57º SBPO
SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
Gramado • RS
DE 5 A 9 DE OUTUBRO DE 2025

Resolução do Problema de Roteamento de Veículos Verde com Coletas e Entregas Simultâneas via Redução de Instâncias e Busca em Vizinhança com Profundidade Variável

Victor Tiezzi Henriques, Napoleão Nepomuceno
Universidade de Fortaleza, Programa de Pós-Graduação em Informática Aplicada
Av. Washigton Soares 1321, CEP: 60811-905, Fortaleza, Ceará, Brasil
victor.tiezzi@edu.unifor.br,napoleaovn@unifor.br

RESUMO

Este trabalho apresenta uma abordagem baseada em Busca em Vizinhança com Profundidade Variável para a resolução do Problema de Roteamento de Veículos Verde com Coletas e Entregas Simultâneas. A proposta explora subinstâncias reduzidas, modeladas em Programação Linear Inteira e resolvidas com auxílio de um *solver* exato. São incorporadas estratégias que orientam a definição da vizinhança de uma solução, com base em critérios de proximidade entre clientes e no uso de múltiplas soluções previamente encontradas. Essas estratégias promovem um equilíbrio entre intensificação e diversificação da busca. Experimentos computacionais em instâncias de *benchmark* demonstram que a abordagem proposta supera métodos existentes em termos de qualidade e robustez das soluções, especialmente em problemas de maior escala. A avaliação individual das estratégias adotadas confirma seu impacto positivo na eficácia do método.

PALAVRAS CHAVE. Problema de Roteamento de Veículo, Busca em Vizinhança, Programação Linear Inteira.

Tópicos (L&T – Logística e Transportes, MH – Metaheurísticas)

ABSTRACT

This paper presents a Variable-Depth Neighborhood Search approach for solving the Green Vehicle Routing Problem with Simultaneous Pickup and Delivery. The method explores reduced subinstances modeled as Integer Linear Programs and solved by an exact solver. Strategies are incorporated to guide the neighborhood definition, based on customer proximity and the reuse of multiple previously found solutions. These mechanisms foster a balance between intensification and diversification. Computational experiments on benchmark instances demonstrate that the proposed approach outperforms existing methods in both solution quality and robustness, particularly on large-scale problems. An ablation study confirms the positive impact of the adopted strategies on overall performance.

KEYWORDS. Vehicle Routing Problem, Neighborhood Search, Integer Linear Programming.

Paper topics (L&T – Logistics and Transportation, MH – Metaheuristics)

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

1. Introdução

A crescente preocupação com os impactos ambientais das atividades econômicas tem impulsionado transformações relevantes na gestão logística. No setor de transportes, um dos principais responsáveis pela emissão de gases de efeito estufa, observa-se a busca por soluções que aliem eficiência operacional à responsabilidade ambiental [McKinnon, 2018]. Nesse contexto, ganham destaque estratégias como a redução no uso de combustíveis fósseis, a adoção de fontes de energia alternativas e a incorporação de práticas de logística reversa nos fluxos de distribuição [Rogers e Tibben-Lembke, 2001; Dekker et al., 2012]. A logística reversa, além de reduzir resíduos, viabiliza o reaproveitamento de recursos e impõe novos desafios ao planejamento de rotas, exigindo a integração de coletas e entregas em percursos eficientes e sustentáveis.

O Problema de Roteamento de Veículos (VRP, do inglês *Vehicle Routing Problem*) é um problema clássico de otimização combinatória que visa determinar rotas eficientes para uma frota de veículos atender a um conjunto de clientes geograficamente distribuídos, respeitando restrições operacionais. Dada sua ampla aplicabilidade, o VRP possui diversas extensões que capturam especificidades de diferentes contextos logísticos. Entre essas, destaca-se o Problema de Roteamento de Veículos com Coletas e Entregas Simultâneas (VRPSPD, do inglês *Vehicle Routing Problem with Simultaneous Pickup and Delivery*), originalmente formulado por Min [1989]. Nesse problema, cada cliente demanda simultaneamente serviços de coleta e entrega, os quais devem ser realizados por um único veículo em uma única visita. Essa característica torna o VRPSPD especialmente relevante em sistemas de logística reversa [Dethloff, 2001].

A incorporação de critérios ambientais ao planejamento logístico motivou o desenvolvimento de variantes do VRP com enfoque sustentável. O Problema de Roteamento de Veículos Verde (G-VRP, do inglês *Green Vehicle Routing Problem*) [Erdoğan e Miller-Hooks, 2012; Asghari e Mirzapour Al-e-hashem, 2021] é uma dessas extensões, considerando fatores como o consumo de combustível, as emissões de CO_2 e o uso de veículos movidos a combustíveis alternativos. Dentre as variantes do G-VRP, destaca-se o Problema de Roteamento de Veículos Verde com Coletas e Entregas Simultâneas (G-VRPSPD, do inglês *Green Vehicle Routing Problem with Simultaneous Pickup and Delivery*), que combina as particularidades do VRPSPD com objetivos ambientais, penalizando o consumo proporcional à carga transportada [Huang et al., 2012; Olgun et al., 2021].

Este trabalho investiga o aprimoramento de um algoritmo de Busca em Vizinhança com Profundidade Variável (VDNS, do inglês *Variable-Depth Neighborhood Search*) [Ahuja et al., 2002; Altner et al., 2014], originalmente proposto por Barroso e Nepomuceno [2024] para o VRPSPD, e sua adaptação à resolução do G-VRPSPD. A vizinhança de uma solução viável é definida por um modelo em Programação Linear Inteira (PLI) e explorada por meio de um solucionador exato, seguindo abordagens híbridas inspiradas na redução da instância do problema [Blum e Raidl, 2016], como os métodos *Generate & Solve* [Saraiva et al., 2013], *Construct, Merge, Solve & Adapt* [Blum et al., 2016] e *Relaxation Induced Neighborhood Search* [Danna et al., 2005]. As mesmas instâncias de *benchmark* utilizadas em Olgun et al. [2021] são adotadas neste estudo.

O restante do artigo está organizado da seguinte forma. A Seção 2 revisa os principais trabalhos sobre o VRPSPD e o G-VRPSPD. A Seção 3 formaliza o problema, apresenta a estrutura de vizinhança e descreve as estratégias incorporadas ao VDNS. A Seção 4 relata os experimentos em instâncias *benchmark* e discute os resultados obtidos. Por fim, a Seção 5 resume as contribuições e aponta direções futuras.

2. Trabalhos Relacionados

O VRPSPD, proposto por Min [1989], ganhou notoriedade por sua aplicabilidade em sistemas de logística reversa, onde coletas e entregas ocorrem na mesma visita ao cliente [Koç et al., 2020]. Essa estrutura permite uma utilização mais eficiente da frota, sendo aplicada com sucesso em setores como bebidas, alimentos e produtos recondicionados [Santos et al., 2023].

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

Diversas estratégias foram desenvolvidas para resolver o VRPSPD. Entre as heurísticas, destacam-se os métodos de inserção por agrupamento [Salhi e Nagy, 1999], busca tabu [Montané e Galvão, 2006], colônia de formigas [Gajpal e Abad, 2009], algoritmos genéticos [Tasan e Gen, 2012] e busca local iterativa [Subramanian et al., 2010]. Mais recentemente, foram propostas metaheurísticas híbridas que combinam componentes construtivos com mecanismos de intensificação, como em [Goksal et al., 2013; Avci e Topaloglu, 2016; Nepomuceno et al., 2023; Barroso e Nepomuceno, 2024]. Métodos exatos [Dell'Amico et al., 2006; Subramanian et al., 2013] também foram investigados na literatura sobre o VRPSPD.

O G-VRPSPD estende o VRPSPD ao incorporar critérios de sustentabilidade. Embora ainda sejam escassos os trabalhos dedicados especificamente a essa variante, há uma literatura mais ampla sobre problemas de roteamento com objetivos ambientais. Nessas outras variantes do VRP, o consumo de combustível e as emissões de CO_2 têm sido modelados como funções da carga transportada, velocidade, topografia, aceleração e frenagem do veículo [Demir et al., 2011; Alinaghian e Naderipour, 2016; Franceschetti et al., 2017]. Outras abordagens consideram o uso de frotas compostas por veículos elétricos ou movidos a combustíveis alternativos, sujeitas a restrições de autonomia e recarga [Erdoğan e Miller-Hooks, 2012; Macrina et al., 2019].

Apesar da relevância ambiental do problema, aspectos como pegada de carbono e emissões de gases de efeito estufa ainda são pouco explorados na literatura dedicada ao VRPSPD. Huang et al. [2012] apresentaram uma das primeiras formulações matemáticas para o G-VRPSPD. Mais recentemente, Olgun et al. [2021] avançaram na modelagem do problema ao propor uma formulação matemática e um algoritmo hiper-heurístico (HH-ILS), baseado em busca local iterativa e descida por vizinhança variável. Os autores conduziram experimentos para avaliar o impacto da função objetivo ambiental sobre o consumo total de combustível, comparando os modelos G-VRPSPD e VRPSPD. Os resultados indicam que a inclusão de critérios ambientais na otimização tem efeito significativo na redução dos custos operacionais.

3. Metodologia

Nesta seção, apresenta-se uma formulação em PLI para o G-VRPSPD, adaptada de [Olgun et al., 2021]. Em seguida, revisita-se a estrutura de vizinhança de profundidade variável proposta por [Barroso e Nepomuceno, 2024] e propõe-se um algoritmo VDNS para resolver o problema, com ênfase no aprimoramento do mecanismo de geração de vizinhanças promissoras.

3.1. Formulação Matemática

A formulação matemática adotada neste trabalho segue a proposta de Olgun et al. [2021], que combina uma modelagem baseada em fluxo com uma função de consumo de combustível dependente da carga transportada e da distância percorrida. O problema é definido sobre um grafo dirigido completo $G = (N, A)$, onde $N = \{0, 1, \ldots, n\}$ representa o conjunto de nós (sendo o depósito o nó 0 e os demais os clientes), e A o conjunto de arcos. O conjunto de clientes é denotado por $N_c = N \setminus \{0\}$. Cada cliente $i \in N_c$ possui uma demanda de entrega d_i e uma demanda de coleta p_i. A distância entre os nós i e j é representada por c_{ij}, e c_0 é a constante associada ao custo unitário de combustível. A capacidade do veículo é dada por Q, sendo $Q_D = \sum_{i \in N_c} d_i$ e $Q_P = \sum_{i \in N_c} p_i$ os totais de entrega e coleta, respectivamente. A taxa de consumo de combustível (FCR, do inglês *Fuel Consumption Rate*) por unidade de distância é modelada como uma função linear da carga transportada. Sejam p^* e p_0 os valores da FCR com o veículo totalmente carregado e vazio, respectivamente. Define-se o coeficiente $\alpha = (p^* - p_0)/Q$, permitindo expressar o consumo no arco (i, j) como $p_0 + \alpha \cdot q_{ij}$, onde q_{ij} é a carga total transportada. Definem-se as variáveis de decisão: $x_{ij} \in \{0, 1\}$: indica se o arco (i, j) é utilizado por algum veículo; $U_{ij} \geq 0$: quantidade de carga de entrega transportada ao longo do arco (i, j); e $V_{ij} \geq 0$: quantidade de carga de coleta transportada ao longo do arco (i, j). O problema pode ser formulação em PLI como:

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

$$\min \quad \sum_{i \in N} \sum_{j \in N} c_0 \cdot c_{ij} \cdot (p_0 \cdot x_{ij} + \alpha \cdot (U_{ij} + V_{ij})) \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in N} x_{ij} = 1 \qquad \forall j \in N_c \tag{2}$$

$$\sum_{j \in N} x_{ij} = 1 \qquad \forall i \in N_c \tag{3}$$

$$U_{ij} + V_{ij} \leq M_{ij} \cdot x_{ij} \qquad \forall i, j \in N \tag{4}$$

$$\sum_{j \in N_c} U_{0j} = Q_D \tag{5}$$

$$\sum_{i \in N_c} U_{i0} = 0 \tag{6}$$

$$\sum_{i \in N_c} V_{i0} = Q_P \tag{7}$$

$$\sum_{j \in N_c} V_{0j} = 0 \tag{8}$$

$$U_{ij} \leq (Q - d_i) \cdot x_{ij} \qquad \forall i \in N_c, j \in N \tag{9}$$

$$V_{ij} \leq (Q - p_j) \cdot x_{ij} \qquad \forall i \in N, j \in N_c \tag{10}$$

$$U_{ij} \geq d_j \cdot x_{ij} \qquad \forall i \in N, j \in N_c \tag{11}$$

$$V_{ij} \geq p_i \cdot x_{ij} \qquad \forall i \in N_c, j \in N \tag{12}$$

$$\sum_{i \in N} V_{ji} - \sum_{i \in N} V_{ij} = p_j \qquad \forall j \in N_c \tag{13}$$

$$\sum_{i \in N} U_{ij} - \sum_{i \in N} U_{ji} = d_j \qquad \forall j \in N_c \tag{14}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in N \tag{15}$$

$$0 \leq U_{ij}, V_{ij} \leq Q \qquad \forall i, j \in N \tag{16}$$

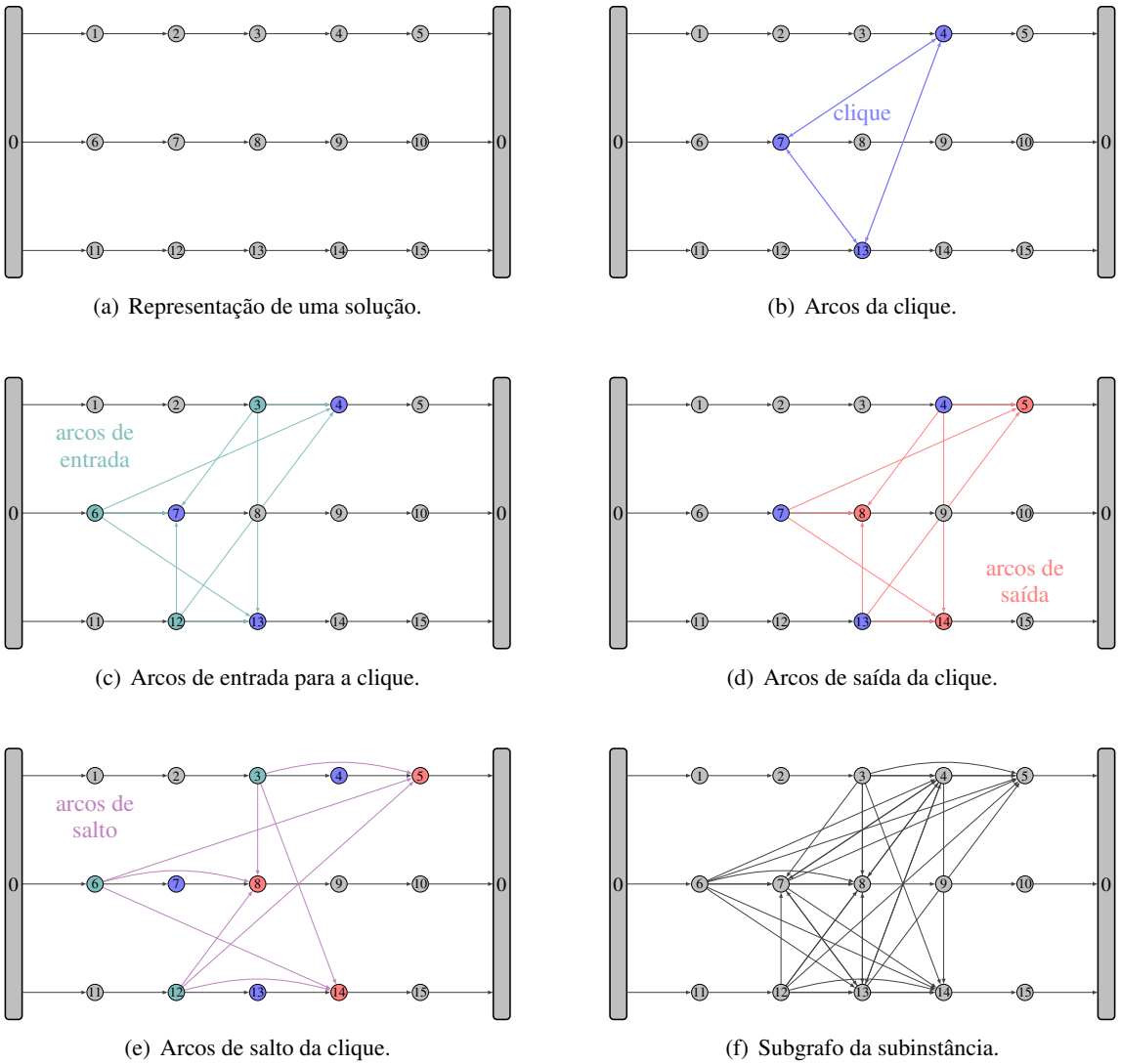
A função objetivo (1) minimiza o consumo total de combustível. As restrições (2) e (3) garantem que cada cliente seja visitado exatamente uma vez. A restrição (4) assegura que apenas arcos ativados transportem carga, utilizando o parâmetro $M_{ij} = Q - \max\{0, d_i - p_i, p_j - d_j\}$, conforme proposto por Rieck e Zimmermann [2013]. As restrições (5)–(8) garantem o fluxo correto de entregas e coletas no depósito. As restrições (9)–(12) limitam a carga transportada com base na capacidade do veículo e nas demandas dos clientes. As restrições (13) e (14) asseguram o equilíbrio de fluxo em cada cliente. Por fim, as restrições (15) e (16) definem o domínio das variáveis. Cabe destacar que, embora algumas formulações do problema — como a apresentada por Olgun et al. [2021] — incluam uma restrição explícita sobre o número máximo de veículos, essa limitação não foi adotada na prática pelo algoritmo HH-ILS [Olgun et al., 2021]. De forma similar, este trabalho assume uma frota ilimitada.

3.2. Estrutura de Vizinhança

A estrutura de vizinhança proposta neste trabalho segue a proposta de Barroso e Nepomuceno [2024], baseada na construção de subinstâncias reduzidas a partir de uma solução viável s. Cada solução é composta por rotas que partem e retornam ao depósito (nó 0), visitando conjuntos disjuntos de clientes (Figura 1(a)). A construção da vizinhança segue um procedimento sistemático,

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

guiado por um parâmetro de profundidade κ, que define o número de clientes selecionados para compor uma clique.



(a) Representação de uma solução.

(b) Arcos da clique.

(c) Arcos de entrada para a clique.

(d) Arcos de saída da clique.

(e) Arcos de salto da clique.

(f) Subgrafo da subinstância.

Figura 1: Etapas de construção da subinstância a partir da solução original.

Inicialmente, escolhe-se um subconjunto de κ clientes e são adicionados à subinstância todos os arcos entre eles, formando uma clique (Figura 1(b)). Em seguida, identificam-se os nós de entrada — vértices que precedem imediatamente algum nó da clique em suas rotas originais — e os nós de saída — vértices que os sucedem. Três conjuntos adicionais de arcos são então incorporados à subinstância: (i) os arcos entre os nós de entrada e os nós da clique, chamados de arcos de entrada (Figura 1(c)); (ii) os arcos entre os nós da clique e os nós de saída, chamados de arcos de saída (Figura 1(d)); e (iii) os arcos diretos entre nós de entrada e nós de saída, denominados arcos de salto (Figura 1(e)). A inclusão desses arcos visa permitir rearranjos mais amplos e expressivos nas rotas originais, expandindo significativamente o espaço de busca explorado pelo algoritmo.

O subgrafo resultante define a vizinhança $\eta(s)$ da solução atual s, como ilustrado na Figura 1(f). Quando apenas os arcos da solução original são considerados, o espaço de busca se

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

restringe à própria solução s. À medida que novos arcos são adicionados, o conjunto de soluções viáveis se expande exponencialmente. Caso todos os arcos do grafo original sejam incluídos, a vizinhança passa a englobar a instância completa do problema, permitindo sua resolução exata — o que, no entanto, é impraticável para instâncias de grande porte.

3.3. Algoritmo de VDNS

O algoritmo proposto constrói subinstâncias reduzidas do problema a partir de uma estrutura de vizinhança parametrizada, exploradas por meio de técnicas exatas de otimização e refinamento heurístico. Embora compartilhe a estrutura geral do VDNS apresentado por Barroso e Nepomuceno [2024], este trabalho introduz avanços importantes em duas frentes principais: (i) a estratégia de seleção dos clientes que compõem a clique, visando melhorar a exploração do espaço de busca; e (ii) a construção mais criteriosa das subinstâncias, baseada em múltiplas soluções correlacionadas, ampliando a diversificação sem comprometer a resolubilidade do subproblema.

O Algoritmo 1 apresenta um resumo do procedimento. O processo inicia-se com a geração de uma solução viável s^* por meio de uma heurística gulosa baseada no vizinho mais próximo [Nepomuceno et al., 2019]. Essa solução é refinada com operadores clássicos de busca local — *2-opt*, *insertion*, *interchange* e *crossover* [Gajpal e Abad, 2009] —, com o objetivo de melhorar sua qualidade antes do início do processo iterativo. A solução refinada é armazenada em um conjunto de soluções S, que será utilizado como base para a construção das vizinhanças nas iterações seguintes.

Algoritmo 1: Pseudocódigo do algoritmo VDNS.

Criar um conjunto de soluções $S = \emptyset$;
Gerar uma solução viável s^* e refiná-la com busca local;
Adicionar s^* ao conjunto de soluções S;
while *critério de parada não satisfeito* **do**
  Selecionar κ clientes para a clique com base em histórico e proximidade;
  Definir os arcos da subinstância como $\mathcal{A} = \bigcup_{s \in S} \eta(s)$;
  Gerar o modelo μ induzido por $\mathcal{A}$;
  Resolver o modelo μ com tempo limite τ_{solver};
  Seja γ o *gap* de otimalidade retornado;
  Ajustar o tamanho da clique κ com base em γ;
  Seja L o conjunto das λ melhores soluções encontradas;
  Refinar cada solução $s \in L$ por busca local;
  Atualizar s^* com a melhor solução em $S \cup L$;
  Atualizar o conjunto de soluções $S = \{s^*\} \cup L$;
end
return s^*;

A cada iteração do VDNS, seleciona-se um subconjunto de κ clientes para compor a clique, a partir da qual será definida a subinstância a ser resolvida. Diferentemente da abordagem de Barroso e Nepomuceno [2024], em que os clientes eram escolhidos aleatoriamente, esta proposta adota uma estratégia sistemática com três objetivos: (i) evitar a repetição de clientes já selecionados, promovendo melhor cobertura do espaço de busca ao longo da execução; (ii) favorecer a seleção de clientes geograficamente próximos, com base em uma roleta ponderada inversamente proporcional à distância em relação a um cliente de referência; e (iii) garantir diversidade, reinicializando a lista de candidatos sempre que todos os clientes tiverem sido utilizados. Essas estratégias buscam compensar o número reduzido de iterações típicas de métodos baseados em subinstâncias, propor-

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

cionando melhor exploração do espaço global de soluções. A subinstância é construída a partir da união das vizinhanças $\eta(s)$ de cada solução $s \in S$:

$$\mathcal{A} = \bigcup_{s \in S} \eta(s)$$

A construção da subinstância é um dos elementos centrais do algoritmo. Em Barroso e Nepomuceno [2024], a vizinhança era definida a partir de uma única solução atual, que também era fornecida como ponto de partida ao *solver* (*warm start*), favorecendo a convergência do *gap* de otimalidade da subinstância. Neste trabalho, adota-se uma abordagem distinta: a solução atual não é fornecida ao *solver*, permitindo uma exploração mais livre da subinstância e favorecendo a geração de um conjunto mais diverso de soluções. Ao final da resolução da subinstância, respeitando-se um limite de tempo τ_{solver}, selecionam-se as λ melhores soluções, que são refinadas por busca local e complementadas pela melhor solução da execução, atualizando o conjunto S de soluções a ser utilizado na próxima iteração. Como essas soluções são obtidas a partir da mesma subinstância, tendem a ser estruturalmente próximas entre si, o que permite explorar vizinhanças amplificadas sem provocar explosão combinatória.

A cada iteração, o valor de κ é ajustado dinamicamente com base no *gap* de otimalidade γ retornado pelo *solver*: se $\gamma \leq 1\%$, o tamanho da clique é aumentado em uma unidade; caso contrário, é reduzido. Essa adaptação busca equilibrar a profundidade da vizinhança com o tempo disponível para resolução do subproblema, permitindo ao *solver* explorá-lo suficientemente bem em cada iteração. O processo se repete até que o tempo total de execução τ_{vdns} seja atingido.

4. Resultados

Para avaliar as potencialidades do método proposto, foram realizados experimentos computacionais em instâncias de *benchmark* da literatura. Os algoritmos foram implementados em Java 23.0.1 e executados em um processador Intel Core i7-14650HX (5.2 GHz) com 64 GB de RAM DDR5. A resolução dos modelos em PLI foi feita com o *solver* IBM ILOG CPLEX 22.1.1.0. Foram utilizadas instâncias *benchmark* da literatura para o VRPSPD, extraídas de [Salhi e Nagy, 1999], conhecidas como CMTX e CMTY, com número de clientes variando entre 50 e 199. Os algoritmos avaliados incluem o HH-ILS [Olgun et al., 2021], o VDNS Original [Barroso e Nepomuceno, 2024] e a versão aprimorada do VDNS proposta neste trabalho. Os valores de $\tau_{vdns} = 3.600\ s$, $\tau_{solver} = 30\ s$ e $\lambda = 5$.

4.1. Avaliação das Estratégias de Construção e Diversificação

Antes de realizar os experimentos, foi conduzido um estudo para avaliar o impacto das diferentes estratégias de construção da clique e formação da subinstância no desempenho do VDNS. A Tabela 1 resume os resultados obtidos com as combinações das três estratégias propostas:

- **Roleta:** favorece a escolha de clientes próximos na construção da clique, com base em uma roleta inversamente proporcional à distância.
- **Cobertura:** impede a repetição de clientes na clique até que todos tenham sido utilizados, promovendo uma melhor exploração do espaço de busca.
- **Multisolução:** constrói a subinstância a partir da união dos arcos de múltiplas soluções correlacionadas, em vez de uma única solução, o que permite um equilíbrio entre diversificação e resolubilidade.

Cada linha da tabela representa uma configuração distinta de estratégias ativadas (✓) ou desativadas (✗). As colunas indicam o melhor valor e a média das soluções encontradas em 10

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

Tabela 1: Resultados das estratégias de construção e diversificação do VDNS

				CMT11X		CMT4X		CMT5X	
Config.	Roleta	Cobert.	Multisol.	Melhor	Média	Melhor	Média	Melhor	Média
A	✗	✗	✗	1422,93	1460,63	1417,33	1450,66	1777,61	1807,38
B	✓	✗	✗	1419,41	1445,31	1406,93	1432,36	1755,29	1785,59
C	✗	✓	✗	1421,60	1436,91	1409,11	1444,25	1749,66	1796,35
D	✗	✗	✓	1419,12	1419,13	1392,54	1393,40	1704,23	1706,23
E	✓	✓	✗	1421,44	1450,62	1423,20	1448,82	1742,46	1783,88
F	✓	✗	✓	1419,12	1419,13	1392,54	1394,31	1704,56	1705,65
G	✗	✓	✓	1419,12	1419,13	1392,54	1393,34	1704,38	1705,91
H	✓	✓	✓	1419,12	1419,12	1392,54	1393,78	1704,38	1706,69

Legenda: ✓: Estratégia ativada ✗: Estratégia desativada

execuções. Os experimentos foram conduzidos em três instâncias do conjunto de *benchmarks*: CMT11X (120 clientes), CMT4X (150 clientes) e CMT5X (199 clientes).

Os resultados indicam que a estratégia de multisolução (configuração D) apresenta o maior impacto isolado na melhoria da qualidade das soluções, com reduções substanciais nos custos médios em todas as instâncias testadas. A estratégia de roleta (configuração B) também contribui positivamente, enquanto a estratégia de cobertura (configuração C) apresentou ganhos mais discretos, mas ainda superiores à ausência total de estratégias (A). A combinação entre as estratégias gera resultados progressivamente melhores. A configuração H, que ativa as três estratégias simultaneamente, obteve os melhores resultados médios em duas das três instâncias e manteve desempenho robusto mesmo na maior delas (CMT5X), confirmando o caráter complementar das abordagens. Por essa razão, a configuração H foi adotada como padrão nas avaliações posteriores.

4.2. Avaliação Comparativa

A Tabela 2 apresenta uma comparação entre três métodos: o HH-ILS (conforme descrito em [Olgun et al., 2021] e reimplementado neste trabalho), o VDNS original proposto por Barroso e Nepomuceno [2024], e a versão aprimorada desenvolvida neste estudo (VDNS proposto). Para cada instância, são reportados o melhor valor obtido e a média das soluções ao longo de 30 execuções.

Os resultados evidenciam que o VDNS proposto apresenta desempenho robusto e consistentemente superior nas instâncias avaliadas. Em nenhuma delas ele foi superado por outro método, seja em termos do melhor valor obtido ou da média das soluções. Em parte das instâncias, houve empate com outros métodos no melhor valor, mas o VDNS proposto manteve superioridade na média, indicando maior robustez. Nas instâncias de menor porte (com até 75 clientes), como CMT01X, CMT06X e CMT02Y, os três métodos alcançaram os mesmos melhores valores. No entanto, a média das soluções favoreceu o VDNS proposto em praticamente todos os casos, demonstrando maior estabilidade. Já em instâncias de maior porte, como CMT04X, CMT05X e CMT10Y (com até 199 clientes), o VDNS proposto obteve as melhores soluções tanto em valor mínimo quanto médio, destacando-se significativamente em cenários de maior complexidade.

Apesar de apresentar bons resultados em instâncias pequenas e moderadas, o desempenho do HH-ILS decai em problemas maiores, tanto em qualidade das soluções quanto em variabilidade dos resultados. Já o VDNS original apresenta desempenho competitivo, mas foi superado pelo VDNS proposto, especialmente em termos de estabilidade das soluções. Em síntese, os dados mostram que as melhorias incorporadas ao VDNS proposto não apenas aumentam a qualidade das soluções finais, mas também conferem maior consistência aos resultados.

4.3. Comparação com a Resolução Direta do Modelo

Nesta análise, compara-se o desempenho do VDNS proposto com o CPLEX aplicado à formulação completa do G-VRPSPD, em duas configurações: (i) com *warm start*, utilizando a

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

Tabela 2: Comparação entre HH-ILS, VDNS Original e VDNS Proposto.

			HH-ILS Imp.		VDNS Original		VDNS Proposto	
Inst.	N_c	[Olgun et al., 2021]	Melhor	Média	Melhor	Média	Melhor	Média
CMT01X	50	745,10	**741,68**	742,26	**741,68**	742,48	**741,68**	**741,68**
CMT01Y	50	742,21	**741,68**	742,41	744,37	744,88	**741,68**	**741,68**
CMT06X	50		**741,68**	742,12	**741,68**	743,56	**741,68**	**741,68**
CMT06Y	50		**741,68**	742,26	**741,68**	743,81	**741,68**	**741,68**
CMT02X	75	1155,69	**1118,68**	1132,62	1121,72	1127,36	**1118,68**	**1118,68**
CMT02Y	75	1124,92	**1118,68**	1127,61	**1118,68**	1126,02	**1118,68**	**1118,68**
CMT07X	75		**1118,68**	1130,33	**1118,68**	1131,21	**1118,68**	**1118,68**
CMT07Y	75		1121,95	1131,28	**1118,68**	1123,74	**1118,68**	**1118,68**
CMT03X	100	1177,70	**1133,06**	1136,56	1133,52	1141,95	**1133,06**	**1133,14**
CMT03Y	100	1224,17	**1133,06**	1135,90	1133,81	1143,31	**1133,06**	**1133,14**
CMT08X	100		**1133,06**	1137,72	**1133,06**	1145,69	**1133,06**	**1133,14**
CMT08Y	100		**1133,06**	1133,66	1133,36	1140,19	**1133,06**	**1133,19**
CMT12X	100	1111,18	**1100,70**	1102,08	**1100,70**	1103,19	**1100,70**	**1100,70**
CMT12Y	100	1150,35	**1100,70**	1102,27	**1100,70**	1108,97	**1100,70**	**1100,70**
CMT14X	100		**1100,70**	1100,76	**1100,70**	1102,68	**1100,70**	**1100,70**
CMT14Y	100		**1100,70**	1100,95	**1100,70**	1105,96	**1100,70**	**1100,70**
CMT11X	120	1529,96	**1419,12**	1422,41	1419,87	1433,51	**1419,12**	**1419,12**
CMT11Y	120	1478,03	**1419,12**	1422,71	1419,20	1438,06	**1419,12**	**1419,12**
CMT13X	120		1419,40	1420,62	1419,32	1427,52	**1419,12**	**1419,12**
CMT13Y	120		**1419,12**	1422,95	1421,50	1430,31	**1419,12**	**1419,12**
CMT04X	150	1446,85	1395,78	1406,88	1412,77	1424,70	**1392,54**	**1392,83**
CMT04Y	150	1462,87	**1392,54**	1403,24	1407,81	1424,08	**1392,54**	**1393,11**
CMT09X	150		**1392,54**	1405,49	1403,08	1419,68	**1392,54**	**1393,25**
CMT09Y	150		1401,26	1405,05	1409,19	1423,43	**1392,54**	**1393,28**
CMT05X	199	1803,86	1718,09	1741,63	1726,28	1757,11	**1703,87**	**1705,46**
CMT05Y	199	1785,54	1717,89	1741,46	1734,00	1772,22	**1703,87**	**1705,05**
CMT10X	199		1709,54	1736,37	1730,36	1752,94	**1703,87**	**1705,49**
CMT10Y	199		1705,90	1739,32	1731,06	1782,04	**1704,23**	**1704,92**

mesma solução inicial adotada pelo VDNS; e (ii) sem solução inicial. Todas as abordagens são executadas com limite de 1 hora por instância. A Tabela 3 resume os resultados, apresentando o valor da solução inicial, os valores finais obtidos pelo CPLEX (com e sem *warm start*) e o valor final alcançado pelo VDNS proposto.

Os resultados demonstram que o VDNS proposto superou o CPLEX em todas as instâncias testadas, independentemente da configuração adotada. Mesmo quando o CPLEX partiu da mesma solução de entrada, foi incapaz de alcançar os mesmos valores obtidos pelo VDNS. A diferença se acentua nas instâncias de maior porte (como CMT05X, CMT09X e CMT10Y), indicando que o mecanismo de geração de subinstâncias do VDNS proporciona um balanceamento mais eficaz entre exploração local intensiva e custo computacional.

5. Conclusões

Este trabalho apresentou avanços na aplicação do algoritmo de Busca em Vizinhança com Profundidade Variável ao Problema de Roteamento de Veículos Verde com Coletas e Entregas Simultâneas. As principais contribuições concentram-se na incorporação de estratégias voltadas à definição mais eficiente das vizinhanças, incluindo mecanismos sistemáticos de seleção de clientes, com base em critérios de cobertura e proximidade geográfica, e a construção de subinstâncias a partir de múltiplas soluções correlacionadas.

Essas estratégias promovem um equilíbrio eficaz entre intensificação e diversificação da busca, resultando em soluções de alta qualidade e maior robustez. Os experimentos computacionais confirmam a superioridade do VDNS proposto em relação ao HH-ILS e ao VDNS original, com destaque para as instâncias de maior porte. Além disso, a comparação direta com a resolução

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

Tabela 3: Comparação entre CPLEX e o VDNS Proposto

Instância	Solução Inicial	CPLEX com SI	CPLEX sem SI	VDNS Proposto
CMT01X	746,85	746,85	753,23	**741,68**
CMT01Y	746,85	746,85	743,01	**741,68**
CMT06X	746,85	745,04	753,23	**741,68**
CMT06Y	746,85	746,85	743,01	**741,68**
CMT02X	1135,02	1135,02	1166,77	**1118,68**
CMT02Y	1130,04	1127,16	1138,85	**1118,68**
CMT07X	1141,94	1134,75	1168,91	**1118,68**
CMT07Y	1138,35	1138,35	1146,16	**1118,68**
CMT03X	1159,46	1150,82	1197,60	**1133,06**
CMT03Y	1159,63	1159,63	1166,12	**1133,06**
CMT08X	1169,36	1169,36	1243,74	**1133,06**
CMT08Y	1157,55	1150,34	1164,23	**1133,06**
CMT12X	1107,60	1102,72	1132,73	**1100,70**
CMT12Y	1104,01	1103,85	1148,82	**1100,70**
CMT14X	1101,29	1101,29	1132,73	**1100,70**
CMT14Y	1102,57	1102,45	1148,82	**1100,70**
CMT11X	1445,79	1445,79	1635,70	**1419,12**
CMT11Y	1451,94	1451,94	1874,76	**1419,12**
CMT13X	1463,42	1463,42	1635,70	**1419,12**
CMT13Y	1476,48	1476,48	1874,76	**1419,12**
CMT04X	1457,31	1457,31	1679,28	**1392,54**
CMT04Y	1453,64	1453,64	1630,31	**1392,54**
CMT09X	1476,07	1456,86	1761,13	**1392,54**
CMT09Y	1469,91	1469,91	1630,31	**1392,54**
CMT05X	1831,09	1830,47	2103,67	**1703,87**
CMT05Y	1832,05	1832,05	2171,37	**1703,87**
CMT10X	1838,78	1838,78	2103,67	**1703,87**
CMT10Y	1844,29	1844,29	2171,37	**1704,23**

completa via CPLEX evidencia a eficiência do uso de subinstâncias, mesmo sem *warm start*, na obtenção de resultados superiores dentro do mesmo tempo de execução.

Como direções futuras, propõe-se investigar mecanismos adaptativos para o controle do parâmetro de profundidade da clique e o uso de estratégias dinâmicas na formação do conjunto de soluções. A aplicação do método a variantes mais complexas do VRP — como aquelas com múltiplos objetivos, janelas de tempo ou frota heterogênea — também representa uma linha promissora de pesquisa.

Referências

Ahuja, R. K., Ergun, Ö., Orlin, J. B., e Punnen, A. P. (2002). A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123(1):75–102.

Alinaghian, M. e Naderipour, M. (2016). A novel comprehensive macroscopic model for time-dependent vehicle routing problem with multi-alternative graph to reduce fuel consumption: A case study. *Computers & Industrial Engineering*, 99:210–222.

Altner, D. S., Ahuja, R. K., Ergun, Ö., e Orlin, J. B. (2014). *Very Large-Scale Neighborhood Search*, p. 339–367. Springer US.

Asghari, M. e Mirzapour Al-e-hashem, S. M. J. (2021). Green vehicle routing problem: A state-of-the-art review. *International Journal of Production Economics*, 231:107899.

Avci, M. e Topaloglu, S. (2016). A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 53:160–17.

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

Barroso, D. O. e Nepomuceno, N. (2024). Um algoritmo de busca em vizinhança com profundidade variável para o problema de roteamento de veículos com coletas e entregas simultâneas. In *Anais do LVI Simpósio Brasileiro de Pesquisa Operacional (SBPO 2024)*, Fortaleza, Brasil. Galoá.

Blum, C., Pinacho, P., López-Ibáñez, M., e Lozano, J. A. (2016). Construct, merge, solve & adapt: A new general algorithm for combinatorial optimization. *Computers & Operations Research*, 68: 75–88.

Blum, C. e Raidl, G. R. (2016). *Hybridization Based on Problem Instance Reduction*, p. 45–62. Springer International Publishing.

Danna, E., Rothberg, E., e Pape, C. L. (2005). Exploring relaxation induced neighborhoods to improve MIP solutions. *Mathematical Programming*, 102(1):71–90.

Dekker, R., Bloemhof, J., e Mallidis, I. (2012). Operations research for green logistics – an overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, 219 (3):671–679. Feature Clusters.

Dell'Amico, M., Righini, G., e Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science*, 40(2): 235–247.

Demir, E., Bektaş, T., e Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5):347–357.

Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *OR Spektrum*, 23(1):79–96.

Erdoğan, S. e Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114. Select Papers from the 19th International Symposium on Transportation and Traffic Theory.

Franceschetti, A., Demir, E., Honhon, D., Van Woensel, T., Laporte, G., e Stobbe, M. (2017). A metaheuristic for the time-dependent pollution-routing problem. *European Journal of Operational Research*, 259(3):972–991.

Gajpal, Y. e Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*, 36(12):3215–3223.

Goksal, F. P., Karaoglan, I., e Altiparmak, F. (2013). A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering*, 65(1):39–53.

Huang, Y., Shi, C., Zhao, L., e Woensel, T. V. (2012). A study on carbon reduction in the vehicle routing problem with simultaneous pickups and deliveries. In *Proceedings of 2012 IEEE International Conference on Service Operations and Logistics, and Informatics*, p. 302–307.

Koç, Ç., Laporte, G., e Tükenmez, İ. (2020). A review of vehicle routing with simultaneous pickup and delivery. *Computers & Operations Research*, 122:1–15.

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br

Macrina, G., Laporte, G., Guerriero, F., e Di Puglia Pugliese, L. (2019). An energy-efficient green-vehicle routing problem with mixed vehicle fleet, partial battery recharging and time windows. *European Journal of Operational Research*, 276(3):971–982.

McKinnon, A. (2018). *Decarbonizing Logistics: Distributing Goods in a Low Carbon World*. Kogan Page, London. ISBN 978-0749480078.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5):377–386.

Montané, F. T. e Galvão, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, 33(3):595–619.

Nepomuceno, N. a., Saboia, R., e Coelho, A. (2023). A milp-based very large-scale neighborhood search for the heterogeneous vehicle routing problem with simultaneous pickup and delivery. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '23, p. 330–338, New York, NY, USA. Association for Computing Machinery.

Nepomuceno, N. V., Barboza Saboia, R., e Rogério Pinheiro, P. (2019). A fast randomized algorithm for the heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Algorithms*, 12(8):158.

Olgun, B., Çağrı Koç, e Altıparmak, F. (2021). A hyper heuristic for the green vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering*, 153:107010.

Rieck, J. e Zimmermann, J. (2013). Exact solutions to the symmetric and asymmetric vehicle routing problem with simultaneous delivery and pick-up. *Business Research*, 6(1):77–92.

Rogers, D. S. e Tibben-Lembke, R. (2001). An examination of reverse logistics practices. *Journal of Business Logistics*, 22(2):129–148.

Salhi, S. e Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *J. Oper. Res. Soc.*, 50(10):1034–1042.

Santos, M. J., Jorge, D., Ramos, T., e Barbosa-Póvoa, A. (2023). Green reverse logistics: Exploring the vehicle routing problem with deliveries and pickups. *Omega*, 118:102864.

Saraiva, R. D., Nepomuceno, N. V., e Pinheiro, P. R. (2013). The generate-and-solve framework revisited: Generating by simulated annealing. In Middendorf, M. e Blum, C., editors, *Evolutionary Computation in Combinatorial Optimization*, p. 262–273. Springer Berlin Heidelberg.

Subramanian, A., Drummond, L., Bentes, C., Ochi, L., e Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911.

Subramanian, A., Uchoa, E., Pessoa, A. A., e Ochi, L. S. (2013). Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optim. Lett.*, 7(7):1569–1581.

Tasan, A. S. e Gen, M. (2012). A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering*, 62(3):755–761.

DOI: 10.59254/sbpo-2025-212314
https://proceedings.science/p/212314?lang=pt-br