

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



**META-HEURÍSTICA HÍBRIDA PARA O PROBLEMA
DE ROTEAMENTO DE VEÍCULOS COM COLETA E
ENTREGA E JANELA DE TEMPO**

Matheus Fernandes Rabelo

GOIÂNIA
2026

Matheus Fernandes Rabelo

**META-HEURÍSTICA HÍBRIDA PARA O PROBLEMA
DE ROTEAMENTO DE VEÍCULOS COM COLETA E
ENTREGA E JANELA DE TEMPO**

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 Ciência
da Computação.

Orientador(a):

Prof. Me. Alexandre Ribeiro

Banca examinadora:

Prof. Dr. Carmen Cecília

Centeno

Prof. Dr. Maria José Pereira

Dantas

GOIÂNIA

2026

Matheus Fernandes Rabelo

**META-HEURÍSTICA HÍBRIDA PARA O PROBLEMA
DE ROTEAMENTO DE VEÍCULOS COM COLETA E
ENTREGA E JANELA DE TEMPO**

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 Ciência da Computação, em ____/____/_____.

Orientador(a): Prof. Me. Alexandre
Ribeiro

Prof^a. Dr. Maria José Pereira
Dantas

Prof^a. Dr. Carmen Cecília Centeno

GOIÂNIA
2026



AGRADECIMENTOS

Aos meus pais e aos meus irmãos, a minha eterna gratidão por todo o suporte, por serem o meu alicerce e por sempre apoiarem as minhas escolhas.

Ao Professor Alexandre Ribeiro, orientador, treinador de maratona e amigo, pelo apoio, confiança depositada e pelos conselhos que fazem parte do meu contínuo desenvolvimento.

A professora Maria José Pereira Dantas por me coorientar, por apoiar e pela oportunidade de participar do seu time de iniciação científica.

A todos os amigos que conheci durante a realização deste curso.

A todos os professores da universidade e do colégio. Obrigado por todo o conhecimento transmitido.

À Pontifícia Universidade Católica de Goiás e à Direção da Escola Politécnica e de Artes, por fornecerem a infraestrutura e o apoio necessários para o desenvolvimento deste projeto.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela ajuda financeira no decurso de meu trabalho.

□

RESUMO

Este trabalho aborda o Problema de Roteamento de Veículos Verde com Coleta e Entrega e Janelas de Tempo, conhecido como Green-PDPTW. Esse problema consiste em construir rotas para uma frota de veículos, de modo que pedidos de coleta e entrega sejam atendidos respeitando restrições de capacidade, precedência entre coleta e entrega e janelas de tempo. Além do custo operacional, o problema também considera aspectos ambientais, por meio da estimativa das emissões de gases de efeito estufa geradas durante os deslocamentos. Para tratar o problema, foi proposta uma meta-heurística híbrida baseada na combinação da Evolução Diferencial com a Busca Tabu. Como a Evolução Diferencial foi originalmente desenvolvida para problemas contínuos, foi necessária uma adaptação para o espaço discreto do Green-PDPTW, utilizando uma representação baseada em rotas e uma pseudo_solução formada pelos pontos de coleta. A abordagem proposta utiliza operadores de mutação, crossover, reparo e uma etapa de melhoria local por Busca Tabu. Os experimentos computacionais foram realizados com instâncias da classe lrc1, de lrc101 a lrc108, e os resultados foram comparados com soluções conhecidas do conjunto de dados e com resultados apresentados na literatura. Os resultados indicaram que a abordagem proposta foi capaz de reduzir a quantidade de veículos utilizados em relação às soluções comparadas, porém apresentou maiores distâncias em parte das instâncias e limitações relacionadas ao atendimento das janelas de tempo. Ainda assim, o

método demonstrou potencial como alternativa para o Green-PDPTW, principalmente por combinar critérios operacionais e ambientais no processo de avaliação das soluções.

Palavras-chave: Green-PDPTW. Evolução Diferencial. Meta-heurística híbrida.

□

ABSTRACT

This work addresses the Green Pickup and Delivery Problem with Time Windows, known as Green-PDPTW. This problem consists of constructing routes for a fleet of vehicles in order to serve pickup and delivery requests while respecting capacity constraints, precedence between pickup and delivery, and time windows. In addition to the operational cost, the problem also considers environmental aspects by estimating greenhouse gas emissions generated during vehicle movements. To deal with this problem, a hybrid metaheuristic based on the combination of Differential Evolution and Tabu Search was proposed. Since Differential Evolution was originally developed for continuous optimization problems, an adaptation to the discrete search space of the Green-PDPTW was required, using a route-based representation and an auxiliary pseudo-solution composed of pickup points. The proposed approach uses mutation, crossover, repair operators, and a local improvement step based on Tabu Search. Computational experiments were carried out using Irc1 instances, from Irc101 to Irc108, and the results were compared with known solutions from the dataset and with results reported in the literature. The results indicated that the proposed approach was able to reduce the number of vehicles used when compared to the reference solutions; however, it produced longer distances in some instances and presented limitations related to time window feasibility. Even so, the method showed potential as an alternative for the Green-PDPTW, mainly because it combines operational and environmental criteria in the solution evaluation process.

Keywords: Green-PDPTW. Differential Evolution. Hybrid metaheuristic.



LISTA DE FIGURAS

Figura 1 - Exemplo de rotas com coleta e entrega	13
Figura 2- Exemplo de um indivíduo	26
Figura 3- Exemplo do procedimento de mutação	29
Figura 4 - Exemplo do procedimento do crossover	31
Figura 5 - Convergência da solução da instância lrc102	40
Figura 6 - Convergência da solução da instância lrc108	40



LISTA DE TABELAS

Tabela 1- Coeficientes de emissão	19
Tabela 2 - Parâmetros utilizados nos experimentos	37
Tabela 3 - Comparação entre o data set e a solução proposta	38
Tabela 4 - Comparação entre os resultados do artigo e a solução proposta	39
Tabela 5 - Tempo de execução por instância	39



LISTA DE ALGORITMOS

Algoritmo 1 - Evolução Diferencial Híbrida para o Green-PDPTW	28
Algoritmo 2 - Mutação discreta baseada em proximidade	30
Algoritmo 3 - Crossover por rotas	32
Algoritmo 4 - Busca Tabu	34



SUMÁRIO

1 INTRODUÇÃO	12
1.1 Objetivo	15
1.2 Organização Textual	15
2 FORMULAÇÃO MATEMÁTICA	16
2.1 Parâmetros de Entrada	16
2.2 Variáveis de Decisão	17
2.3 Formulação do Modelo	17

<u>2.4 Descrição do Modelo</u>	18	
<u>3 REPRESENTATIVIDADE DO DIESEL NO TRANSPORTE DE CARGAS</u>	19	
<u>4 ABORDAGENS HEURÍSTICAS</u>	20	
<u>4.1 Evolução Diferencial</u>	20	
<u>4.1.1 Mutação Diferencial</u>	21	
<u>4.1.2 Cruzamento e Seleção</u>	21	
<u>4.1.3 Adaptação Discreta da Evolução Diferencial para o Problema Do Caixeiro Viajante</u>	22	
<u>4.2 Busca Tabu</u>	23	
<u>4.2.1 Elementos principais da Busca Tabu</u>	24	
<u>5 PROPOSTA DE SOLUÇÃO</u>	25	
<u>5.1 Estrutura do Algoritmo</u>	25	
<u>5.2 Mutação</u>	29	
<u>5.3 Crossover</u>	31	
<u>5.4 Busca Tabu</u>	32	
<u>6 EXPERIMENTOS COMPUTACIONAIS</u>	35	
<u>6.1 Instâncias</u>	36	
<u>6.2 Parâmetros</u>	37	
<u>6.3 Resultados</u>	38	
<u>6.3.1 Comparações</u>	38	
<u>7 CONCLUSÃO</u>	41	
<u>REFERÊNCIAS</u>	43	

□

2

1 INTRODUÇÃO

Problemas de roteamento de veículos são estudados na área de otimização combinatória por sua aplicação em logística, transporte urbano, distribuição de mercadorias e operações de coleta e entrega. Uma rota pode ser entendida como a sequência de pontos que um veículo deve percorrer para atender clientes ou requisições, podendo sair de um depósito, visitando os pontos necessários e retornando ao depósito ao final do percurso. No Problema de roteamento de veículo (VRP - *Vehicle Routing Problem*), o objetivo é definir rotas para uma frota de veículos de forma a atender um conjunto de clientes com o menor custo possível, respeitando restrições como a capacidade dos veículos. Já no Problema de coleta e entrega com janela de tempo (PDPTW - *Pickup and Delivery Problem with Time*

Windows), a rota precisa considerar pares de coleta e entrega, de modo que cada pedido seja coletado em uma origem e entregue em seu respectivo destino, respeitando a precedência entre coleta e entrega, a capacidade do veículo e as janelas de tempo de atendimento. (MOUNIA; DJEBBAR, 2020)

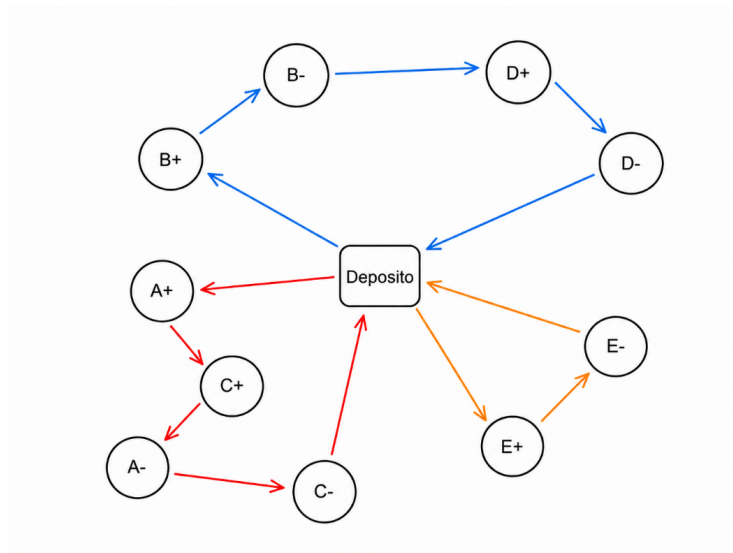
O Problema de Coleta e Entrega com Janela de Tempo com Roteamento Verde (*Green-PPDTW - Green Pickup and Delivery Problem with Time Windows*) busca definir rotas considerando coleta, entrega, carga dos veículos, janelas de tempo e restrições sustentáveis, com foco na redução de custos e emissões de gases do efeito estufa. (BERRAHMANIA; HAMMOU; EL FADI, 2025).

No Problema do Caixeiro Viajante (TSP - *Traveling Salesman Problem*), considera-se apenas um percurso que deve visitar todos os pontos. Já no VRP, essa ideia é ampliada para uma frota de veículos, que deve atender um conjunto de clientes respeitando restrições como capacidade e custo de deslocamento (MOUNIA; DJEBBAR, 2020). A partir do VRP, surgem variações mais próximas de situações reais de transporte. No Problema de Coleta e Entrega (PDP - *Pickup and Delivery Problem*), cada requisição passa a envolver dois pontos relacionados: uma origem, onde ocorre a coleta, e um destino, onde ocorre a entrega de cargas. Assim, o objetivo deixa de ser apenas visitar clientes e passa a ser transportar cargas entre pares de locais, preservando a precedência entre coleta e entrega. Essa estrutura aumenta a complexidade do problema, pois a rota deve garantir que a coleta aconteça antes da entrega correspondente e que a capacidade do veículo não seja ultrapassada durante o percurso (SARTORI; BURIOL, 2020). Em variantes práticas, como problemas com coleta e entrega simultânea, a frota pode partir de um depósito, atender clientes com demandas de entrega e coleta e retornar ao depósito, exigindo uma organização adequada da sequência de atendimento e da carga transportada (HORNSTRA et al., 2020).

Na figura 1 possui um exemplo de 3 veículos atendendo um conjunto de pontos de coleta representados pelo sufixo (+), e um conjunto de pontos de entrega representados pelo sufixo (-), sendo que cada cor representa uma rota de um veículo diferente, todos começando do depósito percorrendo a suas rotas

e posteriormente voltando ao depósito.

Figura 1 - Exemplo de rotas com coleta e entrega



Fonte: elaborada pelo autor.

O PDPTW, acrescenta ao PDP a exigência de que os atendimentos sejam realizados dentro de intervalos de tempo previamente definidos (MOUNIA; DJEBBAR, 2020).

A complexidade do PDPTW não está apenas em definir a ordem de visita dos pontos, mas em construir rotas viáveis que combinem corretamente pedidos, veículos, cargas e intervalos de tempo de atendimento. Uma requisição só pode ser inserida em uma rota quando essa inserção não viola as restrições do problema. Isso significa que o veículo escolhido deve possuir capacidade suficiente para transportar a carga, deve realizar a coleta e a entrega correspondente, deve visitar o ponto de coleta antes do ponto de entrega e deve conseguir atender ambos dentro de suas respectivas janelas de tempo. Além disso, a inserção de uma nova requisição não pode comprometer os atendimentos que já estavam planejados na rota, pois o deslocamento adicional pode gerar atrasos em coletas ou entregas posteriores, tornando a rota inviável do ponto de vista temporal (WANG; DESSOUKY; VAN WOENSEL, 2023). Em algumas aplicações, essa compatibilidade também pode envolver restrições adicionais entre cargas ou pedidos, como ocorre em problemas nos quais determinados produtos não podem ser transportados simultaneamente no mesmo veículo (DENG et al., 2023). Dessa forma, uma solução viável para o PDPTW depende da combinação correta entre

sequência de atendimento, capacidade dos veículos, precedência entre coleta e entrega e restrições temporais (SARTORI; BURIOL, 2020; MOUNIA; DJEBBAR, 2020).

Com a preocupação ambiental, essa família de problemas passa a incorporar também critérios de sustentabilidade. O *Green*-PDPTW mantém as restrições do PDPTW, mas adiciona o impacto ambiental como parte importante da decisão de roteamento. Nesse caso, a melhor rota não é necessariamente a mais curta, pois o consumo de combustível e as emissões de CO₂ dependem também do peso da carga transportada, da distância percorrida e da quantidade de veículos utilizados. Assim, a construção das rotas passa a buscar não apenas menor custo operacional, mas também menor consumo de combustível e menor emissão de poluentes (MOUNIA; DJEBBAR, 2020; BERRAHMANIA; HAMMOU; EL FADI, 2025).

O VRP é classificado como um problema de otimização combinatória formuladas por programação inteira e pertencente à classe NP-difícil (BALINSKI; QUANDT, 1964; LENSTRA; RINNOOY KAN, 1981). Para instâncias de grande porte, não é garantido que as soluções ótimas sejam encontradas dentro de um tempo computacional hábil (Caccetta; Alameen, 2013), como forma de contornar isso, podem ser utilizados métodos combinatórios como meta-heurísticas para resolver o ~~problema em tempo hábil (Gonzalez et al., 2022)~~. Meta-heurísticas, como as de evolução e de inteligência coletiva (ANAND; VINOD CHANDRA, 2022), são usadas para resolver problemas de otimização combinatória, pois oferecem soluções aceitáveis com custo computacional baixo (Hussain et al., 2019). A hibridização tornou-se um campo de pesquisa promissor devido à sua capacidade de adaptar-se a novos problemas. Os algoritmos híbridos visam aproveitar os pontos fortes de diferentes técnicas para compensar os pontos fracos, como equilibrar a diversificação e a intensificação (Lau et al., 2007).

1.1 Objetivo

Assim, o objetivo deste texto é apresentar a abordagem desenvolvida para o problema *Green*-PDPTW. Neste trabalho foi proposto um algoritmo heurístico híbrido utilizando Evolução Diferencial (DE - *Differential*

Evolution) e Busca Tabu (*Tabu Search*).

1.2 Organização Textual

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 apresenta a formulação matemática do problema Green-PDPTW, descrevendo os principais conjuntos, parâmetros, variáveis de decisão, função objetivo e restrições consideradas. O Capítulo 3 discute a representatividade do diesel no transporte rodoviário de cargas e apresenta os coeficientes de emissão utilizados para o cálculo dos gases de efeito estufa. O Capítulo 4 aborda as principais heurísticas utilizadas como base teórica do trabalho, com destaque para a Evolução Diferencial e a Busca Tabu. O Capítulo 5 descreve a proposta de solução desenvolvida, incluindo a estrutura do algoritmo híbrido, a representação das soluções e os procedimentos de mutação, crossover e Busca Tabu. O Capítulo 6 apresenta os experimentos computacionais realizados, as instâncias utilizadas, os parâmetros adotados e a análise dos resultados obtidos. Por fim, o Capítulo 7 apresenta as conclusões do trabalho e sugestões para pesquisas futuras.

□

2 FORMULAÇÃO MATEMÁTICA

A formulação matemática considerada neste trabalho representa o Problema de Roteamento de Veículos Verde com Coleta e Entrega e Janelas de Tempo (*Green-PDPTW*). O modelo considera o custo operacional da solução, formado pela distância total percorrida, pelo custo fixo associado ao uso de veículos e pela penalidade por pedidos não atendidos. Além disso, considera-se o componente ambiental, calculado a partir das emissões geradas nos deslocamentos realizados pelos veículos.

Na implementação computacional, esses componentes são combinados por meio de uma função de avaliação com priorização do custo. Assim, o custo operacional é multiplicado por um valor grande, denominado BigN, e em seguida é somado ao componente ambiental assim simplificando para um problema mono-objetivo. Foi feita inspirada pela formulação matemática do trabalho. A formulação matemática apresentada neste trabalho foi desenvolvida

com base na estrutura proposta por Kunnappapdeelert e Klinsrisuk (2019), que abordam o problema de roteamento verde de veículos por meio da Evolução Diferencial.

2.1 Parâmetros de Entrada

- $P = \{1, \dots, n\}$: conjunto de nós de coleta.
- $D = \{n+1, \dots, 2n\}$: conjunto de nós de entrega.
- $N = P \cup D$: conjunto de pontos de atendimento.
- K : conjunto de veículos disponíveis.
- V : conjunto de vértices do problema.
- A : conjunto de arcos de deslocamento.
- C_k : capacidade de carga do veículo k .
- τ_k e τ'_k : nós de início e fim da rota do veículo k .
- d_{ij} : distância entre o nó i e o nó j .
- t_{ij} : tempo de viagem entre o nó i e o nó j .
- s_i : tempo de atendimento no nó i .
- $[a_i, b_i]$: janela de tempo permitida para atendimento no nó i .
- q_i : quantidade de carga movimentada no nó i .
- ε_{CO_2} , ε_{CH_4} e ε_{N_2O} : fatores de emissão dos gases de efeito estufa por unidade de distância.
- c_v : custo fixo associado à utilização de um veículo.
- p_u : penalidade aplicada para cada pedido de coleta não atendido.
- $BigN$: constante de valor elevado utilizada para priorizar o componente de custo na função de avaliação.
- M : constante suficientemente grande utilizada nas restrições condicionais.

2.2 Variáveis de Decisão

- $x_{ijk} \in \{0, 1\}$: assume valor 1 se o veículo k percorre o arco entre os nós i e j ; caso contrário, assume valor 0.
- $S_{ik} \geq 0$: instante de início do atendimento do veículo k no nó i .
- $Q_{ik} \geq 0$: carga acumulada no veículo k após atender o nó i .
- $y_k \in \{0, 1\}$: assume valor 1 se o veículo k é utilizado na solução; caso contrário, assume valor 0.
- $u_i \in \{0, 1\}$: assume valor 1 se o pedido associado ao nó de coleta i não é atendido; caso contrário, assume valor 0.

2.3 Formulação do Modelo

A função objetivo considerada é definida por:

Minimize

(1)

(2)

sujeito a:

(3)

(4)

(5)

(6)

(7)

(8)

(9)

(10)

(11)

(12)

(13)

(14)

(15)

2.4 Descrição do Modelo

A Equação (1) apresenta o componente de custo operacional da solução. Esse componente é composto pela distância total percorrida, pelo custo fixo de utilização dos veículos e pela penalidade aplicada aos pedidos não atendidos. A Equação (2) representa o componente ambiental do modelo, calculado a partir da distância percorrida nos arcos utilizados e dos fatores de emissão dos gases CO_2 , CH_4 e N_2O . Na implementação, o componente de custo é multiplicado por BigN e somado ao componente ambiental, de modo que o custo operacional tenha prioridade na avaliação das soluções. A Equação (3) garante que cada pedido de coleta seja atendido uma única vez ou, caso não seja atendido, seja contabilizado pela variável de penalidade u_i . A Equação (4) assegura que a coleta e a entrega correspondente sejam realizadas pelo mesmo veículo. As Equações (5) e (6) relacionam o uso do veículo à saída e ao retorno ao depósito: se o veículo k for utilizado, então $y_k = 1$; caso contrário, o veículo não deve iniciar nem finalizar uma rota de atendimento. A Equação (7) representa a conservação de fluxo, garantindo a continuidade das rotas ao longo dos nós de atendimento. As Equações (8), (9) e (10) controlam o aspecto temporal do problema. A Equação (8) atualiza o instante de atendimento entre dois nós consecutivos quando o arco é utilizado. A Equação (9) impõe o respeito às janelas de tempo. A Equação (10) garante a precedência entre coleta e entrega, isto é, a entrega correspondente só pode ocorrer após sua respectiva coleta. O termo $M u_i$ relaxa essa restrição quando o pedido não é atendido. A Equação (11) representa a atualização da carga transportada pelo veículo após cada deslocamento, enquanto a Equação (12) limita a carga acumulada à capacidade máxima do veículo. Por fim, as Equações (13), (14) e (15) definem a natureza binária das variáveis de decisão associadas, respectivamente, aos deslocamentos realizados, à utilização dos veículos e ao não atendimento dos pedidos.

□

3 REPRESENTATIVIDADE DO DIESEL NO TRANSPORTE DE CARGAS

Neste trabalho, optou-se por utilizar os coeficientes de emissão referentes ao diesel, pois esse combustível ainda possui grande predominância no transporte rodoviário de cargas e em veículos pesados. Segundo Gómez Vilchez et al. (2022), os dados europeus de transporte de cargas indicam que o diesel domina a atividade do setor, sendo uma das principais fontes associadas às emissões em veículos comerciais pesados. De forma semelhante, Gerutu et al. (2025) apontam que a demanda por diesel é superior à de outros combustíveis devido ao seu uso disseminado em veículos pesados que realizam percursos de longa distância. Além disso, Almeida et al. (2026) destacam que, mesmo diante das estratégias de descarbonização, os veículos movidos a diesel ainda representam a maior

parcela da frota de transporte rodoviário de cargas na União Europeia, correspondendo a cerca de 90.5% da frota em 2023.

A Tabela 1 apresenta os coeficientes de emissão utilizados no modelo, considerando diferentes níveis de carga do veículo. Esses valores foram obtidos a partir do trabalho de Kunnapapdeelert e Klinsrisuk (2019), que apresenta fatores de emissão para caminhões de 10 toneladas movidos a diesel.

Tabela 1- Coeficientes de emissão

Estado do veículo	Carga do veículo (%)	Fator de emissão CO ₂ (kg/km)	Fator de emissão CH ₄ (kg/km)	Fator de emissão N ₂ O (kg/km)
Vazio	0	0.786442	0.000266	0.005654
Baixa carga	25	0.850208	0.000288	0.006112
Meia carga	50	0.913974	0.000310	0.006570
Alta carga	75	0.975082	0.000330	0.007010
Carga completa	100	1.036191	0.000351	0.007449

Fonte: elaborada pelo autor com base em Kunnapapdeelert e Klinsrisuk (2019).



4 ABORDAGENS HEURÍSTICAS

Neste trabalho, são utilizadas duas abordagens: a Evolução Diferencial e a Busca Tabu. A Evolução Diferencial é empregada como método populacional responsável por gerar e combinar diferentes soluções candidatas ao longo das gerações. Já a Busca Tabu é utilizada como estratégia de melhoria local, buscando refinar as soluções produzidas pela Evolução Diferencial por meio de movimentos de realocação de pares coleta-entrega entre veículos.

Dessa forma, este capítulo apresenta os conceitos fundamentais dessas duas meta-heurísticas, destacando suas principais características e sua relação com a abordagem híbrida proposta para o Green-PDPTW.

4.1 Evolução Diferencial

A Evolução Diferencial, do inglês Differential Evolution (DE), é uma meta-heurística populacional proposta por Storn e Price (1997) para resolver problemas de otimização em espaços contínuos. Em sua formulação original, cada solução candidata é representada por um vetor de números reais, e o

processo de busca ocorre por meio da evolução de uma população de soluções ao longo de sucessivas gerações. O método foi apresentado como uma heurística simples e eficiente para otimização global de funções possivelmente não lineares e não diferenciáveis em espaços contínuos (STORN; PRICE, 1997).

Diferentemente de uma busca local simples, que parte de uma única solução e explora sua vizinhança, o DE trabalha com várias soluções simultaneamente. Essa característica permite manter diversidade populacional durante o processo de busca, possibilitando que diferentes regiões do espaço de soluções sejam exploradas ao mesmo tempo. Por esse motivo, o DE é classificado como um método populacional e estocástico de otimização global, sendo amplamente utilizado em problemas de busca contínua (QIN; HUANG; SUGANTHAN, 2009).

O funcionamento básico do DE envolve quatro etapas principais: inicialização da população, mutação, cruzamento e seleção. Na inicialização, são geradas soluções candidatas. Na mutação, um vetor doador é produzido a partir da combinação entre indivíduos da população. Em seguida, o cruzamento combina informações do vetor alvo com o vetor doador, gerando uma solução experimental. Por fim, a seleção compara a solução experimental com a solução original e mantém aquela que apresenta melhor valor de função objetivo para a próxima geração (STORN; PRICE, 1997).

4.1.1 Mutação Diferencial

A mutação diferencial é o operador mais característico da Evolução Diferencial, pois é responsável por gerar novas soluções a partir da combinação entre indivíduos da própria população. A ideia central é que a diferença entre dois vetores contém informação sobre uma direção de busca no espaço de soluções. Assim, em vez de realizar perturbações totalmente aleatórias, o DE utiliza a distribuição atual da população para produzir novos candidatos, explorando as relações entre soluções já existentes (STORN; PRICE, 1997; DAS; SUGANTHAN, 2011).

Na estratégia clássica conhecida como DE/rand/1, três

indivíduos distintos são selecionados aleatoriamente da população. A diferença entre dois desses indivíduos é multiplicada por um fator de escala F e somada a um terceiro indivíduo, chamado vetor base. Essa operação gera o vetor doador v , conforme apresentado na Equação (16).

Nessa expressão, v representa o vetor doador, x_{r_1} é o vetor base, x_{r_2} e x_{r_3} são vetores distintos da população, e F é o fator de escala responsável por controlar a intensidade da diferença utilizada na mutação. Valores menores de F tendem a produzir modificações mais conservadoras, enquanto valores maiores ampliam o deslocamento gerado pelo vetor diferença, aumentando a capacidade exploratória do algoritmo (OPARA; ARABAS, 2019; GONG; CAI, 2013).

4.1.2 Cruzamento e Seleção

Após a etapa de mutação, o DE realiza o cruzamento entre o vetor doador e o vetor alvo, com o objetivo de gerar um vetor experimental. Esse operador permite combinar informações da solução original com as informações produzidas pela mutação, aumentando a diversidade das soluções candidatas. Em geral, o cruzamento no DE pode ocorrer por diferentes estratégias, sendo o cruzamento binomial uma das formas mais utilizadas. Nesse caso, cada componente do vetor experimental pode ser herdado do vetor doador ou do vetor alvo, de acordo com a taxa de cruzamento CR (STORN; PRICE, 1997; DAS; SUGANTHAN, 2011).

A taxa CR controla a probabilidade de o vetor experimental receber componentes do vetor doador. Quando CR possui valor elevado, o vetor experimental tende a herdar mais características da solução gerada pela mutação. Por outro lado, valores menores de CR fazem com que o vetor experimental preserve mais características do vetor alvo. Assim, esse parâmetro influencia diretamente o equilíbrio entre exploração de novas regiões do espaço de busca e preservação de características já presentes na população (DAS; SUGANTHAN, 2011).

Após o cruzamento, ocorre a etapa de seleção. No DE clássico, a seleção é realizada de forma gulosa, comparando o vetor experimental com o vetor alvo. Em problemas de minimização, o vetor experimental

substitui o vetor alvo apenas se apresentar menor valor de função objetivo; caso contrário, o vetor alvo permanece na população. Essa estratégia é conhecida como seleção um-para-um, pois cada vetor experimental compete diretamente com o vetor alvo que o originou (STORN; PRICE, 1997; STORN, 2008).

Dessa forma, o cruzamento contribui para gerar novas soluções a partir da combinação entre indivíduos, enquanto a seleção garante que apenas soluções iguais ou melhores, segundo a função objetivo adotada, avancem para a geração seguinte. Em problemas de roteamento, essa função objetivo pode considerar distância, custo operacional, tempo de atendimento, emissões ou uma combinação ponderada desses critérios, dependendo da formulação utilizada.

4.1.3 Adaptação Discreta da Evolução Diferencial para o Problema Do Caixeiro Viajante

O Problema do Caixeiro Viajante (TSP-*Traveling Salesman Problem*), é um dos problemas clássicos da otimização combinatória. De forma geral, seu objetivo é encontrar uma rota de menor custo que visite todos os vértices exatamente uma vez e retorne ao vértice de origem. Dantzig, Fulkerson e Johnson (1954) apresentam uma das formulações clássicas do problema, tratando-o como a busca por uma rota mínima que percorre um conjunto de cidades e retorna ao ponto inicial. Embora o TSP tenha uma formulação simples, sua resolução exata pode se tornar computacionalmente difícil em instâncias maiores. Essa dificuldade está relacionada ao fato de o problema estar entre os problemas combinatórios associados à classe NP-completa. (KARP, 1972).

O artigo de Figueiredo, Fernandes e Zavala (2018) aborda exatamente esse problema. Os autores propõem uma adaptação da Evolução Diferencial para o TSP, utilizando as coordenadas cartesianas dos vértices como forma de preservar parte da lógica contínua do DE. Em vez de aplicar diretamente operações aritméticas sobre a permutação, o método calcula pontos auxiliares no plano e depois converte esses pontos em vértices reais do problema.

A ideia pode ser entendida em três etapas. Primeiro, selecionam-se vértices pertencentes a

soluções da população. Segundo, calcula-se um ponto no espaço contínuo a partir das coordenadas desses vértices, utilizando uma expressão inspirada na mutação diferencial. Terceiro, identifica-se o vértice real mais próximo desse ponto e o utiliza-se na composição da nova solução.

Essa estratégia permite que o DE continue usando uma ideia contínua para guiar a busca, mas sem abandonar a natureza discreta do TSP. O resultado final permanece sendo uma sequência de vértices válidos, e não um vetor de valores reais sem significado para o problema.

4.2 Busca Tabu

A Busca Tabu é uma meta-heurística proposta por Glover (1989) com o objetivo de ampliar a capacidade dos métodos de busca local em problemas de otimização combinatória. Diferentemente de uma busca local tradicional, que tende a aceitar apenas movimentos que melhoram imediatamente a solução, a Busca Tabu permite explorar soluções vizinhas de forma mais flexível, inclusive aceitando movimentos que não geram melhora imediata. Essa característica ajuda o algoritmo a escapar de ótimos locais, evitando que a busca fique presa em uma região limitada do espaço de soluções.

A principal estratégia da Busca Tabu é o uso de estruturas de memória para controlar o processo de busca. A memória de curto prazo, geralmente chamada de lista tabu, registra atributos de movimentos realizados recentemente e impede que esses movimentos sejam repetidos ou revertidos por um número determinado de iterações. Dessa forma, o algoritmo reduz a possibilidade de ciclos, isto é, de retornar repetidamente às mesmas soluções (GLOVER, 1989).

Entretanto, a lista tabu não funciona como uma proibição absoluta. Em muitos casos, utiliza-se um critério de aspiração, que permite aceitar um movimento tabu caso ele produza uma solução suficientemente boa, por exemplo, melhor que a melhor solução encontrada até o momento. Assim, a Busca Tabu combina restrição e flexibilidade: restringe movimentos recentes para evitar ciclos, mas permite exceções quando uma solução candidata apresenta qualidade

relevante para o processo de busca (GLOVER, 1989).

4.2.1 Elementos principais da Busca Tabu

A Busca Tabu pode ser compreendida a partir de quatro elementos principais: solução atual, vizinhança, lista tabu e critério de aspiração. A solução atual representa o ponto em que a busca se encontra em uma determinada iteração. A vizinhança corresponde ao conjunto de soluções que podem ser obtidas por pequenas modificações na solução atual. A lista tabu armazena atributos de movimentos recentes, impedindo sua repetição imediata. Já o critério de aspiração permite aceitar um movimento tabu quando ele leva a uma solução melhor que a melhor solução encontrada até então (GLOVER, 1989).

Em problemas de roteamento, a vizinhança pode ser construída por diferentes operadores de movimento, como troca de clientes, inserção, remoção e realocação entre veículos. Gendreau, Hertz e Laporte (1994), por exemplo, propuseram uma heurística de Busca Tabu para o Problema de Roteamento de Veículos em que soluções vizinhas são geradas pela remoção de um vértice de sua rota atual e sua reinserção em outra rota. Essa ideia se aproxima da lógica utilizada neste trabalho, pois a Busca Tabu implementada também realiza movimentos de realocação entre veículos, porém considerando o par coleta-entrega como uma unidade de movimentação.

No caso do PDPTW, os movimentos de vizinhança precisam considerar restrições adicionais em relação ao VRP tradicional. Cada pedido é composto por uma coleta e uma entrega correspondente, sendo necessário garantir que a entrega não ocorra antes da coleta e que ambas sejam atendidas pelo mesmo veículo. Por isso, movimentos simples, como deslocar apenas um nó isolado, podem gerar soluções estruturalmente inválidas. Nesse contexto, a realocação do par completo (p,d) , formado por um coleta p e sua respectiva entrega d , é uma estratégia adequada para preservar a relação de precedência e acoplamento entre os nós.

□

5 PROPOSTA DE SOLUÇÃO

Mesmo sendo menos recorrente em problemas de

roteamento, alguns estudos mostram que a Evolução Diferencial pode ser adaptada e utilizada de forma eficiente nesse contexto. Kunnapapdeelert e Klinsrisuk (2019), por exemplo, aplicaram o DE ao PDPTW e ao Green-PDPTW, demonstrando que o algoritmo foi capaz de construir rotas para ambos os problemas. De forma semelhante, Lai e Cao (2010) propuseram uma versão aprimorada da Evolução Diferencial para o problema de roteamento de veículos com coletas e entregas simultâneas e janelas de tempo, reforçando a viabilidade do método em variantes complexas do VRP.

A escolha da Evolução Diferencial como base deste trabalho ocorreu pela intenção de aplicar uma meta-heurística ainda pouco explorada em problemas de roteamento. Durante a revisão da literatura, foi identificado o estudo de Kunnapapdeelert e Klinsrisuk (2019), que utilizou o algoritmo DE para resolver o PDPTW e o Green-PDPTW. Além disso, a proposta deste trabalho foi motivada pela própria observação dos autores, ao afirmarem que “o desempenho da Evolução Diferencial pode ser ainda mais aprimorado se for aplicada a combinação do algoritmo DE com busca local ou outras heurísticas de busca” (KUNNAPAPDEELERT; KLINSRISUK, 2019, p. 405, tradução nossa). Dessa forma, este estudo busca explorar uma abordagem híbrida, combinando Evolução Diferencial com Busca Tabu para o problema Green-PDPTW.

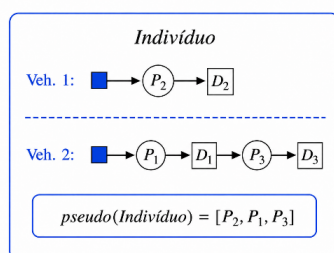
5.1 Estrutura do Algoritmo

A abordagem proposta neste trabalho combina a Evolução Diferencial com a Busca Tabu, com o objetivo de resolver o Green-PDPTW considerando aspectos de custos operacionais e ambientais. Nesse contexto, cada solução é representada por um conjunto de rotas associadas aos veículos disponíveis, de modo que um indivíduo da população corresponde a uma possível distribuição dos pedidos de coleta e entrega entre esses veículos, conforme ilustrado na Figura 2, o veículo 1 deve, em sua rota, passar no ponto P2 para coletar para depois entregar no ponto de entrega D2, já no veículo 2 ele irá passar nos pontos P1 para coleta, depois D1 para entrega, P3 para coleta e posteriormente D3 para entrega, assim demonstrando que para uma entrega ser realizada, é necessário que sua coleta correspondente tenha sido efetuada anteriormente. Além disso, a coleta

e a entrega de um mesmo pedido devem ser atendidas pelo mesmo veículo, garantindo a relação de precedência e continuidade do transporte.

Uma solução S pode ser representada como um conjunto de K rotas, sendo K a quantidade de veículos disponíveis na instância. Cada rota, por sua vez, é formada por pontos de coleta e entrega, representados respectivamente pelos prefixos P (coleta) e D (entrega). Além da representação completa da solução, o algoritmo utiliza também uma representação auxiliar chamada *pseudo_solução*. Essa *pseudo_solução* contém apenas os pontos de coleta e é utilizada nas etapas de mutação e crossover do DE, após a aplicação desses operadores, a solução é reconstruída por meio de um processo de reparo, que transforma a *pseudo_solução* em uma solução. Dessa forma, a solução é utilizada para avaliação, como cálculo da distância percorrida, estimativa das emissões e aplicação da Busca Tabu, enquanto a *pseudo_solução* facilita a manipulação dos indivíduos durante os operadores evolutivos. Essa separação facilita a adaptação da Evolução Diferencial, originalmente proposta para problemas contínuos, ao contexto discreto do Green-PDPTW, que envolve coleta, entrega, capacidade dos veículos e janelas de tempo.

Figura 2- Exemplo de um indivíduo



Fonte: elaborada pelo autor.

O Algoritmo 1 recebe como entrada os dados da instância do Green-PDPTW e os parâmetros que controlam a execução da meta-heurística. O parâmetro *tasks* representa o conjunto de tarefas da instância, incluindo os pontos de coleta e entrega, e suas coordenadas(x,y), demandas, tempos de serviço, janelas de tempo e relações entre coleta e entrega.

O parâmetro K indica a quantidade de veículos disponíveis, enquanto Q representa a capacidade máxima de carga de cada veículo. Assim, cada indivíduo da população pode ser interpretado como um conjunto de até K rotas, respeitando a capacidade definida por Q .

O parâmetro NP define o tamanho da população, ou seja, a quantidade de soluções candidatas mantidas em cada geração. Já $G_{\max}$ representa o número máximo de gerações, funcionando como critério de parada do algoritmo.

O fator F é utilizado na etapa de mutação, controlando a intensidade do deslocamento entre o indivíduo selecionado aleatoriamente e o melhor indivíduo da geração. A taxa CR , por sua vez, é utilizada no crossover e define a probabilidade de uma rota do indivíduo atual ser substituída por uma rota da solução mutante.

Além desses parâmetros, o algoritmo também utiliza configurações associadas à Busca Tabu, como o número de iterações, o tamanho da lista tabu e a quantidade de movimentos candidatos avaliados. Esses parâmetros regulam a intensidade do refinamento local aplicado às soluções geradas durante o processo evolutivo.

O Algoritmo 1 apresenta a estrutura geral da meta-heurística híbrida proposta, combinando Evolução Diferencial com Busca Tabu para o Green-PDPTW. Inicialmente, na linha 2, é gerada uma população aleatória com NP indivíduos, em que cada indivíduo representa uma possível solução composta por rotas associadas aos veículos disponíveis. Em seguida, na linha 3, é definido S^* , correspondente ao melhor indivíduo da população inicial segundo a função de avaliação.

O processo evolutivo ocorre entre as linhas 4 e 23, sendo repetido até atingir o número máximo de gerações $G_{\max}$. A cada geração, uma nova população é inicializada vazia na linha 5, e o melhor indivíduo da geração anterior e representado por S_g , é selecionado na linha 6. Em seguida, para cada indivíduo S_j da população anterior, executam-se as etapas de geração de uma nova solução candidata das linhas 7 a 21.

Na linha 8, um indivíduo X_r é selecionado

aleatoriamente da população. Esse indivíduo é utilizado junto com o melhor indivíduo da geração (x_{best}) no processo de mutação, realizado na linha 9 e gerado a pseudo_solução mutante V , buscando explorar novas regiões do espaço de busca a partir da combinação entre uma solução aleatória e a melhor solução da geração.

Após isso, na linha 10, é realizado o *crossover* entre o indivíduo atual S_j e a pseudo_solução mutante V , gerando uma nova solução candidata U , os operadores da Evolução Diferencial podem produzir soluções inconsistentes para o Green-PDPTW, a dentro do crossover e feito um reparo em U . Essa etapa reconstrói a solução a partir das coletas.

Na linha 12, a solução candidata U é aprimorada por meio da Busca Tabu. Essa busca local realiza movimentos de realocação de pares coleta-entrega entre rotas, procurando melhorar a solução sem retornar imediatamente as soluções já visitadas.

Por fim, entre as linhas 13 e 20, ocorre a etapa de seleção. Caso a nova solução U apresente um *fitness* menor que o *fitness* do indivíduo original S_j , ela é considerada uma solução melhor e passa a compor a nova população. Caso contrário, a própria solução S_j é mantida. Além disso, se U tiver um *fitness* melhor que o *fitness* da solução global S^* , então S^* é atualizado recebendo a solução U . Na linha 22, a população anterior é substituída pela nova população. O algoritmo termina retornando S^* , isto é, a melhor solução encontrada durante todo o processo.

Algoritmo 1 - Evolução Diferencial Híbrida para o Green-PDPTW

Entrada: tasks, K , Q , NP , F , CR , G_{max}

Saída: Melhor solução encontrada S^*

```

1 Início
2   popu <- Gerar população inicial aleatória com  $NP$  indivíduos
3    $S^*$  <- Melhor indivíduo de popu segundo a função fitness
4   Para  $g$  <- 1 até  $G_{max}$  faça
5     nova_popu <-  $\emptyset$ 
6      $X_{best}$  <- Melhor indivíduo da geração anterior
7     Para cada indivíduo  $S_j$  em popu faça
8       Selecionar aleatoriamente um indivíduo  $X_r$  da população
9        $V$  <- Mutação( $X_r$ ,  $X_{best}$ ,  $K$ , tasks,  $F$ )
10       $U$  <- Crossover( $S_j$ ,  $V$ ,  $K$ ,  $CR$ )
11       $U$  <- Aplicar Busca Tabu em  $U$ 
12      Se fitness( $U$ ) < fitness( $S_j$ ) então
13        Adicionar  $U$  em nova_popu
14      Se fitness( $U$ ) < fitness( $S^*$ ) então

```

```

15       $S^* \leftarrow U$ 
16      FimSe
17      Senão
18          Adicionar  $S_j$  em nova_popu
19      FimSe
20      FimPara
21      popu  $\leftarrow$  nova_popu
22      FimPara
23      Retornar  $S^*$ 
24 Fim

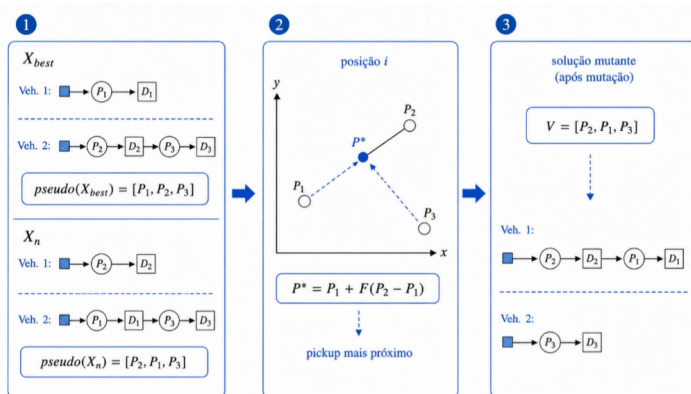
```

5.2 Mutação

A mutação, é uma função que é utilizada na linha 9 do Algoritmo 1, tem como objetivo gerar uma nova solução candidata a partir da combinação entre um indivíduo escolhido aleatoriamente da população, representado por X_r , e a melhor solução da geração atual, representada por X_{best} . Como o Green-PDPTW é um problema discreto, a mutação não é aplicada diretamente sobre vetores reais, como ocorre na Evolução Diferencial clássica. Por isso, antes da aplicação do operador, as soluções são convertidas em pseudo_solução.

Na Figura 3, o passo 1 ilustra essa conversão. Cada indivíduo é originalmente representado por rotas completas, contendo pontos de coleta e entrega distribuídos entre os veículos, no passo 2 mostra um exemplo do processo de escolha de um ponto que P , já no passo 3 mostra a pseudo_solução mutante após as operações, que posteriormente vai ser reparada em uma solução mutante V .

Figura 3- Exemplo do procedimento de mutação



Fonte: elaborada pelo autor.

Nas linhas 2 e 3 é feita uma pseudo_solução de X_r e X_{best} que são chamadas de $perm_r$ e $perm_{best}$, já na linha 4 é obtido o tamanho de $perm_r$. Em cada iteração, o algoritmo seleciona a coleta da posição i em $perm_r$ e a coleta correspondente em X_{best} , como apresentado nas linhas 8 e 9. Em seguida, nas linhas 10 e 11, são obtidas as coordenadas desses dois pontos.

A partir das coordenadas de coletas selecionadas, é gerado um ponto intermediário no espaço contínuo, utilizando o fator de escala F . Esse cálculo corresponde às linhas 12 e 13 do Algoritmo 2 e demonstrado pelas equações 17 e 18.

$$(17)$$

$$(18)$$

Esse novo ponto $P^* = (P_x, P_y)$ representa uma tentativa de deslocamento da coleta do indivíduo aleatório em direção a coleta da melhor solução. No entanto, como P^* é um ponto calculado no espaço contínuo, ele pode não corresponder exatamente a uma coleta existente na instância. Por isso, o algoritmo seleciona a coleta disponível mais próximo de P^* , utilizando uma estrutura *KD-Tree* para localizar o ponto de coleta mais próximo que pertence a P . Dessa forma, o valor contínuo produzido pela mutação é convertido novamente para uma representação discreta, conforme indicado na linha 14 do Algoritmo 2. Esse p_{new} é inserido na *pseudo_solução_mutante* e removido do conjunto disponível para escolha da *KD-Tree* e atualiza a aparição do p_{new} , como demonstrado nas linhas 14 a 17. Conforme a linha 19 e 20 e contruida a *solução_mutante* V a partir da *pseudo_solução_mutante* V e retornado,

Algoritmo 2 - Mutação discreta baseada em proximidade

Entrada: $X_r, X_{best}, K, \text{tasks}, F$

Saída: Solução mutante V

```

1 Início
2   $perm_r \leftarrow$  Extrair coletas únicos de  $X_r$ 
3   $perm_{best} \leftarrow$  Extrair coleta únicos de  $X_{best}$ 
4   $n \leftarrow$  tamanho de  $perm_r$ 
5  disponíveis  $\leftarrow$  Conjunto com todos as coletas
6   $pseudo\_solução\_mutante \leftarrow \emptyset$ 
7  Para  $i \leftarrow 1$  até  $n$  faça
8     $p_r \leftarrow perm_r[i]$ 
9     $p_{best} \leftarrow perm_{best}[i]$ 
10    $(x_r, y_r) \leftarrow$  Coordenadas de  $p_r$ 
11    $(x_{best}, y_{best}) \leftarrow$  Coordenadas de  $p_{best}$ 
12    $P_x \leftarrow x_r + F * (x_{best} - x_r)$ 
13    $P_y \leftarrow y_r + F * (y_{best} - y_r)$ 
14    $p_{new} \leftarrow$  Coletas disponíveis mais próximo de  $(P_x, P_y)$ 
15    $pseudo\_solução\_mutante[i] \leftarrow p_{new}$ 
16   Remover  $p_{new}$  do conjunto disponíveis
17   Atualizar contagem de uso de  $p_{new}$ 
18 FimPara
19  $V \leftarrow$  Reconstruir solução a partir de  $pseudo\_solução\_mutante$ 
20 Retornar  $V$ 
21 Fim

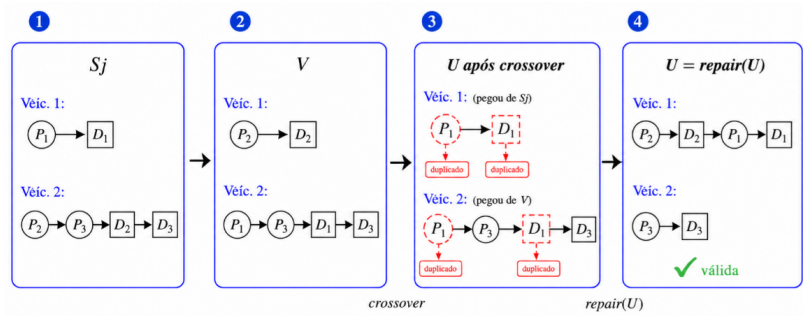
```

5.3 Crossover

O *crossover* tem como objetivo gerar uma solução experimental U a partir da combinação entre o indivíduo atual da população, representado por S_j , e a solução mutante V. Além de receber a quantidade de veículos K e a taxa de cruzamento CR.

Na Figura 4, as duas primeiras partes mostram as soluções S_j e V, respectivamente. Já na terceira parte mostra um possível cenário após o crossover entre S_j e V e na quarta parte mostra a solução U após ser reparada.

Figura 4 - Exemplo do procedimento do crossover



Fonte: elaborada pelo autor.

Na linha 2 do algoritmo 3 inicia criando U como uma cópia de S_j , essa cópia é necessária porque o crossover deve criar uma nova solução candidata sem modificar diretamente o indivíduo original da população. Já nas 3 a 7 é feito um processo de escolha de quais veículos que compõem S_j e V vão fazer parte da solução U. Das linhas 4 a 6, o algoritmo decide se o veículo (i) da solução (U) receberá o veículo da solução mutante (V_i) ou se manterá a rota original de ($S_j(i)$). Essa decisão é feita a partir de um número aleatório (r), gerado entre 0 e 1. Se ($r < CR$), então (U_i) recebe (V_i); caso contrário, (U_i) permanece igual à cópia inicial de ($S_j(i)$).

Algoritmo 3 - Crossover por rotas

Entrada: S_j , V, K, CR

Saída: solução experimental U

1 Início

2 $U \leftarrow S_j$

```

3  Para  $v \leftarrow 1$  até  $K$  faça
4      se  $\text{aleatório}(0,1) < CR$  então
5           $U[v] \leftarrow V[v]$ 
6      fim
7  fim
8   $U \leftarrow \text{repair}(U)$ 
9  retorne  $U$ 
10 Fim

```

5.4 Busca Tabu

A Busca Tabu é utilizada neste trabalho como uma etapa de melhoria local aplicada à solução gerada pela Evolução Diferencial. Após a construção da solução candidata U , obtida pelo cruzamento e pela etapa de reparo, a Busca Tabu busca melhorar essa solução por meio de pequenos movimentos locais. No contexto do Green-PDPTW, esses movimentos consistem na realocação de pedidos entre veículos, preservando a estrutura básica de coleta e entrega do problema.

O Algoritmo 4 (Busca Tabu) recebe como entrada a solução S , que representa a solução candidata gerada anteriormente (U) pela Evolução Diferencial. Essa solução é composta por um conjunto de veículos, em que cada veículo possui uma sequência de pontos de coleta e entrega. A Busca Tabu utiliza essa solução como ponto de partida para realizar pequenas modificações locais. O parâmetro *tenure* define por quantas iterações um pedido movimentado permanece na lista tabu. Dessa forma, após um par coleta-entrega ser realocado, a respectiva coleta fica temporariamente impedido de ser movimentado novamente, evitando que o algoritmo desfça imediatamente movimentos recentes. O parâmetro *tabu_iters* indica o número máximo de iterações da Busca Tabu. Em cada iteração, o algoritmo tenta encontrar uma melhoria local para a solução atual. Já o parâmetro *tabu_cands* define quantos movimentos candidatos serão gerados e avaliados em cada iteração. Assim, quanto maior o valor de *tabu_cands*, maior será a quantidade de alternativas testadas antes de escolher o melhor movimento admissível.

Nas linhas 2 e 3, o algoritmo inicializa duas soluções: S_{atual} e S^* . A solução S_{atual} representa a solução que será modificada ao longo da busca, enquanto S^* armazena a melhor solução encontrada até o momento. Na linha 4, é criada a lista tabu, inicialmente vazia. Essa lista registra temporariamente as coletas

recentemente movimentadas, evitando que o algoritmo repita ou reverta movimentos muito recentes.

O processo principal ocorre entre as linhas 5 e 28. A cada iteração, o algoritmo procura o melhor movimento admissível entre um conjunto de candidatos. Na linha 6, a variável `melhor_candidato` é inicializada como vazia, pois nenhum candidato foi avaliado ainda. Em seguida, entre as linhas 7 e 19, o algoritmo gera e avalia `tabu_cands` movimentos candidatos.

Entre as linhas 8 e 11, o algoritmo escolhe um veículo de origem `vA`, seleciona uma coleta `p` atendido por esse veículo, identifica a entrega correspondente e escolhe um veículo de destino `vB`. O veículo de origem `vA` representa o veículo do qual um pedido será temporariamente retirado. Já o veículo de destino `vB` indica onde esse pedido será reinserido, podendo ser o mesmo veículo ou outro veículo da solução.

Na linha 12, é criada uma solução candidata por meio da realocação do par (p,d) de `vA` para `vB`. Esse movimento remove o par coleta-entrega do veículo de origem e tenta reinseri-lo em uma nova posição no veículo de destino. A estrutura básica do Green-PDPTW é preservada porque a coleta e a sua respectiva entrega são movimentadas em conjunto. Além disso, a reinserção mantém a coleta antes da entrega, respeitando a relação de precedência entre coleta e entrega.

Na linha 13, o algoritmo verifica se o candidato é permitido. Essa verificação considera a lista `tabu` e o critério de aspiração. Caso a coleta movimentada esteja temporariamente `tabu`, o movimento normalmente é evitado. Entretanto, se esse movimento gerar uma solução melhor que a melhor solução encontrada até o momento, ele pode ser aceito pelo critério de aspiração.

Entre as linhas 14 e 18, o algoritmo compara o candidato atual com o melhor candidato já encontrado na iteração. Caso o candidato apresente menor valor de `fitness`, ele passa a ser armazenado como `melhor_candidato`, e a coleta movimentado é registrado como `melhor_movimento`. Dessa forma, ao final da avaliação dos candidatos, o algoritmo mantém o melhor movimento admissível da iteração.

Nas linhas 20 a 22, caso nenhum candidato

admissível tenha sido encontrado, a busca é encerrada. Caso contrário, na linha 23, a solução S_{atual} é atualizada com o melhor_candidato. Em seguida, na linha 24, o melhor_movimento é adicionado à lista tabu, impedindo que a mesma coleta seja movimentada novamente por um determinado número de iterações.

Por fim, entre as linhas 25 e 27, o algoritmo verifica se a solução atual possui melhor valor de fitness que S^* . Se isso ocorrer, S^* é atualizada. Ao final do processo, a linha 29 retorna S^* , isto é, a melhor solução encontrada pela Busca Tabu. Dessa forma, a Busca Tabu funciona como um refinamento local da solução, testando realocações de pares coleta-entrega entre veículos e buscando reduzir o valor da função de avaliação.

Algoritmo 4 - Busca Tabu

Entrada: S, tasks, K, tenure, tabu_iters, tabu_cands

Saída: Melhor solução S^*

```

1  Início
2   $S_{atual} \leftarrow S$ 
3   $S^* \leftarrow S$ 
4  lista_tabu  $\leftarrow \emptyset$ 
5  Para it  $\leftarrow 1$  até tabu_iters faça
6    melhor_candidato  $\leftarrow$  vazio
7    Para c  $\leftarrow 1$  até tabu_cands faça
8      vA  $\leftarrow$  veículo de origem aleatório
9      p  $\leftarrow$  coleta aleatório de vA
10     d  $\leftarrow$  entrega correspondente de p
11     vB  $\leftarrow$  veículo de destino aleatório
12     candidato  $\leftarrow$  realocar (p, d) de vA para vB
13     se candidato é permitido então
14       se fitness(candidato) < fitness(melhor_candidato) então
15         melhor_candidato  $\leftarrow$  candidato
16         melhor_movimento  $\leftarrow$  p
17     fim
18   fim
19   fim
20   se melhor_candidato = vazio então
21     parar busca
22   fim
23    $S_{atual} \leftarrow$  melhor_candidato
24   adicionar melhor_movimento em lista_tabu
25   se fitness( $S_{atual}$ ) < fitness( $S^*$ ) então
26      $S^* \leftarrow S_{atual}$ 
27   fim
28   fim
29   retorne  $S^*$ 
30 Fim

```

□

6 EXPERIMENTOS COMPUTACIONAIS

Neste capítulo são descritos os experimentos computacionais realizados para avaliar a abordagem proposta para o Green-PDPTW. Os testes foram conduzidos com o objetivo de analisar o desempenho do algoritmo híbrido, composto pela Evolução

Diferencial e pela Busca Tabu, considerando métricas como número de veículos utilizados, distância total percorrida, emissões totais, valor da função objetivo e tempo de execução.

A abordagem proposta neste trabalho, baseada na combinação da meta-heurística Differential Evolution (DE) com uma estratégia de busca local, foi inspirada em uma sugestão de trabalhos futuros apresentada por Kunnapapdeelert e Klinsrisuk (2019). Como não foi possível obter contato com os autores, foi desenvolvida uma implementação própria do DE aplicada ao Green-PDPTW. Para isso, tomou-se como referência a adaptação da Evolução Diferencial para problemas combinatórios apresentada por Figueiredo, Fernandes e Zavala (2018), no contexto do Problema do Caixeiro Viajante (TSP). A partir dessa base, foram realizadas as modificações necessárias para adequar o método às características e restrições específicas do Green-PDPTW.

A função de avaliação, ou fitness, foi utilizada para comparar as soluções geradas pelo algoritmo durante o processo de busca. Cada solução S é avaliada por meio de dois componentes principais: o custo operacional e o componente ambiental. O custo operacional considera a distância total percorrida, o custo fixo associado aos veículos utilizados e a penalidade aplicada aos pedidos não atendidos.

(18)

Na equação (18), $D(S)$ representa a distância total percorrida pela solução, c_v é o custo fixo por veículo, $V(S)$ corresponde à quantidade de veículos utilizados, p_u representa a penalidade por pedido não atendido e $U(S)$ indica o número de pedidos não atendidos. O componente ambiental é calculado pela soma das emissões dos gases considerados no modelo: dióxido de carbono, metano e óxido nitroso. Assim, as emissões totais da solução são representadas pela equação 19.

(19)

A partir desses dois componentes, a função objetivo utilizada no algoritmo é definida pela equação (20). O custo base é multiplicado por uma constante elevada, denominada BigN e representada por B na equação, e depois somado ao total de emissões. Essa estratégia

faz com que o algoritmo priorize o custo operacional da solução, enquanto as emissões funcionam como critério complementar na avaliação.

(20)

Além do valor do fitness, a comparação entre soluções também considera a quantidade de veículos utilizados. Dessa forma, uma solução que utiliza menos veículos é considerada melhor. Caso duas soluções utilizem a mesma quantidade de veículos, escolhe-se aquela com menor valor de $F(S)$. Portanto, o critério de avaliação adotado busca primeiro reduzir a quantidade de veículos empregados e, em seguida, minimizar o custo operacional e as emissões geradas pelas rotas. Essa estrutura permite avaliar as soluções de forma coerente com o objetivo do Green-PDPTW, combinando aspectos econômicos e ambientais.

O algoritmo foi implementado na linguagem Python, utilizando a versão 3.12.7. Os experimentos foram executados em um computador com sistema operacional Ubuntu 24.04.3 LTS, processador AMD Ryzen Threadripper PRO 5965WX, com 48 threads, frequência de até 4,5 GHz e 140 MB de cache, além de 128 GB de memória RAM DDR4. O ambiente também conta com SSD NVMe Kingston NV2 de 2 TB e duas unidades de disco Seagate Barracuda de 8 TB. Embora o equipamento possua cinco GPUs PNY Quadro RTX A4000, cada uma com 16 GB de memória GDDR6, a implementação utilizada neste trabalho foi executada em CPU, sem uso de aceleração por GPU. O código-fonte desenvolvido neste trabalho está disponível em: <https://github.com/suehtaMFr/IC-2024-2025>.

6.1 Instâncias

As instâncias utilizadas nos experimentos foram obtidas no repositório de instâncias do PDPTW, disponibilizado pelo SINTEF. Neste trabalho, foram selecionadas as instâncias da classe LRC1, especificamente de lrc101 até lrc108. De forma geral, uma instância do PDPTW descreve o conjunto de pedidos que devem ser atendidos por uma frota de veículos. Cada linha da instância representa um nó do problema, que pode ser o depósito, um ponto de coleta ou um ponto de entrega. Para cada nó são informadas suas coordenadas no plano cartesiano (x , y), a quantidade de carga associada, o intervalo de tempo

permitido para atendimento, o tempo de serviço necessário e os índices que indicam a relação entre coleta e entrega.

Assim, a partir desses dados, o algoritmo deve construir rotas que respeitem a capacidade dos veículos, as janelas de tempo e a precedência entre coleta e entrega, garantindo que cada entrega seja realizada somente após sua coleta correspondente e pelo mesmo veículo.

6.2 Parâmetros

Na tabela 2 mostra os parâmetros utilizados na solução proposta para obter os resultados que são demonstrados nas tabelas 3 e 4. Os parâmetros utilizados no algoritmo foram definidos e calibrados por meio de testes *offline*. Durante essa etapa de experimentação, observou-se que os parâmetros mais sensíveis para o comportamento do modelo são o fator de escala (F), a taxa de cruzamento (CR), o custo fixo do veículo, além do número de iterações (*tabu_iters*) e de candidatos (*tabu_cands*) da Busca Tabu.

A variação desses valores impactou diretamente o desempenho do método nas seguintes frentes:

- **Custo Fixo do Veículo:** Este parâmetro exerce uma pressão direta sobre a função de avaliação para a minimização da frota. Se o valor atribuído for muito baixo, o algoritmo perde o incentivo para reduzir a quantidade de veículos utilizados na solução.
- **Busca Tabu (*tabu_iters* e *tabu_cands*):** O incremento desses parâmetros resulta em uma convergência inicial mais acelerada, intensificando a melhoria da qualidade da solução logo nas primeiras iterações da busca local.
- **Evolução Diferencial (F e CR):** Ambos são determinantes para ditar a velocidade de convergência geral do algoritmo, definindo a rapidez com que a qualidade das rotas melhora ao longo das gerações.

Tabela 2 - Parâmetros utilizados nos experimentos

Parâmetro	Valor	Descrição
NP	80	Tamanho da população da Evolução Diferencial.
F	0,6	Fator de escala utilizado na etapa de mutação.
CR	0,7	Taxa de cruzamento utilizada para formar a solução experimental.
Gmax	150	Número máximo de gerações do algoritmo.
penalidade_por_não_atendimento	10000,0	Penalidade aplicada para cada pedido não atendido.
custo_fixo_do_veiculo	1000,0	Custo fixo associado a cada veículo utilizado na solução.
BigN	1000000,0	Peso grande utilizado para priorizar o componente de custo na função objetivo.
tabu_iters	50	Número máximo de iterações da Busca Tabu.
tabu_tenure	Quantidade de veículos da instância	Indica que o tempo tabu é definido automaticamente pelo algoritmo.
tabu_cands	75	Quantidade de movimentos candidatos avaliados em cada iteração da Busca Tabu.
		Semente usada para controlar a aleatoriedade

6.3 Resultados

Para fins de comparação, foram selecionadas as mesmas instâncias utilizadas por Kunnappapdeelert e Klinsrisuk (2019) que trabalhou com o Grenn-PDPTW. As Tabelas 3 e 4 apresentam as comparações entre os resultados obtidos pela abordagem proposta neste trabalho. Além disso, para manter a coerência com o estudo de referência, foram utilizados os coeficientes de emissão de gases associados ao diesel como combustível, conforme descrito no artigo Kunnappapdeelert e Klinsrisuk (2019).

6.3.1 Comparações

Na tabela 3 faz a comparação dos resultados obtidos a partir da solução proposta com as melhores soluções do *data set* até o momento de desenvolvimento deste trabalho, a diferença percentual foi calculada considerando o *data set* como valor de referência. Assim, essa medida indica o quanto a solução proposta está acima ou abaixo da melhor solução conhecida até o momento para cada instância. Valores positivos indicam uma solução com distância maior, enquanto valores negativos indicam uma solução com distância menor. A base de dados foi originalmente desenvolvida para o problema PDPTW, portanto não apresenta informações relacionadas às emissões de gases de efeito estufa.

Tabela 3 - Comparação entre o data set e a solução proposta

Instância	Data Set		Solução Proposta		Diferença (%)
	Distância	Veículos	Distância	Veículos	
Irc101	1708.80	14	1755.58	8	2.74
Irc102	1558.07	12	2038.23	8	30.82
Irc103	1258.74	11	1807.04	8	43.56
Irc104	1128.40	10	1764.52	8	56.37
Irc105	1637.62	13	1837.07	9	12.18
Irc106	1424.73	11	1993.70	8	39.94
Irc107	1230.14	11	1739.78	8	41.43
Irc108	1147.43	10	1659.46	8	44.62

Fonte: elaborada pelo autor.

Já na tabela 4 mostramos a comparação dos resultados da solução proposta neste trabalho com os resultados obtidos por Kunnappapdeelert e Klinsrisuk (2019).

Tabela 4 - Comparação entre os resultados do artigo e a solução proposta

Instância	Função objetivo	Resultados do artigo					Solução proposta					Diferença (%)
		Desempenho		Emissões diesel (kg)			Desempenho		Emissões diesel (kg)			
		Dist.	Veic.	CO2	CH4	N2O	Dist.	Veic.	CO2	CH4	N2O	Dist.
Irc101	Min. custo	1764.08	14	1283.753	0.434859	9.229	1755.58	8	1416.022	0.479184	10.180	-0.48

	<i>Min. GHG</i>	1872.64	16	1123.190	0.380470	8.074	x	x	x	x	x	-6.25
Irc102	Min. custo	1783.14	14	1125.997	0.381421	8.095	2038.23	8	1650.063	0.558424	11.863	14.31
	<i>Min. GHG</i>	2194.68	17	1123.390	0.380540	8.076	x	x	x	x	x	-7.13
Irc103	Min. custo	1522.24	13	1380.089	0.467492	9.921	1807.00	8	1446.550	0.489442	10.400	18.71
	<i>Min. GHG</i>	1629.16	13	965.699	0.327120	6.942	x	x	x	x	x	-17.66
Irc104	Min. custo	1221.72	11	1815.664	0.615039	13.052	1764.52	8	1430.083	0.483988	10.281	44.43
	<i>Min. GHG</i>	1325.74	11	1638.420	0.555000	11.778	x	X	x	x	x	33.10
Irc105	Min. custo	1786.68	15	1278.679	0.433141	9.192	1837.07	9	1484.421	0.502346	10.672	2.82
	<i>Min. GHG</i>	1859.26	15	1073.940	0.363790	7.720	x	X	x	x	x	-1.19
Irc106	Min. custo	1764.00	14	1573.333	0.532952	11.310	1993.70	8	1614.931	0.546533	11.610	13.02
	<i>Min. GHG</i>	1880.24	14	1092.570	0.370100	7.854	x	x	x	x	x	6.03
Irc107	Min. custo	1732.29	14	1474.067	0.499326	10.597	1739.78	8	1406.091	0.475842	10.109	0.43
	<i>Min. GHG</i>	2233.55	16	1360.050	0.460700	9.777	x	x	x	x	x	-22.11
Irc108	Min. custo	1646.81	12	1594.691	0.540187	11.464	1659.46	8	1340.215	0.453542	9.635	0.77
	<i>Min. GHG</i>	1648.92	13	1560.560	0.528630	11.219	x	x	x	x	x	0.64

Fonte: elaborada pelo autor.

Na tabela 5 demonstra o tempo médio de execução de cada instancia pela solução proposta, cada instancia foi executada 20 vezes, não comparamos o tempo de execução da solução proposta com as do *data set* nem com o trabalho Kunnapapdeelert e Klinsrisuk (2019), pois eles não informaram o tempo de execução das suas soluções.

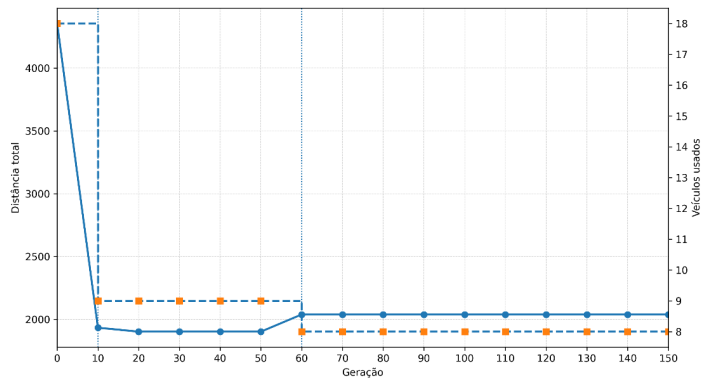
Tabela 5 - Tempo de execução por instância

Instância	Tempo de execução
Irc101	02:12:19
Irc102	02:12:22
Irc103	02:10:35
Irc104	02:11:22
Irc105	02:15:02
Irc106	02:11:17
Irc107	02:11:13
Irc108	02:07:32

Fonte: elaborada pelo autor.

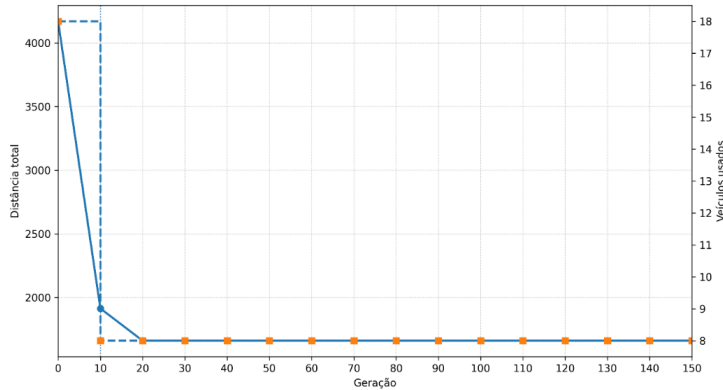
As Figuras 5 e 6 apresentam a evolução da melhor solução encontrada pela abordagem proposta para as instâncias Irc102 e Irc108, respectivamente. Em ambos os gráficos, o eixo horizontal representa as gerações do algoritmo, enquanto o eixo vertical esquerdo indica a distância total percorrida e o eixo vertical direito representa a quantidade de veículos utilizados. A linha contínua mostra a variação da distância total ao longo das gerações, enquanto a linha tracejada representa a quantidade de veículos utilizados pela solução.

Figura 5 - Convergência da solução da instância Irc102



Fonte: elaborada pelo autor.

Figura 6 - Convergência da solução da instância Irc108



Fonte: elaborada pelo autor.



7 CONCLUSÃO

Neste trabalho de conclusão de curso foi abordado o problema Green-PDPTW, uma variação do problema de roteamento de veículos com coleta e entrega e janelas de tempo que também incorpora aspectos ambientais relacionados às emissões de gases de efeito estufa. Nesse problema, busca-se construir rotas para uma frota de veículos de modo que os pedidos de coleta e entrega sejam atendidos respeitando restrições de capacidade, precedência entre coleta e entrega e janelas de tempo. Além disso, o modelo considera o impacto ambiental dos deslocamentos, calculado a partir das emissões de CO_2 , CH_4 e N_2O .

Para tratar o problema, foi proposta uma abordagem heurística híbrida baseada na combinação da Evolução Diferencial com a Busca Tabu. A escolha da Evolução Diferencial foi motivada por sua aplicação anterior ao Green-PDPTW e pela possibilidade de adaptação para problemas combinatórios. Como o DE foi originalmente desenvolvido para problemas contínuos, foi necessário

adaptar sua representação para o contexto discreto do roteamento. Para isso, as soluções foram representadas por rotas de veículos e manipuladas, durante os operadores evolutivos, por meio de uma representação auxiliar baseada nos pontos de coleta.

A etapa de mutação foi adaptada utilizando as coordenadas dos pontos de coleta, permitindo gerar novos candidatos a partir da aproximação entre uma solução aleatória e a melhor solução da geração. Em seguida, o crossover combinou rotas da solução atual e da solução mutante, sendo aplicado um procedimento de reparo para reconstruir as soluções. A Busca Tabu foi utilizada como mecanismo de melhoria local, realizando movimentos de realocação de pares coleta-entrega entre veículos, com o objetivo de refinar as soluções produzidas pela Evolução Diferencial.

A função de avaliação utilizada no algoritmo considerou dois componentes principais: o custo operacional e o componente ambiental. O custo operacional foi formado pela distância total percorrida, pelo custo fixo associado aos veículos utilizados e pela penalidade por pedidos não atendidos. Já o componente ambiental foi calculado pela soma das emissões de CO_2 , CH_4 e N_2O . Além disso, a comparação entre soluções priorizou inicialmente a redução da quantidade de veículos utilizados e, em seguida, o valor da função objetivo. Dessa forma, a abordagem buscou combinar aspectos econômicos e ambientais no processo de avaliação das soluções.

Os experimentos computacionais foram realizados com instâncias da classe LRC1, de lrc101 a lrc108. Os resultados indicaram que a abordagem proposta conseguiu gerar soluções com menor quantidade de veículos em comparação às soluções utilizadas como referência. No entanto, essa redução deve ser interpretada com cautela, pois foram identificadas violações em janelas de tempo. Assim, as soluções obtidas não podem ser consideradas plenamente viáveis em relação a todas as restrições do Green-PDPTW. A menor quantidade de veículos observada está possivelmente associada a essa violação das restrições temporais, uma vez que o atendimento fora das janelas de tempo permite a construção de rotas mais compactas.

Em relação às melhores soluções conhecidas até o

momento de desenvolvimento do trabalho, a solução proposta apresentou, em geral, maiores distâncias percorridas, embora tenha utilizado menos veículos. Na comparação com os resultados de Kunnapapdeelert e Klinsrisuk, algumas instâncias apresentaram distâncias próximas ou inferiores aos resultados reportados no artigo. Entretanto, devido à ocorrência de violações nas janelas de tempo, esses resultados não devem ser interpretados como superiores às soluções da literatura, mas sim como indicativos do comportamento da abordagem proposta e dos pontos que ainda precisam ser aprimorados.

De modo geral, os resultados mostram que a hibridização entre Evolução Diferencial e Busca Tabu apresenta potencial para o tratamento do Green-PDPTW, principalmente por permitir a combinação entre critérios operacionais e ambientais. No entanto, a abordagem ainda necessita de melhorias no controle de viabilidade das soluções, especialmente no tratamento das janelas de tempo. A ausência de um mecanismo mais rigoroso para penalizar ou reparar violações temporais fez com que o algoritmo favorecesse soluções com menos veículos, mas que não respeitam completamente as restrições do problema.

Como trabalhos futuros, sugere-se investigar o uso de estratégias de controle adaptativo de parâmetros, nas quais os valores dos parâmetros do algoritmo são ajustados dinamicamente durante a execução. Também podem ser exploradas outras estratégias de busca local, além de abordagens multiobjetivo que tratem separadamente distância, número de veículos e emissões. Outra possibilidade consiste em investigar novas formas de adaptação da Evolução Diferencial ao espaço discreto.

□

REFERÊNCIAS

ACEA - EUROPEAN AUTOMOBILE MANUFACTURERS' ASSOCIATION. Vehicles on European Roads 2025. Brussels: **ACEA**, 2025. Disponível em: <https://www.acea.auto/publication/report-vehicles-on-european-roads-2025/>. Acesso em: 12 fev. 2025. Acesso em: 15 out. 2025.

ALMEIDA, Ricardo; SERRANO, Luis; SILVA, Diogo;

SANTOS, Helder; PEREIRA, João; SILVA, Manuel Gameiro da. Decarbonization of EU Road Freight Transport in the Short and Medium Term Through Renewable Liquid Fuels—A Review. **Energies**, v. 19, n. 5, artigo 1382, 2026. DOI: 10.3390/en19051382. Acesso em: 15 mar. 2025. Acesso em: 23 abr. 2025.

ANAND, H. S.; VINOD CHANDRA, S. S. Nature inspired meta heuristic algorithms for optimization problems. *Computing*, v. 104, p. 251-269, 2022. DOI: 10.1007/s00607-021-00955-5. Acesso em: 9 abr. 2024.

BALINSKI, M. L.; QUANDT, R. E. On an integer program for a delivery problem. **Operations Research**, v. 12, n. 2, p. 300-304, 1964. DOI: 10.1287/opre.12.2.300. Acesso em: 9 abr. 2025.

BERRAHMANIA, N.; HAMMOU, E. H.; EL FADI, L. Short review in green vehicle routing problem with pick-up and delivery and time windows using metaheuristics. **The Open Transportation Journal**, v. 19, 2025. DOI: 10.2174/0126671212375850250212065147. Acesso em: 9 jun. 2025.

CACCETTA, L.; ALAMEEN, M.; ABDUL-NIBY, M. An improved Clarke and Wright algorithm to solve the capacitated vehicle routing problem. **Engineering, Technology & Applied Science Research**, v. 3, n. 2, p. 413-415, 2013. DOI: [10.48084/etasr.292](https://doi.org/10.48084/etasr.292). Acesso em: 10 abr. 2024.

DANTZIG, G. B.; FULKERSON, R.; JOHNSON, S. M. Solution of a large-scale traveling-salesman problem. **Operations Research**, v. 2, n. 4, p. 393-410, 1954. DOI: 10.1007/978-3-540-68279-0_1. Acesso em: 9 jan. 2026.

DAS, S.; SUGANTHAN, P. N. Differential evolution: a survey of the state-of-the-art. **IEEE Transactions on Evolutionary Computation**, v. 15, n. 1, p. 4-31, 2011. DOI: [10.1109/TEVC.2010.2059031](https://doi.org/10.1109/TEVC.2010.2059031). Acesso em: 14 set. 2025.

DENG, F.; QIN, H.; LI, J.; CHENG, C. The pickup and delivery problem with time windows and incompatibility constraints in cold chain transportation. **Transportation Science**, v. 57, n. 2, p. 444-462, 2023. DOI: 10.1287/trsc.2022.1167. Acesso em: 10 out. 2025.

FIGUEIREDO, D. D.; FERNANDES, C. R.; ZAVALA, A. B. P. Problema do Caixeiro Viajante abordado pelo Algoritmo de Evolução Diferencial. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 6, n. 2, 2018. DOI: 10.5540/03.2018.006.02.0314. Acesso em: 20 mai. 2024.

GENDREAU, M.; HERTZ, A.; LAPORTE, G. A tabu search heuristic for the vehicle routing problem. **Management Science**, v. 40, n. 10, p. 1276-1290, 1994. DOI: 10.1287/mnsc.40.10.1276. Acesso em: 12 abr. 2024.

GLOVER, F. Tabu search — Part I. **ORSA Journal on Computing**, v. 1, n. 3, p. 190-206, 1989. DOI: 10.1287/ijoc.1.3.190. Acesso em: 9 nov. 2024.

GÓMEZ VILCHEZ, Jonatan J.; JULEA, Andreea; LODI, Chiara; MAROTTA, Alessandro. An analysis of trends and policies supporting alternative fuels for road freight transport in Europe. **Frontiers in Energy Research**, v. 10, artigo 897916, 2022. DOI: 10.3389/fenrg.2022.897916. Acesso em: 15 fev. 2025.

GONG, W.; CAI, Z. Differential evolution with ranking-based mutation operators. **IEEE Transactions on Cybernetics**, v. 43, n. 6, p. 2066-2081, 2013. DOI: 10.1109/TCYB.2013.2239988. Acesso em: 16 mai. 2025.

GONZALEZ, M. et al. A hyper-matheuristic approach for solving mixed integer linear optimization models in the context of data envelopment analysis, **PeerJ Computer Science**, 2022. Disponível em: [10.7717/peerj-cs.828](https://doi.org/10.7717/peerj-cs.828). Acesso em: 11 abr. 2024.

HORNSTRA, R. P. et al. The vehicle routing problem with simultaneous pickup and delivery and handling costs. **Computers & Operations Research**, v. 115, 2020. DOI: 10.1016/j.eswa.2022.119118. Acesso em: 9 abr. 2026.

HUSSAIN, K.; MOHD SALLEH, M. N.; CHENG, S.; SHI, Y. Metaheuristic research: a comprehensive survey. **Artificial Intelligence Review**, v. 52, n. 4, p. 2191-2233, 2019. DOI: 10.1007/s10462-017-9605-z. Acesso em: 13 abr. 2024. Acesso em: 13 abr. 2024.

KARP, R. M. Reducibility among combinatorial problems. In: MILLER, R. E.; THATCHER, J. W. (org.). **Complexity of Computer Computations**. New York: Plenum Press, 1972. p. 85-103. DOI: 10.1007/978-3-540-68279-0_8. Acesso em: 9 abr. 2025.

KUNNAPAPDEELERT, S.; KLINSRISUK, R. Determination of green vehicle routing problem via differential evolution. **International Journal of Logistics Systems and Management**, v. 34, n. 3, p. 395-410, 2019. Acesso em: 15 abr. 2024

LAI, M.; CAO, E. An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows. **Engineering Applications of Artificial Intelligence**, v. 23, n. 2, p. 188-195, 2010. DOI: 10.1504/IJLSM.2019.103091. Acesso em: 22 abr. 2026.

LAU, H. C.; WAN, W. C.; HALIM, S. A software framework for fast prototyping of meta-heuristics hybridization. **International Transactions in Operational Research**, v. 14, n. 2, p. 123-141, 2007. DOI: [10.1111/j.1475-3995.2007.00578.x](https://doi.org/10.1111/j.1475-3995.2007.00578.x). Acesso em: 13 abr. 2024.

LENSTRA, J. K.; RINNOOY KAN, A. H. G. Complexity of vehicle routing and scheduling problems. **Networks**, v. 11, n. 2, p. 221-227, 1981. DOI: 10.1002/net.3230110211. Acesso em: 9 abr. 2026.

LI, H.; LIM, A. A metaheuristic for the pickup and delivery problem with time windows. In: Proceedings of the 13th **IEEE International Conference on Tools with Artificial Intelligence**. Dallas: IEEE, 2001. DOI: 10.1109/ICTAI.2001.974461. Acesso em: 14 abr. 2025.

MOUNIA, D. A.; DJEBBAR, B. A hybrid discrete artificial bee colony for the green pickup and delivery problem with time windows. **Informatica**, v. 44, n. 4, p. 507-519, 2020. DOI: 10.31449/inf.v44i4.3000. Acesso em: 28 abr. 2025.

NANRY, W. P.; BARNES, J. W. Solving the pickup and delivery problem with time windows using reactive tabu search. **Transportation Research Part B: Methodological**, v. 34, n. 2, p. 107-121, 2000. DOI: 10.1016/S0191-2615(99)00016-8. Acesso em: 27 mai.

2025.

OPARA, K. R.; ARABAS, J. Differential evolution: a survey of theoretical analyses. **Swarm and Evolutionary Computation**, v. 44, p. 546-558, 2019. DOI: 10.1016/j.swevo.2018.06.010. Acesso em: 15 fev. 2026.

QIN, A. K.; HUANG, V. L.; SUGANTHAN, P. N. Differential evolution algorithm with strategy adaptation for global numerical optimization. **IEEE Transactions on Evolutionary Computation**, v. 13, n. 2, p. 398-417, 2009. DOI: 10.1109/TEVC.2008.927706. Acesso em: 19 jan. 2026.

SARTORI, C. S.; BURIOL, L. S. A study on the pickup and delivery problem with time windows: matheuristics and new instances. **Computers & Operations Research**, v. 124, 105065, 2020. DOI: 10.1016/j.cor.2020.105065. Acesso em: 26 dez. 2025.

SINTEF. Pickup and delivery problem with time windows benchmark. Disponível em: <https://www.sintef.no/projectweb/top/pdptw/>. Acesso em: 15 abr. 2024.

STORN, R. Differential evolution research: trends and open questions. In: **CHAKRABORTY, U. K.** (ed.). *Advances in Differential Evolution*. Berlin: Springer, 2008. DOI: 10.1007/978-3-540-68830-3_1. Acesso em: 29 jul. 2025.

STORN, R.; PRICE, K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, p. 341-359, 1997. DOI: 10.1023/A:1008202821328. Acesso em: 12 abr. 2024.

WANG, Z.; DESSOUKY, M.; VAN WOENSEL, T. Pickup and delivery problem with hard time windows considering stochastic and time-dependent travel times. **EURO Journal on Transportation and Logistics**, v. 12, 100099, 2023. DOI: 10.1016/j.ejtl.2022.100099. Acesso em: 12 abr. 2025.

