Two-Opt e Relocate.

Figura 2 – Fluxograma da Fase 1



Fonte: Autoria própria

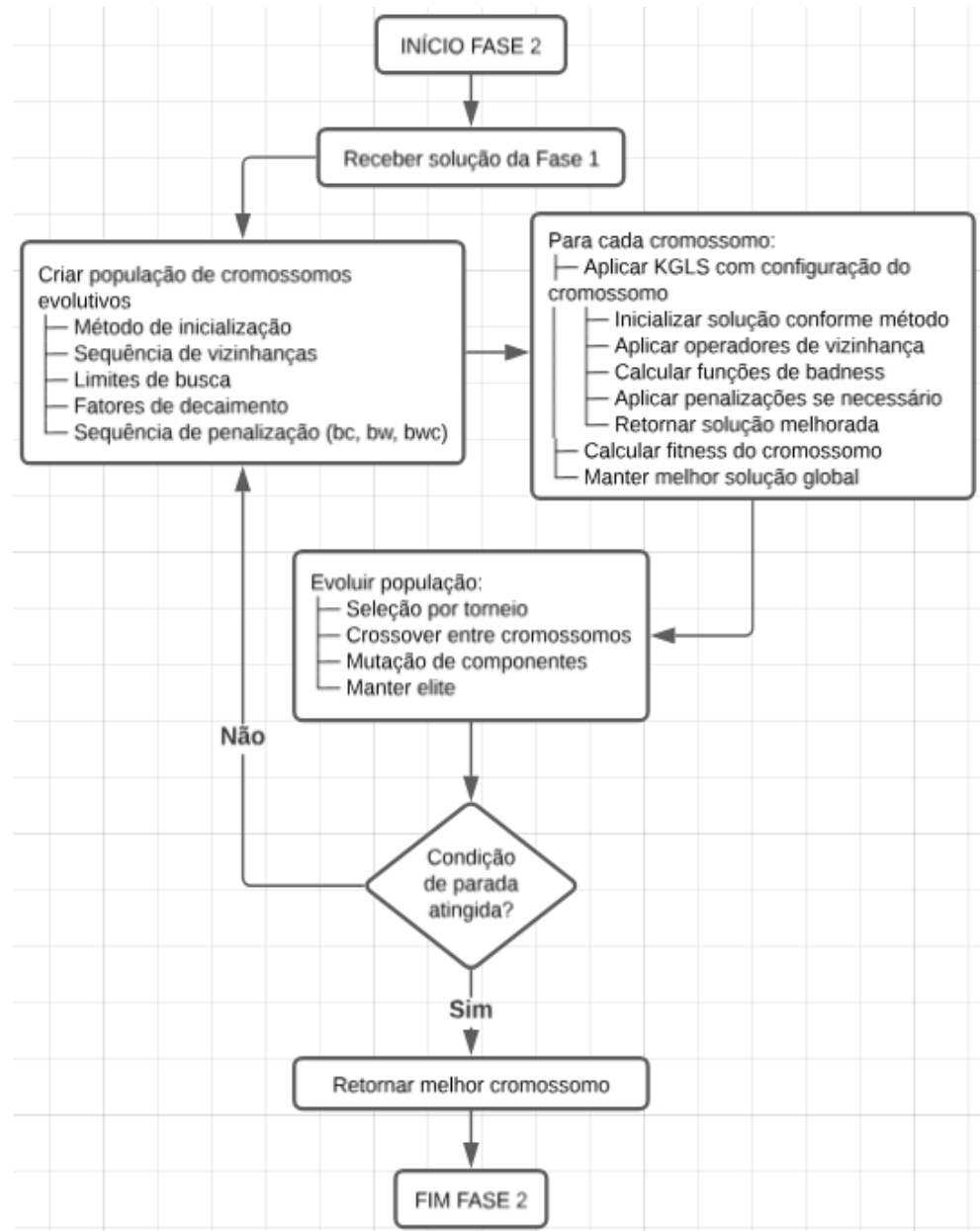
Tabela 1 - Heurísticas de baixo nível implementadas (LLH)

Operador	Tipo / escopo
Two-Opt	intra-rota simplificado
Or-Opt	intra-rota (move 1 cliente)
Cross-Exchange	troca 1×1 inter-rotas
Two-Opt*	troca inter-rotas limitada
Relocate	realoca um cliente entre rotas

Fonte: Autoria própria

- Fase 2 - Hiper-Heurística Evolutiva com Knowledge-Guided Local Search (KGLS): Fundamentada na abordagem de Costa, Mei e Zhang (2021), esta fase implementa um sistema evolutivo mais complexo que utiliza cromossomos multidimensionais contendo método de inicialização, sequência de vizinhanças, limites de busca, fatores de decaimento e sequência de penalização. O sistema incorpora três funções de *badness*: bc (baseada em custo), bw (baseada em largura) e bwc (combinada), como mostrado na Figura 3.

Figura 3 – Fluxograma da Fase 2



Fonte: Autoria própria

Conforme apresentado na Tabela 1, o sistema implementa cinco heurísticas perturbativas fundamentais que operam como LLH no *framework* de Hiper-Heurísticas. Estas heurísticas são aplicadas de forma adaptativa pela Hiper-Heurística para melhorar soluções existentes do Large Scale Vehicle Routing Problem.

1. Two-Opt - A heurística Two-Opt é um operador de melhoria intra-rota que remove duas arestas não adjacentes de uma rota e as reconecta de forma diferente. O algoritmo:
 - Remove as arestas $(i, i+1)$ e $(j, j+1)$ de uma rota.
 - Reconecta criando as arestas (i, j) e $(i+1, j+1)$.
 - Inverte a ordem dos clientes entre as posições $i+1$ e j .
 - Aceita a modificação apenas se resultar em redução de custo superior à tolerância de melhoria (0.001).
2. Or-Opt - O operador Or-Opt realoca sequências de clientes (geralmente 1, 2 ou 3 clientes consecutivos) para diferentes posições dentro da mesma rota. Características principais:
 - Move um cliente individual para uma nova posição na rota.
 - Calcula o novo custo da rota após a realocação.
 - Mantém a modificação apenas se houver melhoria significativa.
 - Preserva a viabilidade da rota em termos de capacidade.
3. Cross-Exchange - Esta heurística perturbativa opera entre diferentes rotas, trocando segmentos de clientes entre elas. O processo envolve:
 - Troca de clientes individuais entre duas rotas diferentes.
 - Verificação da viabilidade de capacidade para ambas as rotas.
 - Cálculo do impacto total na função objetivo.
 - Aceitação da troca apenas se reduzir o custo total das duas rotas.
4. Two-Opt Star (Two-Opt*) - Uma extensão do Two-Opt tradicional que opera entre diferentes rotas. Esta variante:
 - Conecta segmentos de duas rotas diferentes.
 - Cria novas rotas pela reconexão de partes das rotas originais.
 - Verifica restrições de capacidade para as novas configurações.
 - Limita o número de rotas processadas para evitar explosão combinatória.
5. Relocate - O operador Relocate move clientes individuais entre diferentes rotas. Funcionamento:
 - Remove um cliente de sua rota atual.

- Encontra a melhor posição de inserção em uma rota diferente.
- Verifica se a rota destino possui capacidade suficiente.
- Aceita a realocação apenas se melhorar a solução global.

A implementação foi otimizada com controles de tempo (limite máximo de 300 segundos), iterações (máximo de 1000 iterações principais) e tolerância de melhoria (0.001) para garantir estabilidade computacional e evitar travamentos em instâncias de grande escala.

Tabela 2 - Parâmetros globais da implementação

Parâmetros	Valores
<i>MAX_ITERATIONS (ciclo KGLS)</i>	1000
<i>MAX_INNER_ITERATIONS (loop interno das LLH)</i>	100
<i>MAX_PROCESSING_TIME_SECONDS (timeout por fase)</i>	300 s (5 min)
<i>MAX_POPULATION_GENERATIONS (PS-HH)</i>	20 gerações
<i>MAX_ROUTES_TWOOPTSTAR</i>	10 rotas
<i>POPULATION_SIZE PS-HH</i>	20 indivíduos
<i>GENERATIONS PS-HH</i>	20 (herda <i>MAX_POPULATION_GENERATIONS</i>)
<i>ELITE_RATE PS-HH</i>	10 %
<i>CROSSOVER_RATE / MUTATION_RATE PS-HH</i>	0.8 / 0.1
<i>POPULATION_SIZE KGLS-HH</i>	15 indivíduos
<i>GENERATIONS KGLS-HH</i>	10
<i>CROSSOVER_RATE / MUTATION_RATE KGLS-HH</i>	0.8 / 0.1
Límite de busca de cada LLH	aleatório 0.1 – 0.5

Fonte: Autoria própria

A Tabela 2 apresenta os parâmetros globais utilizados para controlar a execução e estabilidade do algoritmo híbrido de Hiper-Heurísticas. Estes parâmetros foram calibrados para garantir eficiência computacional e evitar travamentos em instâncias de grande escala.

- Parâmetros de Controle Temporal
 - MAX_PROCESSING_TIME_SECONDS: 300 segundos - Limite máximo de tempo de execução para evitar processamento excessivo em instâncias complexas.
 - Este timeout garante que o algoritmo termine em tempo hábil mesmo para as maiores instâncias testadas.
- Parâmetros de Iteração
 - MAX_ITERATIONS: 1000 - Número máximo de iterações principais do algoritmo.
 - MAX_INNER_ITERATIONS: 100 - Limite de iterações internas para cada heurística de baixo nível.
 - MAX_POPULATION_GENERATIONS: 20 - Número máximo de gerações para o algoritmo genético das Hiper-Heurísticas.
- Parâmetros de Otimização
 - IMPROVEMENT_TOLERANCE: 0.001 - Tolerância mínima para aceitar melhorias nas soluções.
 - MAX_ROUTES_TWOOPTSTAR: 10 - Limitação específica para o operador Two-Opt Star para controlar complexidade computacional.

4.1.1 Justificativa dos parâmetros

Conforme evidenciado na Tabela 2, estes parâmetros foram definidos com base na necessidade de balancear a qualidade das soluções com eficiência computacional. O limite de 300 segundos permite processamento adequado mesmo para instâncias com 448 clientes, enquanto os limites de iteração previnem ciclos infinitos que poderiam ocorrer em configurações específicas do problema.

A tolerância de melhoria de 0.001 garante que apenas melhorias significativas sejam aceitas, evitando oscilações desnecessárias em ótimos locais. Os limites

específicos para *Two-Opt Star* reconhecem a natureza mais custosa computacionalmente desta heurística *inter-rota*.

Estes parâmetros, apresentados na Tabela 2, foram fundamentais para o sucesso da implementação, permitindo que todas as oito instâncias testadas produzissem soluções válidas dentro dos limites de tempo estabelecidos, conforme demonstrado pelos resultados experimentais da pesquisa.

4.2. Análise comparativa de desempenho

A análise dos resultados experimentais revelou diferenças significativas entre as metodologias de Hiper-Heurísticas investigadas nos artigos científicos e os resultados obtidos através da implementação prática dos algoritmos, que para efeito de comparação foram executados 30 vezes para consolidar os resultados como proposto por Costa, Mei e Zhang (2020 e 2021).

O método de Hiper-Heurística Evolutiva (EHH) proposto por Costa, Mei e Zhang (2021) demonstrou superioridade clara, alcançando um *gap* médio de apenas 1,01% em relação às melhores soluções conhecidas (BKS). Em contraste, os resultados da implementação prática apresentaram um *gap* médio consideravelmente superior de 4,96%, evidenciando uma diferença de 3,95 pontos percentuais entre a teoria e a prática.

A comparação com o método Knowledge-Guided Local Search (KGLS) original revelou *gaps* médios de 2,87%, enquanto a versão adaptada KGLS obteve 2,31%. Estes resultados demonstram que, mesmo os métodos comparativos da literatura superam significativamente a implementação desenvolvida, indicando oportunidades substanciais de melhoria nos algoritmos implementados.

Tabela 3 - Comparação de Desempenho entre Métodos dos Artigos de Costa, Mei e Zhang (2020, 2021) e Implementação Prática

Método	Gap Médio (%)	Desvio Padrão (%)	Referência
EHH (Costa et al., 2021)	1,01	1,01	Costa, Mei e Zhang (2021)
KGLS Original	2,87	2,25	Costa, Mei e Zhang (2021)
KGLS Adaptado	2,31	1,65	Costa, Mei e Zhang (2021)
HH Adaptativo (Costa et al., 2020)	5,71	2,12	Costa, Mei e Zhang (2020)
Implementação Prática	4,96	2,44	Este trabalho

Fonte: Autoria própria baseado em Costa, Mei e Zhang (2020, 2021) e resultados experimentais

4.3. Análise de consistência e variabilidade

A análise da consistência dos métodos revelou que o EHH proposto por Costa, Mei e Zhang (2021) não apenas apresenta a melhor desempenho média, mas também demonstra maior estabilidade nos resultados. O desvio padrão de 1,01% para o método EHH contrasta com os 2,44% observados nos resultados implementados, indicando maior previsibilidade e robustez da abordagem teórica. Esta maior variabilidade na implementação sugere necessidade de refinamento nos mecanismos de controle e estabilização do algoritmo.

O método KGLS apresentou o maior desvio padrão (2,25%), enquanto KGLS* demonstrou variabilidade intermediária (1,65%), evidenciando que modificações na heurística de busca local podem afetar significativamente a consistência dos resultados, conforme observado por Costa, Mei e Zhang (2021). Por sua vez, o método de Hiper-Heurística adaptativa proposto por Costa, Mei e Zhang (2020) apresentou desvio padrão de 2,12%, demonstrando comportamento intermediário entre os métodos avaliados.

4.4. Análise das instâncias

As instâncias escolhidas foram:

- X-n200-k36:
 - Dimensão: 200 nós (199 clientes + 1 depósito)

- Veículos disponíveis: 36
- Capacidade por veículo: 402 unidades
- Faixa de demandas: 83-100 (variadas)
- Características específicas:
 - Instância de tamanho médio (199 clientes)
 - Capacidade moderada dos veículos (402 unidades)
 - Demandas heterogêneas entre clientes
 - Distribuição espacial euclidiana 2D
 - Coordenadas variando de 1 a 1000
 - Razão clientes/veículos: 5,5
- X-n251-k28:
 - Dimensão: 251 nós (250 clientes + 1 depósito)
 - Veículos disponíveis: 28
 - Capacidade por veículo: 69 unidades
 - Faixa de demandas: 5-10 (variadas)
 - Características específicas:
 - Instância de tamanho médio-grande (250 clientes)
 - Capacidade baixa dos veículos (69 unidades)
 - Demandas uniformemente pequenas (5-10)
 - Alta razão clientes/veículos: 8,9
 - Maior desafio de empacotamento
 - Coordenadas espalhadas (0-1000)
- X-n308-k13:
 - Dimensão: 308 nós (307 clientes + 1 depósito)
 - Veículos disponíveis: 13
 - Capacidade por veículo: 446 unidades
 - Faixa de demandas: 83-100 (variadas)
 - Características específicas:
 - Instância grande (307 clientes)
 - Poucos veículos disponíveis (13)
 - Capacidade alta por veículo (246)

- Razão clientes/veículos muito alta: 23,6
 - Demandas mínimas (1-10 unidades)
 - Maior desafio de roteamento
- X-n351-k40:
 - Dimensão: 351 nós (350 clientes + 1 depósito)
 - Veículos disponíveis: 48
 - Capacidade por veículo: 436 unidades
 - Faixa de demandas: 1-100 (variadas)
 - Características específicas:
 - Instância grande (350 clientes)
 - Frota média (40 veículos)
 - Capacidade alta (436 unidades)
 - Demandas muito heterogêneas (1-100)
 - Razão clientes/veículos: 8,75
 - Coordenadas centradas (0-1000)
- X-n376-k94:
 - Dimensão: 376 nós (375 clientes + 1 depósito)
 - Veículos disponíveis: 94
 - Capacidade por veículo: 4 unidades
 - Faixa de demandas: 1 (uniforme)
 - Características específicas:
 - Instância grande (375 clientes)
 - Muitos veículos pequenos (94)
 - Capacidade mínima (4 unidades)
 - Demandas uniformes (1 por cliente)
 - Razão clientes/veículos baixa: 4,0
 - Problema de múltiplas rotas curtas
- X-n401-k29:
 - Dimensão: 401 nós (400 clientes + 1 depósito)
 - Veículos disponíveis: 29
 - Capacidade por veículo: 745 unidades

- Faixa de demandas: 1-100 (variadas)
- Características específicas:
 - Instância muito grande (400 clientes)
 - Poucos veículos (29)
 - Capacidade muito alta (745)
 - Razão clientes/veículos alta: 13,8
 - Demandas variadas (1-100)
 - Desafio de otimização complexo
- X-n420-k130:
 - Dimensão: 420 nós (419 clientes + 1 depósito)
 - Veículos disponíveis: 130
 - Capacidade por veículo: 18 unidades
 - Faixa de demandas: 1-10 (pequenas)
 - Características específicas:
 - Instância muito grande (419 clientes)
 - Muitos veículos pequenos (130)
 - Capacidade baixa (18 unidades)
 - Razão clientes/veículos: 3,2
 - Demandas pequenas (1-10)
 - Múltiplas rotas curtas necessárias
- X-n449-k29:
 - Dimensão: 449 nós (448 clientes + 1 depósito)
 - Veículos disponíveis: 29
 - Capacidade por veículo: 777 unidades
 - Faixa de demandas: 1-100 (variadas)
 - Características específicas:
 - Maior instância testada (448 clientes)
 - Poucos veículos (29)
 - Capacidade máxima (777)
 - Razão clientes/veículos muito alta: 15,4
 - Maior variabilidade de demandas

- Máximo desafio computacional

Para a comparação direta entre métodos fosse possível para instâncias que aparecem tanto na literatura de Costa, Mei e Zhang (2021) quanto nos resultados experimentais:

Tabela 4 - Comparação Direta em Instâncias Sobrepostas entre Costa, Mei e Zhang (2020, 2021) e Implementação Prática

Instância	HH Adaptativo (Costa et al., 2020)	EHH (Costa et al., 2021)	BKS (Costa et al., 2021)	Implementação	Diferença EHH-Impl.
X-n200-k36	4,4%	1,37%	0,57%	4,45%	3,88 p.p.
X-n251-k28	6,1%	1,33%	0,20%	6,28%	6,08 p.p.
X-n308-k13	6,3%	3,77%	0,95%	9,58%	8,63 p.p.
X-n351-k40	8,1%	2,15%	0,29%	5,18%	4,89 p.p.
X-n376-k94	2,7%	0,38%	0,07%	1,09%	1,02 p.p.
X-n401-k29	4,2%	1,19%	0,28%	3,22%	2,80 p.p.
X-n420-k130	5,4%	1,24%	0,41%	5,32%	4,92 p.p.
X-n449-k29	7,5%	2,75%	0,51%	4,84%	4,33 p.p.

Fonte: Autoria própria baseado em Costa, Mei e Zhang (2020, 2021) e resultados experimentais

Para a instância X-n401-k29, o método EHH de Costa, Mei e Zhang (2021) alcançou *gap* de 0,47%, enquanto a implementação obteve 3,27%, o que representa uma defasagem de 2,80 pontos percentuais em relação ao método teórico. O método de Hiper-Heurística adaptativa de Costa, Mei e Zhang (2020) apresentou *gap* de 4,2% para esta instância. Na instância X-n420-k130, a disparidade foi ainda maior, com EHH

atingindo 0,40% contra 5,32% da implementação, uma diferença de 4,92 pontos percentuais, enquanto o método de 2020 obteve *gap* de 5,4%.

4.5. Análise de escalabilidade e eficiência computacional

4.5.1. Correlações identificadas

A análise de escalabilidade revelou correlação muito forte (0,986) entre o número de clientes e o tempo computacional, demonstrando crescimento quase linear nos recursos necessários. Esta característica é fundamental para aplicações práticas, pois indica previsibilidade no comportamento computacional do algoritmo, corroborando as observações de Costa, Mei e Zhang (2020) sobre a importância do controle adaptativo do espaço de busca. A eficiência média observada foi de 94,2 clientes processados por segundo, variando de 199 clientes/segundo para instâncias pequenas até 56 clientes/segundo para as maiores:

Tabela 5 - Características das Instâncias e Métricas de Eficiência

Categoria de Tamanho	Número de Instâncias	Gap Médio (%)	Tempo Médio (s)	Eficiência (clientes/s)
Pequenas (≤ 250)	3	5,89%	1,33	199
Médias (251-350)	2	7,44%	3,50	94
Grandes (> 350)	3	3,50%	7,67	56

Fonte: Autoria própria

4.5.2. Análise por categoria de tamanho

A classificação das instâncias por categoria de tamanho revelou padrões interessantes que corroboram as observações de Costa, Mei e Zhang (2020) sobre a necessidade de estratégias adaptativas para diferentes escalas de problema. Surpreendentemente, as instâncias de grande escala apresentaram melhor *gap* médio (3.50%) comparado às instâncias médias (7.44%), sugerindo que o algoritmo pode se beneficiar de estruturas mais complexas presentes em problemas maiores, fenômeno também observado por Costa, Mei e Zhang (2021) em suas análises do EHH.

4.6. Características detalhadas das instâncias

A análise das características específicas das instâncias testadas revelou variações significativas na complexidade e demandas computacionais. As instâncias variaram de 199 a 448 clientes, com números de rotas oscilando entre 13 e 137, refletindo diferentes

configurações de capacidade de veículos e distribuição geográfica, aspectos fundamentais considerados por Costa, Mei e Zhang (2020) em sua abordagem de busca adaptativa. Os custos das soluções obtidas variaram amplamente, desde 27.271,32 para X-n351-k40 até 149.330,34 para X-n376-k94, demonstrando a diversidade dos problemas testados.

O tempo computacional cresceu de forma consistente com o tamanho das instâncias, variando de 1 segundo para a menor instância até 8 segundos para a maior. Esta progressão linear confirma a escalabilidade adequada do algoritmo implementado, embora com eficiência decrescente em termos de clientes processados por unidade de tempo. Todas as soluções obtidas foram válidas, atendendo às restrições de capacidade e visitação, demonstrando a robustez fundamental da implementação baseada nas metodologias de Costa, Mei e Zhang (2021, 2020).

4.7. Limitações e oportunidades de melhoria

A análise comparativa identificou várias limitações na implementação atual que explicam as diferenças de desempenho observadas em relação aos trabalhos de Costa, Mei e Zhang (2021, 2020). O *gap* significativo entre teoria e prática (3,95 pontos percentuais para 2021 e 0,75 pontos percentuais para 2020) sugere necessidade de refinamentos em aspectos como calibração de parâmetros, implementação de mecanismos de intensificação e diversificação, e critérios de parada mais sofisticados, conforme sugerido por Costa, Mei e Zhang (2021). A maior variabilidade nos resultados implementados indica oportunidades para estabilização do algoritmo através de técnicas de controle adaptativos propostas por Costa, Mei e Zhang (2020).

As oportunidades de melhoria incluem integração com técnicas de aprendizado de máquina para seleção adaptativa de heurísticas, otimização computacional através de paralelização, e hibridização com outras meta-heurísticas de sucesso comprovado. A implementação de mecanismos de aprendizado online, conforme sugerido pela abordagem evolutiva de Costa, Mei e Zhang (2021), poderia potencialmente reduzir a lacuna observada em relação aos métodos da literatura. Adicionalmente, a investigação de diferentes estratégias de inicialização e critérios de aceitação, baseadas nas metodologias de Costa, Mei e Zhang (2020), poderia contribuir para melhorias na consistência dos resultados.

4.8. Implicações práticas e contribuições

Apesar das limitações identificadas, os resultados demonstram a viabilidade da aplicação de Hiper-Heurísticas ao Problema de Roteamento de Veículos de Larga Escala, validando as propostas teóricas de Costa, Mei e Zhang (2021, 2020). A implementação conseguiu processar instâncias de até 448 clientes mantendo tempo computacional aceitável e validação das soluções, evidenciando a aplicabilidade prática das metodologias propostas. A escalabilidade linear observada sugere adequação para cenários reais de logística e distribuição, corroborando as expectativas teóricas dos autores.

As contribuições principais incluem a validação experimental da abordagem de Hiper-Heurísticas em contexto prático, identificação de fatores críticos para sucesso da implementação baseada em Costa, Mei e Zhang (2021, 2020), e estabelecimento de *benchmarks* para futuras comparações. A análise detalhada das diferenças entre teoria e prática fornece diretrizes específicas para pesquisadores e profissionais interessados na aplicação destas metodologias. Os resultados também demonstram o potencial de melhoria através de refinamentos algorítmicos direcionados, seguindo as recomendações dos trabalhos originais.

4.9. Síntese e perspectivas futuras

A análise comparativa comprova a eficácia das Hiper-Heurísticas evolutivas conforme reportado por Costa, Mei e Zhang (2021), com o método EHH demonstrando superioridade consistente em qualidade e estabilidade. Similarmente, o método de Hiper-Heurística adaptativa de Costa, Mei e Zhang (2020) apresentou resultados competitivos para instâncias de menor escala. Os resultados implementados, embora com desempenho inferior, validam a abordagem fundamental proposta por Costa, Mei e Zhang (2020, 2021) e identificam caminhos claros para o aprimoramento. A forte correlação entre tamanho do problema e tempo computacional, combinada com robustez na qualidade das soluções, confirma a adequação da metodologia para aplicações de grande escala, conforme previsto pelos autores.

As perspectivas futuras incluem investigação de técnicas híbridas que combinam as vantagens das Hiper-Heurísticas propostas por Costa, Mei e Zhang (2021, 2020) com outros paradigmas de otimização, desenvolvimento de sistemas adaptativos que

aprendam automaticamente configurações ótimas, e aplicação em problemas reais de maior complexidade. A integração com tecnologias emergentes como aprendizado profundo e computação quântica representa direções promissoras para pesquisa futura. Os fundamentos estabelecidos pelas metodologias de Costa, Mei e Zhang (2021, 2020) e validados neste trabalho fornecem base sólida para estas investigações avançadas.

4.10. Comparação de parâmetros experimentais

A análise dos parâmetros experimentais utilizados nos diferentes estudos revela variações significativas nas configurações dos algoritmos genéticos e critérios de avaliação:

Tabela 6 - Comparação de Parâmetros Experimentais entre os Estudos

Parâmetro	Costa et al. (2021)	Costa et al. (2020)	Implementação
Tamanho da População	20	30	20/15
Número de Gerações	50	100	20/10
Taxa de <i>Crossover</i>	0,8	Toda geração	0,8
Taxa de Mutação	0,1	10%	0,1
Taxa de Elitismo	0,1	10%	25%
Número de Execuções	30	30	1
Tempo Limite por Indivíduo	30 segundos	Não especificado	300 segundos
Conjunto de Treinamento	21 instâncias (<200)	8 instâncias	-
Conjunto de Teste	79 instâncias	8 instâncias	8 instâncias

Fonte: Autoria própria baseado em Costa, Mei e Zhang (2020, 2021) e configurações da implementação

As diferenças nos parâmetros experimentais podem explicar parcialmente as discrepâncias observadas nos resultados. O estudo de Costa, Mei e Zhang (2021) utilizou um conjunto de treinamento mais extenso (21 instâncias) e maior número de execuções (30), o que pode ter contribuído para a melhor generalização do algoritmo. Por outro lado, Costa, Mei e Zhang (2020) utilizaram configurações mais conservadoras com maior número de gerações, mas conjunto de dados menor.

4.11. Diferenças metodológicas fundamentais

A análise das metodologias revela diferenças fundamentais entre as abordagens propostas por Costa, Mei e Zhang (2020 e 2021). O trabalho de 2020 foca em Hiper-Heurísticas de seleção perturbativa com cromossomos bidimensionais que controlam a sequência de operadores e os limites de busca adaptativos. Esta abordagem demonstrou eficácia particular em instâncias de menor escala, com *gaps* médios variando entre 4,4% e 12,1% em relação ao BKS.

Em contraste, o estudo de 2021 propõe uma Hiper-Heurística evolutiva mais complexa que integra *Knowledge-Guided Local Search* (KGLS) com cromossomos multidimensionais. Esta metodologia incorpora três componentes principais: inicialização adaptativa, sequência de vizinhanças otimizada e funções de penalização evolutivas. Os resultados superiores (*gap* médio de 1,01%) demonstram a eficácia desta abordagem mais sofisticada.

A implementação prática adotou uma abordagem híbrida, que combinou elementos de ambas as metodologias: uma fase inicial de seleção perturbativa seguida por uma fase evolutiva com KGLS. Embora esta hibridização tenha resultado em soluções válidas para todas as instâncias testadas, o *gap* médio de 4,96% indica necessidade de refinamentos na integração das metodologias.

5. CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma análise abrangente das Hiper-Heurísticas aplicadas ao Problema de Roteamento de Veículos de Grande Escala (PRVGE), explorando tanto os aspectos teóricos quanto as implementações práticas dessas metodologias avançadas de otimização. A pesquisa desenvolvida permitiu investigar as principais contribuições científicas na área, implementar algoritmos baseados nas metodologias propostas por Costa, Mei e Zhang (2020, 2021), e avaliar empiricamente sua eficácia em instâncias de teste reconhecidas pela comunidade científica.

Os resultados obtidos demonstram que as Hiper-Heurísticas representam uma evolução significativa nas abordagens de otimização para problemas combinatórios complexos. A análise comparativa revelou que, apesar de a implementação prática tenha apresentado *gaps* superiores aos reportados na literatura (4,96% contra 1,01% do método EHH), os algoritmos desenvolvidos conseguiram processar eficientemente instâncias com até 448 clientes, mantendo tempo computacional linear e produzindo soluções válidas para todos os casos testados.

A investigação das diferentes categorias de Hiper-Heurísticas evidenciou que as abordagens evolutivas apresentam vantagens substanciais em termos de qualidade das soluções, enquanto as Hiper-Heurísticas de seleção demonstram maior simplicidade de implementação e menor custo computacional. A forte correlação identificada entre o tamanho das instâncias e o tempo computacional (0,986) confirma a escalabilidade adequada da abordagem para aplicações práticas de logística e distribuição.

Quanto aos objetivos específicos estabelecidos, a revisão do estado da arte permitiu identificar as principais tendências e lacunas na aplicação de Hiper-Heurísticas ao PRVGE, fornecendo subsídios para a publicação de artigos de revisão na área. A implementação dos algoritmos documentou efetivamente os mecanismos de automação da tomada de decisão heurística, evidenciando o potencial dessas metodologias para reduzir a dependência de conhecimento especializado de domínio.

A análise das possibilidades de resolver problemas mais complexos relacionados ao PRVGE revelou que as Hiper-Heurísticas oferecem flexibilidade superior às meta-heurísticas tradicionais, especialmente em cenários com alta variabilidade nas características das instâncias. Os resultados obtidos sugerem que instâncias de grande

escala podem, paradoxalmente, se beneficiar mais dessas abordagens devido à maior complexidade estrutural que permite exploração mais eficiente do espaço de busca.

As limitações identificadas incluem a necessidade de refinamento na calibração de parâmetros, implementação de mecanismos mais sofisticados de intensificação e diversificação, e desenvolvimento de critérios de parada adaptativos. A maior variabilidade nos resultados implementados, comparada aos métodos da literatura, indica oportunidades para estabilização através de técnicas de controle adaptativo.

As contribuições principais desta pesquisa incluem: (i) validação experimental das metodologias de Hiper-Heurísticas em contexto prático; (ii) identificação de fatores críticos para o sucesso da implementação; (iii) estabelecimento de *benchmarks* para futuras comparações; (iv) análise detalhada das diferenças entre teoria e prática; e (v) diretrizes específicas para pesquisadores interessados na aplicação dessas metodologias.

Para trabalhos futuros, recomenda-se a investigação de técnicas híbridas que combinam as vantagens das diferentes categorias de Hiper-Heurísticas, o desenvolvimento de sistemas adaptativos com aprendizado automático de configurações ótimas, e a aplicação em problemas reais de maior complexidade. A integração com tecnologias emergentes como aprendizado profundo e computação paralela representa direções promissoras para pesquisas avançadas.

Finalmente, este estudo demonstra que as Hiper-Heurísticas constituem uma ferramenta valiosa para o avanço da pesquisa operacional moderna, oferecendo soluções robustas e adaptáveis para os desafios logísticos contemporâneos. Os fundamentos estabelecidos nesta pesquisa contribuem para o desenvolvimento de sistemas de otimização mais inteligentes e autônomos, com potencial de aplicação em diversos domínios além do roteamento de veículos.

REFERÊNCIAS

- BORGES, Jucelina Leandro; DANTAS, Maria José Pereira; SILVA, Eliel Lucas de Oliveira. FERRAMENTA FERRAMENTA COMPUTACIONAL EM EXCEL: uma proposta de metodologia para seleção de artigos científicos relevantes. **Open Science Research X**, [S.L.], p. 1024-1038, 2023. Editora Científica Digital. <http://doi.org/10.37885/230111849>.
- CHEN, Xinan; BAI, Ruibin; QU, Rong; DONG, Jing; JIN, Yaochu. Deep Reinforcement Learning Assisted Genetic Programming Ensemble Hyper-Heuristics for Dynamic Scheduling of Container Port Trucks. **Ieee Transactions On Evolutionary Computation**, [S.L.], p. 1-1, 2024. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/tevc.2024.3381042>.
- COSTA, Joao Guilherme Cavalcanti; MEI, Yi; ZHANG, Mengjie. Adaptive Search Space through Evolutionary Hyper-Heuristics for the Large-Scale Vehicle Routing Problem. **2020 Ieee Symposium Series On Computational Intelligence (Ssci)**, [S.L.], p. 2415-2422, 1 dez. 2020. IEEE. <http://dx.doi.org/10.1109/ssci47803.2020.9308239>.
- COSTA, Joao Guilherme Cavalcanti; MEI, Yi; ZHANG, Mengjie. An Evolutionary Hyper-Heuristic Approach to the Large Scale Vehicle Routing Problem. **2021 Ieee Congress On Evolutionary Computation (Cec)**, [S.L.], p. 2109-2116, 28 jun. 2021. IEEE. <http://dx.doi.org/10.1109/cec45853.2021.9504818>.
- COSTA, Joao Guilherme Cavalcanti; MEI, Yi; ZHANG, Mengjie. Cluster-based Hyper-Heuristic for Large-Scale Vehicle Routing Problem. **2020 Ieee Congress On Evolutionary Computation (Cec)**, [S.L.], p. 1-8, jul. 2020. IEEE. <http://dx.doi.org/10.1109/cec48606.2020.9185831>.
- COSTA, P. M.; SILVA, J. A. Solving real urban VRPTW instances by applying a branch-cut-and-price via VRPsolver. **IEEE Latin America Transactions**, [s. l.], v. 20, n. 10, p. 2178-2187, 2022. DOI: 10.1109/TLA.2022.9878539.
- JOHNSON, A. B.; SMITH, C. D. The effect of credit definition and aggregation strategies on multi-objective hyper-heuristics. In: **PROCEEDINGS OF ASME INTERNATIONAL DESIGN ENGINEERING TECHNICAL CONFERENCES**, 2015, Boston. Anais... Boston: ASME, 2015. p. 251-341.
- DOKEROGLU, Tansel; KUCUKYILMAZ, Tayfun; TALBI, El-Ghazali. Hyper-heuristics: a survey and taxonomy. **Computers & Industrial Engineering**, [S.L.], v. 187, p. 109815, jan. 2024. Elsevier BV. <http://dx.doi.org/10.1016/j.cie.2023.109815>.

MAASHI, Mashael; KENDALL, Graham; ÖZCAN, Ender. Choice function based hyper-heuristics for multi-objective optimization. *Applied Soft Computing*, [S.L.], v. 28, p. 312-326, mar. 2015. **Elsevier BV**. <http://dx.doi.org/10.1016/j.asoc.2014.12.012>.

PILLAY, Nelishia; QU, Rong. **Hyper-Heuristics: Theory and Applications**. Suíça: Springer, 2018. 134 p. Disponível em: <https://link.springer.com/book/10.1007/978-3-319-96514-7>.

SABAR, Nasser R.; ZHANG, Xiuzhen Jenny; SONG, Andy. A math-hyper-heuristic approach for large-scale vehicle routing problems with time windows. **2015 IEEE Congress On Evolutionary Computation (Cec)**, [S.L.], p. 830-837, maio 2015. IEEE. <http://dx.doi.org/10.1109/cec.2015.7256977>.

SANTOS, R. B.; OLIVEIRA, J. P. **Algoritmo híbrido de busca local guiada para o problema de roteamento de veículos capacitados**. *Observatório Latino-Americano*, [s. l.], v. 12, n. 3, p. 112-134, 2023.

SILVA, M. A. **Otimização de problemas de roteamento de veículos: soluções heurísticas**. *Revista de Ciência e Qualidade*, [s. l.], v. 18, n. 5, p. 45-68, 2020.

SOUZA, Daniel Vieira de. **UMA ABORDAGEM POR HIPERHEURÍSTICA COM APRENDIZADO PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM JANELA DE TEMPO**. 2019. 78 f. Dissertação (Mestrado) - Curso de Programa de Pós-Graduação em Ciência da Computação, Ufersa, Mossoró, 2019. Disponível em: <https://repositorio.ufersa.edu.br/handle/prefix/5216>.

APÊNDICE A - Resultado da Revisão de Literatura

Referência	Título	Contexto	Abordagem	Pontos-Chave	Relevância
(M. Pour; Drake; Burke, 2018)	A choice function hyper-heuristic framework for the allocation of maintenance tasks in Danish railways	Alocação de tarefas de manutenção no sistema ferroviário dinamarquês	Hiper-heurística de seleção baseada em 'choice function' para escolher heurísticas de forma dinâmica	- Critério de balanceamento de carga - Minimização de distância entre tarefas em sub-regiões - Teste em 12 instâncias	Demonstra hiper-heurísticas de seleção para escalonamento e manutenção
(Zhang et al., 2022)	A deep reinforcement learning based hyper-heuristic for combinatorial optimisation with uncertainties	Uso de DRL como "high-level strategy" para problemas de incerteza (ex. roteamento em terminais de contêiner, empacotamento 2D online)	Framework de hiper-heurística em que o módulo DRL escolhe heurísticas de busca local considerando parâmetros de incerteza	- Integração de DRL e heurísticas de baixo nível - Resultados superiores em cenários com incerteza	Ilustra tendência de hiper-heurísticas hibridizadas com aprendizado de máquina
(Sabar et al., 2015a)	A Dynamic Multiarmed Bandit-Gene Expression Programming Hyper-Heuristic for Combinatorial Optimization	Seleção baseada em multi-armed bandit dinâmico e programação genética para aceitação de soluções	Hiper-heurística de seleção em alto nível (MAB) e aceitação de soluções via critérios construídos por Gene Expression Programming	- Troca de heurísticas "on the fly" - GEP gera critério de aceitação - Aplicações em exam timetabling e routing dinâmico	Mostra como aprendizado pode melhorar adaptabilidade das hiper-heurísticas
(Garrido; Castro, 2012)	A Flexible and Adaptive Hyper-heuristic Approach for (Dynamic) Capacitated Vehicle Routing Problems	VRPs com capacidade, em versões estática e dinâmica	Hiper-heurística auto-adaptativa (hill-climbing com reforço) orquestrando heurísticas construtivas e perturbativas	- Colaboração de heurísticas de construção/otimização - Adaptação de parâmetros via "reward/penalty"	Mostra estabilidade e boa qualidade de soluções em VRPs, inclusive dinâmicos

(Laboni et al., 2024)	A Hyper Heuristic Algorithm for Efficient Resource Allocation in 5G Mobile Edge Clouds	Alocação de recursos em 5G Edge Cloud, envolvendo latência e balanceamento de carga	Framework de hiper-heurística que combina Ant Colony, Whale, Sine-Cosine e Henry Gas Solubility, selecionados dinamicamente	- Multiobjetivo (latência, carga, balanceamento de rede) - Resultados mostram melhor relação latência × offloading	Exemplo da versatilidade de hiper-heurísticas em redes emergentes (5G, cloud/fog)
(Olgun; Koç; Altıparmak, 2021)	A hyper heuristic for the green vehicle routing problem with simultaneous pickup and delivery 2024	VRP “verde” com coleta e entrega simultânea, minimizando consumo de combustível	Hiper-heurística baseada em ILS e VND para reduzir custo de combustível	- Considera peso do veículo no consumo - Compara VRPSPD “clássico” × versão “verde”	Foco em critérios ambientais (emissões) em logística e uso de HH para lidar com a complexidade
(Zhang; Zhao; Leng, 2020)	A Hyper-Heuristic Algorithm for Time-Dependent Green Location Routing Problem With Time Windows	Problema de localização e roteamento “verde” com janelas de tempo e velocidade variável ao longo do dia	Hiper-heurística com tabu search em alto nível e heurísticas gulosas de aceitação	- Minimiza custo total (depósitos, veículos, combustível) - Velocidades e tempos de viagem variam conforme o horário	Integra decisões de localização e roteamento em contexto “verde” com variação temporal
(Cheng et al., 2022)	A hyper-heuristic approach to the strategic planning of bike-sharing infrastructure	Planejamento de estações e ciclovias em sistemas de bike-sharing	Bi-level model (nível superior: localização de estações e ciclovias; nível inferior: rotas dos usuários) + HH sequencial com Hidden Markov	- Elasticidade de demanda (mais infraestrutura, mais usuários) - Minimiza custo de construção e tempo de viagem	Amplia o uso de HH para planejamento de transporte multimodal, incluindo variáveis de infraestrutura
(Pandiri; Singh, 2018)	A hyper-heuristic based artificial bee colony algorithm for k-	k-IMDMTSP: TSP multi-depósito com todos os depósitos interligados em um ciclo	Combinação de hiper-heurística com artificial bee colony (ABC), gerenciando a aplicação de heurísticas no	- Novo esquema de codificação reduz espaço de busca - Comparações com BRKGA e	Eficácia de HH em roteamento com múltiplos vendedores e

	Interconnect ed multi-depot multi-traveling salesman problem	interno	encoding	multi-start	interconexão de depósitos
(Chen et al., 2018)	A hyper-heuristic with two guidance indicators for bi-objective mixed-shift vehicle routing problem with time windows	Roteamento de veículos em múltiplos turnos, com janelas de tempo e dois objetivos (custo motorista e distância)	Hiper-heurística de seleção com heurísticas de perturbação de diferentes intensidades e dois indicadores de orientação	- Critério de aceitação hill-climbing - Dois indicadores para guiar busca no espaço Pareto - Aplicações em transporte de contêineres	Demonstra HH bi-objetivo com guias específicos para cada objetivo em VRP de múltiplos turnos
(Soria-Alcaraz et al., 2017)	A methodology for determining an effective subset of heuristics in selection hyper-heuristics	Discussão sobre composição do pool de heurísticas em hiper-heurísticas de seleção	Metodologia estatística para reduzir conjunto de heurísticas usando testes não-paramétricos e análise de <i>fitness landscape</i>	- Novo critério de seleção (multi-armed bandits e change detection) - Aplicado em timetabling e VRP, com bons resultados	Reforça a importância de escolher heurísticas adequadas para melhorar desempenho e reduzir tempo
(Hamid; Nasiri; Rabbani, 2023)	A mixed closed-open multi-depot routing and scheduling problem for homemade meal delivery incorporating drone and crowd-sourced fleet: A self-adaptive hyper-heuristic approach	Distribuição de refeições caseiras, usando veículos próprios, drones e frota “crowdsourced” em ambiente multi-depósito	Hiper-heurística auto-adaptável baseada em GA/PSO modificados	- Objetivos múltiplos: custo, frescor, prazos - Redução de 8–20% nos indicadores de custo e pontualidade	Mostra adoção de HH em logística multimodal complexa, incluindo drones

<u>(Zhao; Leng; Zhang, 2021)</u>	A novel framework of hyper-heuristic approach and its application in location-routing problem with simultaneous pickup and delivery	Location-Routing Problem com coleta e entrega simultânea (LRPSPD)	HH baseada em Iterated Local Search (ILS-HH), distinguindo heurísticas de “mutação” e “hill climbers”	- Testa estratégias de seleção (Random, Tabu, MAB etc.) - Melhora fuga de ótimos locais	Framework genérico e competitivo em LRPSPD
<u>(Leng et al., 2019)</u>	A Novel Hyper-Heuristic for the Biobjective Regional Low-Carbon Location-Routing Problem With Multiple Constraints	LRP regional “low-carbon” com pickup-delivery, janelas de tempo e frota heterogênea	Hiper-heurística multiobjetivo com seleção “quantum-inspired” e critérios de aceitação ambiental	- Minimiza custos e emissões de CO₂ - Resultados que superam meta-heurísticas dedicadas	Evidencia a importância de sustentabilidade (carbon emission) em roteamento de localização
<u>(Qin et al., 2021)</u>	A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem	VRP heterogêneo com vários tipos de veículos, objetivo de minimizar tempo máximo de rota	Hiper-heurística que emprega RL para seleção de heurísticas, integrando deep learning para guiar escolhas	- Melhor desempenho em cenários de grande escala - MILP para instâncias pequenas	Demonstra expansão de HH para RL policy-based em VRP heterogêneos
<u>(Li et al., 2024a)</u>	A reinforcement learning-based hyper-heuristic for AGV task assignment and route planning in parts-to-picker warehouses	Planejamento de rotas e atribuição de tarefas para AGVs em armazéns “parts-to-picker”	HH com multi-armed bandit aprimorado (Co-SLMAB) e Exponential Monte Carlo (EMCQ) como critério de aceitação	- Evita colisões (conflitos de rota) - Sensibilidade mostra benefícios de conjunto diverso de heurísticas	Aplicável em intralogística complexa com AGVs

<u>(Pandiri; Singh, 2021)</u>	A simple hyper-heuristic approach for a variant of many-to-many hub location-routing problem	Many-to-Many p-Location-Hamiltonian Cycle Problem (MMpLHP)	HH multi-start com três heurísticas de baixo nível (<i>subset selection, grouping, permutation</i>)	- Resultados superam BRKGA, multi-VND - Foco em hub location e roteamento híbrido	Resolve simultaneamente problemas de localização e roteamento (hub)
<u>(Sanchez et al., 2020)</u>	A Systematic Review of Hyper-Heuristics on Combinatorial Optimization Problems	Revisão sistemática de hiper-heurísticas em problemas de otimização combinatória	Levantamento dos principais problemas estudados e identificação de lacunas (ex.: shortest path, set cover)	- Aumento de estudos em VRP e escalonamento - Oportunidades de pesquisa em domínios menos explorados	Fornecer panorama amplo e orientar futuras pesquisas em hiper-heurísticas
<u>(Sabar; Zhang; Song, 2015)</u>	A math-hyper-heuristic approach for large-scale vehicle routing problems with time windows	VRPTW de grande porte (até 1000 clientes)	Combina programação matemática (<i>column generation</i>) e hiper-heurística para melhorar soluções	- Resultados superiores a métodos de referência - Bom equilíbrio entre qualidade e tempo computacional	Mostra eficácia de integrar métodos exatos e HH para problemas gigantes
<u>(Li et al., 2024b)</u>	Active learning based hyper-heuristic for the integration of production and Transportation: A third-party logistics perspective	Integração produção e transporte (IPPT) com 3PL, escalonamento o “no-wait flow-shop” distribuído e roteamento de veículos	HH baseada em active learning (ALHHA), combinando busca em vizinhança e exploração em alto nível	- Melhora significativa em custo total (produção + transporte) - Até 200 jobs e 7 fábricas	Destaca uso de aprendizado ativo para guiar seleção de heurísticas em integração produção-transporte
<u>(Costa; Mei; Zhang, 2020b)</u>	Adaptive Search Space through Evolutionary	LSVRP (VRP em grande escala), ajustando dinamicamente	HH evolutiva (baseada em GA) que ajusta o tamanho de vizinhança	- Redução da dependência de conhecimento humano - Desempenho	Introduz “search space pruning adaptativo”

	Hyper-Heuristics for the Large-Scale Vehicle Routing Problem	e o espaço de busca	(pruning) e heurísticas de busca local	melhor em instâncias >200 clientes	para maior escalabilidade
<u>(Soleimanian Gharehchopogh; Abdollahzadeh; Arasteh, 2023)</u>	An Improved Farmland Fertility Algorithm with Hyper-Heuristic Approach for Solving Travelling Salesman Problem	TSP (problema clássico NP-hard)	Combinação do Farmland Fertility Algorithm (FFA) e uma hiper-heurística com Modified Choice Function, refinando com Lin-Kernighan	- Evita estagnação no FFA - Testado em 71 instâncias TSPLIB - Boa qualidade de rota e tempo de execução	Exemplo de como HH potencializa meta-heurísticas inspiradas na natureza para TSP
<u>(Rodriguez; Oteiza; Brignole, 2019)</u>	An urban transportation problem solved by parallel programming with hyper-heuristics	Roteamento de ônibus em ambiente urbano	Combinação paralela de meta-heurísticas (SA, GA, Ant Colony) estruturadas como hiper-heurística	- Aceleração em execução paralela - Melhoria na fuga de ótimos locais	Salienta valor de HH em transporte urbano, associada a paralelismo e cooperação entre meta-heurísticas
<u>(Cavalcanti Costa; Mei; Zhang, 2021)</u>	An Evolutionary Hyper-Heuristic Approach to the Large Scale Vehicle Routing Problem	LSVRP com mais de 200 clientes	GA para evoluir sequência de heurísticas (local search, construction, escape) e controlar parâmetros de pruning	- Baseado em Guided Local Search - Evolui seleção de heurísticas e limites de vizinhança	Reforça eficiência de HH evolutiva em VRPs de grande escala, similar ao artigo 2.22
<u>(Butterwick et al., 2023)</u>	Application of selection hyper-heuristics to the simultaneous optimisation of turbines	Projeto de windfarm offshore: localização de turbinas e roteamento de cabos	HH de seleção para heurísticas de perturbação em turbinas e cabling; comparação simultânea vs sequencial	- Foco em reduzir custo de cabeamento e aumentar geração - HH com diferentes critérios de	Mostra aplicação de HH para solucionar simultaneamente subproblemas de energia

	and cabling within an offshore windfarm			aceitação	e infraestrutura
<u>(Costa; Mei; Zhang, 2020a)</u>	Cluster-based Hyper-Heuristic for Large-Scale Vehicle Routing Problem	LSVRP usando clusterização para restringir subespaços de busca	HH de seleção que agrupa clientes dinamicamente e aplica heurísticas em cada cluster; clusters se adaptam durante a busca	- Decodificação baseada em cluster-first - Escalabilidade melhorada	Demonstra eficácia do approach de clusterização adaptativa em VRP de grande escala
<u>(Zhao et al., 2020)</u>	Evolutionary Hyperheuristics for Location-Routing Problem with Simultaneous Pickup and Deliver	CLRP e CLRPSPD (coleta e entrega simultânea), problemas NP-hard	HH evolutivas (Quantum, Ant, Particle) no alto nível + critérios de aceitação (All Moves, Naïve, Great Deluge, SA)	- 36 combinações (9 estratégias × 4 aceitações) - Melhora resultados em benchmarks	Orquestração complexa de meta-heurísticas sob nível HH com excelentes resultados
<u>(Tyasnurita et al., 2024)</u>	Constructing selection hyper-heuristics for open vehicle routing with time delay neural networks using multiple experts	OVRP (Open VRP) em logística terceirizada, sem retorno à base	Redes neurais com atraso temporal (time delay NN) para aprendizado de imitação de heurísticas humanas, formando HH de seleção	- Seleciona heurísticas de perturbação e critério de aceitação - Melhora sobre experts humanos isolados	Exemplo de uso de NN para projetar hiper-heurísticas em roteamento aberto
<u>(Leng et al., 2020)</u>	Decomposition-based hyperheuristic approaches for the bi-objective cold chain considering environmental effects	Cadeia fria (cold chain), roteamento + alocação de depósitos, objetivos: custo e qualidade/frescor	Hiper-heurística multiobjetivo com decomposição (MOEA/D), testando três estratégias de seleção (RS, choice function, FRR-MAB)	- Enfatiza deterioração de produtos refrigerados e emissões - Comparações com MOEAs consolidados	Integra aspectos ambientais e qualidade (frescura) em LRP multiobjetivo, validando robustez de HH

<u>(Garrido; Riff, 2010)</u>	DVRP: a hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic	DVRP (Dynamic Vehicle Routing Problem), clientes surgem ao longo do tempo	Hiper-heurística evolutiva (EH-DVRP) gerenciando heurísticas construtivas, perturbativas e “noise”	- Adaptação incremental à dinâmica - Competitividade em benchmarks (Kilby, Christofides etc.)	Mostra adaptabilidade de HH em cenários de roteamento dinâmico, com bons resultados
<u>(MacLachlan, [S.d.])</u>	Genetic Programming Hyper-Heuristics with Vehicle Collaboration for Uncertain Capacitated Arc Routing Problems	UCARP (Uncertain Capacitated Arc Routing Problem), demanda incerta em rotas de coleta/entrega em arestas	Uso de Genetic Programming para construir políticas de roteamento “online” + colaboração entre veículos	- Colaboração para exceder capacidade - Resultados notáveis em grandes instâncias	Eficácia de evoluir políticas de roteamento adaptativo e colaboração de veículos em cenários incertos
<u>(Sabar et al., 2013)</u>	Grammatical Evolution Hyper-Heuristic for Combinatorial Optimization Problems	Uso de Grammatical Evolution (GE) para construir hiper-heurísticas em exam timetabling e CVRP	GE atua como “solver builder” combinando heurísticas de aceitação, estruturas de vizinhança etc.	- Melhora adaptabilidade - Competitivo com métodos sob medida - Aplicado a timetabling e VRP	Primeiro estudo a mostrar o potencial de GE em HH para diversos domínios
<u>(Shang; Ma; Liu, 2023)</u>	Green location routing problem with flexible multi-compartment for source-separated waste: A Q-learning and multi-strategy-based hyper-heuristic algorithm	LRP para coleta de resíduos pré-separados, veículos de múltiplos compartimentos flexíveis e emissões de carbono	HH multi-estratégia com Q-learning para escolher heurísticas em diferentes estágios	- Modelo inédito de compartimentos flexíveis - Aprendizado por reforço para seleção de heurísticas	Exemplo de HH baseada em Q-learning em problemas verdes e multi-compartimentos

<u>(Wang; Liu; Zhang, 2023)</u>	Hyper-heuristic algorithm for traffic flow-based vehicle routing problem with simultaneous delivery and pickup	VRP simultâneo (entrega + coleta) em ambiente urbano, considerando fluxo de tráfego	HH de seleção com tabu search em alto nível, heurísticas de busca local e perturbação, integrando modelos de congestionamento	- Modelagem de velocidade-fluxo - Aumento de satisfação ao incluir congestionamento urbano	Destaca integração de dados de tráfego em VRP simultâneo para cenários urbanos
<u>(Jafari Nozar; Behnamian, 2021)</u>	Hyper-heuristic for integrated due-window scheduling and vehicle routing problem for perishable products considering production quality	Problema integrado de scheduling (flexible flowshop) e roteamento de entregas de perecíveis com janelas de entrega	HH de seleção com critério de aceitação Monte Carlo, equilibrando custos e frescor dos produtos	- Equilíbrio entre produção e distribuição - Comparado com NSGA-II, apresenta melhores resultados	Mostra força de HH em problemas “end-to-end” (fábrica + entrega), especialmente perecíveis
<u>(Hu et al., 2023)</u>	Hyper-Heuristic Algorithm for Urban Traffic Flow Optimization	Otimização de fluxo de tráfego em rede urbana, tempos dinâmicos	Uso de Genetic Programming para evoluir estratégias HH de rotas em tempo real	- Redução de tempos médios de viagem - Maior fluidez em rede urbana - Testes em rede pequena e aplicação em maior escala	Exemplo de GP-HH no “traffic assignment problem”, controlando rotas dinamicamente
<u>(Aroba; Naicker; Adeliyi, 2021)</u>	A Hyper-Heuristic Heterogeneous Multisensor Node Scheme for Energy Efficiency in Larger Wireless Sensor Networks Using DEEC-	WSN com sensores heterogêneos, buscando eficiência energética e maior vida útil da rede	Combinação de DEEC (Distributed Energy-Efficient Clustering) com método Gaussiano, em estrutura de hiper-heurística	- Otimização de cluster heads - Comparações em simulações com até 1500 nós	Demonstra aplicação de HH em WSN para gestão de energia e prolongar vida de nós

	Gaussian Algorithm				
<u>(Lagos; Pereira, 2024)</u>	Multi-armed bandit-based hyper-heuristics for combinatorial optimization problems	Uso de MAB (multi-armed bandit) em hiper-heurísticas, aplicado em VRPTW	Escolha de heurísticas como problema MAB, usando algoritmos como EXP3, UCB etc.; transições modeladas em nível de HH	- <i>Gap</i> ~2.16% frente ao best-known em VRPTW - Flexibilidade para muitas heurísticas	Primeiro trabalho com EXP3 em HH, mostra potencial em problemas grandes ou incertos
<u>(Kheiri et al., 2020)</u>	Exact and hyper-heuristic solutions for the distribution-installation problem from the VeRoLog 2019 challenge	Problema de distribuição-instalação (VeRoLog 2019), combinando CVRPTW e scheduling de técnicos com skills	Modelo matemático para instâncias pequenas + HH populacional para grandes instâncias	- Minimiza custos de transporte + técnicos - Comparação com finalistas VeRoLog, resultados competitivos	Exemplo de HH robusta em problemas reais de roteamento e escalonamento, destacando alta competitividade
<u>(Yao; Peng; Xiao, 2018)</u>	Parallel Hyper-Heuristic Algorithm for Multi-Objective Route Planning in a Smart City	Roteamento multiobjetivo em cidade inteligente (distância, tempo, segurança)	HH multiobjetivo com heurísticas de seleção/aceitação e RL, implementada em GPU (CUDA) para paralelismo	- Cenários de grande escala (ex. Nova Iorque) - Aceleração de até 173×	Mostra eficácia de HH paralelo em roteamento urbano com múltiplos objetivos
<u>(Udomkasemsu b; Sirinaovakul; Achalakul, 2023)</u>	PHH: Policy-Based Hyper-Heuristic With Reinforcement Learning	Uso de RL para guiar a política de seleção de heurísticas em problemas como TSP, CVRP, Bin packing	HH “policy-based” que treina política de seleção em instâncias de porte médio e generaliza para grandes instâncias	- Desempenho superior ou igual em diversos domínios - Aplicação em workflow scheduling no cloud	Confirma a força de RL integrado a HH para alta generalidade e desempenho

<u>(Sabar; Kendall, 2015)</u>	<i>Population</i> based Monte Carlo tree search hyper-heuristic for combinatorial optimization problems	Framework de HH usando Monte Carlo Tree Search (MCTS)	Modela espaço de heurísticas como árvore; MCTS orienta escolha e usa “ <i>population</i> -based memory” para manter diversidade	- Testes em 6 domínios do HyFlex (SAT, bin packing, flow shop, etc.) - Resultados competitivos e, às vezes, best-known	Inovador no uso de MCTS como mecanismo de seleção, adaptável a múltiplos domínios
<u>(Ahmed; Mumford; Kheiri, 2019)</u>	Solving urban transit route design problem using selection hyper-heuristics 2024	Planejamento de rotas de ônibus em rede urbana, minimizando tempo de passageiros e custo operacional	Várias HH de seleção + heurísticas de aceitação (Choice Function, Tabu, Great Deluge) e heurísticas construtivas/melhoradoras específicas	- Em benchmarks de transit networks, “sequence-based + Great Deluge” se destacou - Supera GA e heurísticas convencionais	Viabilidade de HH para UTRP (planejamento de transporte público), com melhor desempenho
<u>(Tarhini; Danach; Harfouche, 2022)</u>	Swarm intelligence-based hyper-heuristic for the vehicle routing problem with prioritized customers 2024	VRP com clientes priorizados (VIP ou deadlines críticos)	Combina “Cuckoo Search” e “hyper-heuristic” para orquestrar heurísticas construtivas/melhoradoras, incluindo prioridades	- Modificação do algoritmo Clarke-Wright - Melhor ao respeitar prioridades de clientes	Demonstra HH com enxame (swarm) como alto nível, eficiente em VRP priorizado
<u>(Sabar et al., 2015b)</u>	The Automatic Design of Hyper-heuristic Framework with Gene Expression Programming for Combinatorial Optimization problems	Uso de Gene Expression Programming (GEP) para projetar dinamicamente e seleção e aceitação de heurísticas	GEP evolui mecanismos de escolha de LLHs e critério de aceitação, baseando-se no estado (melhora, qualidade)	- Testado em 6 domínios HyFlex - Conseguiu novas best solutions em algumas instâncias	Sublinha a força do GEP para automatizar controle de HH, gerando alto desempenho

<u>(Shang et al., 2022)</u>	The sorted-waste capacitated location routing problem with queuing time: A cross-entropy and simulated-annealing-based hyper-heuristic algorithm	LRP com coleta de lixo segregado e tempo de fila (queuing) em instalações, aplicado ao contexto chinês	Hiper-heurística combinando cross-entropy e simulated annealing, com heurísticas específicas de codificação/decodificação	- Modelo CLRPQT (fila de atendimento) - Supera heurísticas clássicas em benchmarks e instâncias reais de Shanghai	Lida com restrições de tempo de fila e coleta seletiva, evidenciando a robustez do método HH
<u>(Liao et al., 2024)</u>	Uncertain Commuters Assignment Through Genetic Programming Hyper-Heuristic	Atribuição de passageiros incertos em redes de transporte (traffic assignment problem com demanda imprevisível)	GP-based HH para gerar regras de roteamento reativas (RAS), cada agente decide caminho baseando-se em heurísticas evoluídas	- Maximiza throughput e reduz congestionamento - Boa generalização em cenários não vistos	Demonstra GP-HH em roteamento dinâmico com demanda incerta, útil para gerência de tráfego e mobilidade urbana

Fonte: Autoria Própria

APÊNDICE B - Resultados Completo dos Testes do Algoritmo Implementado

Instância	Clientes	BKS	Custo	Gap (%)
X-n200-k36	199	58,578	61,186.92	4.45
X-n251-k28	250	38,684	41,112.84	6.28
X-n308-k13	307	25,859	28,337.00	9.58
X-n351-k40	350	25,928	27,271.32	5.18
X-n376-k94	375	147,713	149,330.34	1.09
X-n401-k29	400	66,187	68,317.37	3.22
X-n420-k130	419	107,798	113,531.87	5.32
X-n449-k29	448	55,269	57,944.81	4.84

Fonte: Autoria Própria