LVI SBPO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
DE 04 ATÉ 07 NOVEMBRO 2024
FORTALEZA | CEARÁ

UMA SOLUÇÃO PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E ENTREGA, JANELAS DE TEMPO E MÚLTIPLOS DEPÓSITOS USANDO UM ALGORITMO BASEADO EM COLÔNIA DE FORMIGAS

Marcelo Caramuru Pimentel Fraga
CEFET-MG
CEFET-MG Campus Divinópolis, Rua Álvares de Azevedo 400 - Bela Vista - Divinópolis
caramuru@cefetmg.br

Sérgio Ricardo de Souza
Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG
Av. Amazonas, 7.675, Bairro Nova Gameleira, Belo Horizonte/MG
sergio@cefetmg.br

Eduardo Goecking Carabetti
Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG
Av. Amazonas, 7.675, Bairro Nova Gameleira, Belo Horizonte/MG
carabetti@gmail.com

RESUMO

Este trabalho propõe um algoritmo baseado em Colônia de Formigas para resolver o Problema de Roteamento de Veículos com Coleta e Entrega, Janelas de Tempo e Múltiplos Depósitos (PRVCEJTMD). Devido à falta de instâncias específicas para o problema, foram utilizadas instâncias adaptadas de outros trabalhos da literatura para permitir a comparação dos resultados. Os resultados computacionais mostram que o algoritmo produz soluções competitivas em comparação com os melhores da literatura. Futuras pesquisas podem explorar novos movimentos de vizinhança e estratégias de refinamento para melhorar ainda mais a qualidade das soluções.

PALAVRAS CHAVE. Problema de Roteamento de Veículos com Janela de Tempo, Coleta e Entrega e Múltiplos Depósitos, Múltiplos depósitos, Problemas de Coleta e entrega.

Tópicos: Logística e Transportes. Metaheurística.

ABSTRACT

This paper proposes an Ant Colony Optimization algorithm to solve the Multi-Depots Pick-up and Delivery Problems with Time Windows - MDPDPTW. Due to the lack of specific problem instances, adapted instances from other literature papers were used for result comparison. Computational results demonstrate that the algorithm produces competitive solutions compared to the literature's best. Future research could explore new neighborhood movements and refinement strategies to enhance solution quality further.

KEYWORDS. Multi-Depots Pick-up and Delivery Problems with Time Windows. Multi-depot. Pickup and delivery Problems.

Paper topics: Logistics and Transport. Metaheuristic

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

1. Introdução

Este trabalho propõe a aplicação da metaheurística Colônia de Formigas para resolver o Problema Multidepósito de Roteamento de Veículos com Coleta e Entrega e Janelas de Tempo (PRVCEJTMD) (*Multi-Depots Pick-up and Delivery Problems with Time Windows* - MDPDPTW).

O transporte de cargas é vital para a segurança, qualidade de vida e desenvolvimento econômico de um país, fornecendo matéria-prima para as indústrias e entregando produtos essenciais aos consumidores. Ele reduz custos, aumenta a competitividade das empresas e promove o desenvolvimento regional. A eficiência no transporte garante a resiliência das cadeias de suprimentos, permitindo respostas ágeis a crises e emergências, assegurando que alimentos, medicamentos e outros bens essenciais cheguem rapidamente às populações necessitadas. Além disso, a conveniência da entrega de produtos diretamente nas casas dos consumidores melhora significativamente a qualidade de vida, proporcionando conforto e economia de tempo. Com a crescente adoção de tecnologias inovadoras e sustentáveis, o setor de transporte de cargas está se tornando mais eficiente e ambientalmente responsável, atendendo às demandas modernas de consumidores e empresas.

De acordo com Toth e Vigo [2002], a distribuição de produtos envolve determinar variáveis como o número e tipo de veículos necessários e a criação de rotas eficientes para minimizar custos e tempo de entrega, relacionando-se com o Problema de Roteamento de Veículos (PRV) proposto por Dantzig e Ramser [1959] para resolver desafios similares na distribuição de gasolina entre postos de abastecimento. O Problema de Roteamento de Veículos (PRV) é NP-difícil [Cordeau et al., 2005].

Os algoritmos de otimização de colônia de formigas (*Ant Colony Optimization* - ACO) são inspirado no comportamento das formigas reais em busca de alimentos. Esses algoritmos têm sido aplicado com sucesso em diversas áreas, como otimização de rotas, problemas de alocação de recursos e até mesmo em problemas complexos de otimização combinatória.

A proposta deste trabalho é o desenvolvimento de um algoritmo baseado em Colônia de Formigas para solucionar o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT). Estas adaptações visam aprimorar a eficácia do algoritmo, assegurando uma solução mais robusta e eficiente para o desafio de roteamento de veículos em situações com janelas de tempo. Os resultados iniciais são satisfatórios e indicam que as técnicas propostas tem potencial.

2. Descrição do problema

O Problema de Roteamento de Veículos com Coleta e Entrega, Janelas de Tempo e Múltiplos Depósitos (PRVCEJTMD) é uma combinação de diversas variantes do Problema de Roteamento de Veículos, especificamente o Problema de Roteamento de Veículos Capacitado (PRVC), o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT) e o Problema de Roteamento de Veículos com Coleta e Entrega (PRVCE).

No PRVC, uma frota de veículos deve atender a consumidores com diferentes demandas de produtos a partir de um único depósito, respeitando a capacidade de carga dos veículos. As demandas são conhecidas previamente e cada consumidor é atendido por um único veículo, com as rotas começando e terminando no depósito. Originalmente proposto por Dantzig e Ramser [1959] para solucionar um problema da distribuição de gasolina entre postos de abastecimento. O PRVJT é uma extensão do PRVC que adiciona restrições relacionadas aos horários em que cada consumidor deve ser atendido, além da restrição de capacidade. O PRVCE, por sua vez, busca determinar as rotas ótimas para uma frota de veículos que deve atender a um conjunto de pedidos, cada um com um ponto de coleta e um ponto de entrega, respeitando diversas restrições operacionais.

O PRVCEJTMD integra todos esses elementos, mas com a complexidade adicional de envolver múltiplos depósitos. Nesse problema, a frota de veículos pode estar distribuída em vários

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

depósitos, e cada veículo deve atender a demandas que incluem tanto pontos de coleta quanto de entrega, respeitando as janelas de tempo específicas para cada atendimento e a capacidade de carga dos veículos. Além disso, as rotas de cada veículo devem começar e terminar em qualquer um dos depósitos disponíveis. Este problema é particularmente relevante para grandes operações logísticas que precisam otimizar a eficiência e cumprir rigorosos requisitos de serviço.

O Problema de Coleta e Entrega se divide entre o Problema de Roteamento de Veículos com Retornos (Vehicle Routing Problems with Backhauls - VRPB) e o Problema de Roteamento de Veículos com Coletas e Entregas (*Vehicle Routing Problems with Pickups and Deliveries* - VRPPD) [Parragh et al., 2008]. De acordo com [Parragh et al., 2006], a classe VRPPD abrange problemas de transporte nos quais as mercadorias são coletadas e entregues desde a origem até o destino. Essa classe é subdividida em duas subclasses principais. A primeira trata de situações em que os pontos de coleta e entrega não estão emparelhados, sendo abordado principalmente o caso do veículo único, conhecido como Problema do Caixeiro Viajante Capacitado com Coletas e Entregas. A segunda subclasse envolve o clássico Problema de Coleta e Entrega (PDP) e o Problema Dial-A-Ride (DARP), que lidam com solicitações de transporte, cada uma associada a uma origem e um destino, resultando em pontos de coleta e entrega emparelhados. A PDP trata do transporte de mercadorias, enquanto o DARP trata do transporte de pessoas. O restante do artigo é organizado em torno da formulação matemática consistente desses problemas, seguida pela discussão de métodos de solução e uma visão geral da pesquisa representativa em cada domínio.

2.1. Modelo Matemático

O modelo matemático do PRVCEJTMD considerado neste estudo foi retirado de Harbaoui Dridi et al. [2020] e visa minimizar a distância total percorrida. As definições concernentes à formulação matemática apresentada estão postas na Tabela 1.

Tabela 1: Definições concernentes à formulação matemática (1)-(14)

L	Conjunto de depósitos, dado por $\{1, \ldots, \text{dep}\}$
H	Conjunto de nós (coleta e entrega) dado por $\{1, ldots, n\}$
H^+	Conjunto de nós de coleta, dado por $\{1, ldots, n/2\}$
H^-	Conjunto de nós de entrega, dado por $\{1, ldots, n/2\}$
H_c	Conjunto de pares (entrega e coleta), na forma $\{1, ldots, n/2\}$
C_i	O par (c_i, f_i) onde f_i é o nó de coleta e c_i é o nó de entrega correspondente, $\forall\ i\ in\ \{1, ldots, n/2\}$
V_m	Conjunto de veículos disponíveis no depósito m (i.e., o número máximo de veículos do depósito m que podem ser usados para coleta e entrega), $\{V_1, ldots, V_{\text{dep}}\}$
d_{ij}	Distância euclidiana entre os nós i e j
K	Número total de veículos disponíveis para todos os depósitos
k	Índice do veículo em cada depósito m, $\{1, ldots, V_m\}$
q_i	Quantidade de mercadorias solicitadas no nó i (se $q_i < 0$ é um nó de entrega, se $q_i > 0$ é um nó de coleta)
t_{ij}^k	Tempo necessário para o veículo k viajar do nó i para o nó j
Q	Capacidade máxima de um veículo
y_i^k	A carga do veículo k após sair do nó i
ET_i	O mais cedo que o nó i pode ser visitado por um veículo
LT_i	O último tempo permitido para deixar o nó i
S_i	Tempo de serviço no nó i
A_i	Tempo de chegada do veículo designado ao nó i
D_i	Tempo de partida do veículo do nó i
W_i	Tempo de espera do veículo no nó i
T_i	Tempo de atraso do veículo no nó i

A variável de decisão x_{ij}^{mk}, que modela a sequência em que o veículo visita os nós, possui valor $x_{ij}^{mk} = 1$ se o veículo k do depósito m viajar ao longo do arco (i, j); e $x_{ij}^{mk} = 0$, caso

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

contrário. A formulação matemática é, então, dada por:

$$\min \quad \sum_{m\in L}\sum_{k\in V_m}\sum_{i\in(H\cup m)}\sum_{j\in(H\cup m)} d_{ij}x_{ij}^{mk} \tag{1}$$

$$\text{suj. a} \quad \sum_{m\in L}\sum_{i\in H\cup L}\sum_{k\in V_m} x_{ij}^{mk} = 1, \quad \forall j \in H\cup L \tag{2}$$

$$\sum_{m\in L}\sum_{j\in H\cup L}\sum_{k\in V_m} x_{ij}^{mk} = 1, \quad \forall i \in H\cup L \tag{3}$$

$$\sum_{j\in H} x_{ij}^{mk} = \sum_{j\in H} x_{ji}^{mk}, \quad \forall i = m \in L, k \in V_m \tag{4}$$

$$x_{ij}^{mk} = 1 \Rightarrow y_i^k = 0, \quad \forall i \in L, j \in H, k \in V_m \tag{5}$$

$$x_{ji}^{mk} = 1 \Rightarrow y_i^k = 0, \quad \forall i \in L, j \in H, k \in V_m \tag{6}$$

$$x_{ij}^{mk} = 1 \Rightarrow y_j^k = y_i^k + q_j, \quad \forall i, j \in H, k \in V_m \tag{7}$$

$$0 \le y_i^k \le Q, \quad \forall i \in H, k \in V_m \tag{8}$$

$$x_{ij}^{mk} = 1 \Rightarrow A_j = D_i + t_{ij}^k, \quad \forall k \in V_m \tag{9}$$

$$D_i = A_i + s_i, \quad \forall i \in H \tag{10}$$

$$D_i = s_i = 0, \quad \forall i \in L \tag{11}$$

$$A_i < ET_i \Rightarrow W_i = ET_i - A_i, \quad \forall i \in H \tag{12}$$

$$T_i = \max(0, D_i - LT_i), \quad \forall i \in H \tag{13}$$

$$D_{f_i} < D_{c_i}, \quad \forall i \in H_c, f_i \in H^+, c_i \in H^- \tag{14}$$

As expressões (2) e (3) garantem que cada solicitação seja atendida uma vez pelo mesmo veículo. A expressão (4) assegura que cada veículo inicie e termine sua rota no mesmo depósito. As expressões (5) e (6) impõem que todos os veículos que saem e retornam ao depósito estejam descarregados. Para cada veículo de cada depósito, a carga do veículo k ao sair do nó j é definida conforme a expressão (7). A restrição de capacidade (8) garante que, em qualquer momento, a carga do veículo k não exceda a capacidade do veículo. Cada nó i possui um intervalo de tempo $[ET_i, LT_i]$ em que o serviço deve ocorrer. Esses intervalos de tempo, definidos nas expressões (9), (10) e (11), representam, respectivamente, o instante de chegada, o instante de partida e o tempo de serviço em cada depósito. O serviço de cada nó deve começar na janela de tempo aplicável. Se o veículo chegar

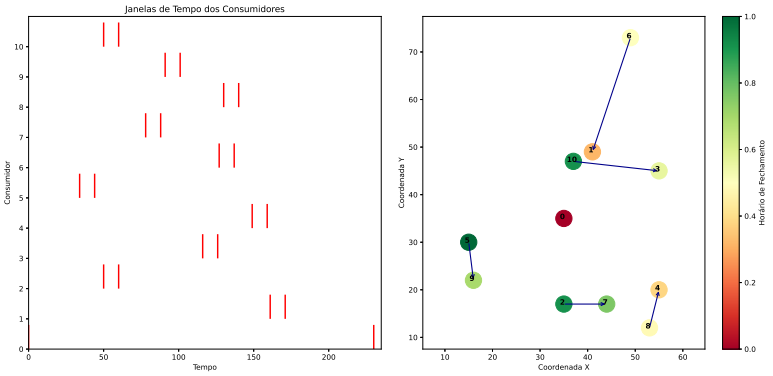


Figura 1: Pares de coleta e entrega no PRVJTCEMD.

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

ao nó antes do início do tempo de serviço aplicável, um tempo de espera é calculado conforme a expressão (12). Se o tempo de partida de um nó for maior que o seu último tempo de serviço, calcula-se um tempo de atraso, conforme a expressão (13). A restrição de precedência (14) garante que o nó de coleta (f_i) de cada par i seja visitado antes do nó de entrega correspondente (c_i). A Figura 1 mostra uma instância fictícia do PRVJTCEMD com 5 pares de consumidores de coleta e entrega e suas janelas de tempo.

3. Algoritmo de Colônia de Formigas

Os algoritmos baseados em Colônia de Formigas (*Ant Colony Optimization* - ACO) têm origem nos trabalhos de Marco Dorigo que, em Dorigo et al. [1991], propôs o algoritmo *Ant System* (AS) para solucionar o Problema do Caixeiro Viajante (PCV). Esse algoritmo consiste em um sistema multi-agente, no qual o comportamento de cada agente é inspirado no comportamento das formigas na natureza (veja, para uma revisão completa, Dorigo e Blum [2005]). A partir de então, diversas variações desse algoritmo têm sido reportadas na literatura, nas mais diversas aplicações. A essência dos algoritmos ACO é a combinação de informações sobre a estrutura do problema, conhecidas *a priori* e imutáveis, com as informações obtidas a partir dos resultados gerados, conhecidas *a posteriori* e dinâmicas. Outra grande vantagem dos algoritmos ACO é a facilidade com que pode ser paralelizado computacionalmente.

No *Ant System* (AS), a cada iteração, as formigas decidem, de acordo com uma medida probabilística, qual o caminho a seguir dentre aqueles não visitados. Cada formiga tem uma memória, também chamada de lista tabu, que armazena os caminhos percorridos por ela e previne que um caminho seja visitado pela formiga mais de uma vez. A probabilidade de escolha de um caminho é proporcional ao rastro de feromônio e à atratividade do mesmo - que varia de acordo com o tipo e a modelagem do problema em que o algoritmo está sendo aplicado. Se a formiga já percorreu aquele caminho, a probabilidade de escolha é zero; caso contrário, é positiva.

O algoritmo *Ant Colony System* (ACS) foi proposto por Gambardella e Dorigo [1996], Dorigo e Gambardella [1997a] e Dorigo e Gambardella [1997b] para superar performance do algoritmo *Ant System* e utiliza um critério de seleção mais agressivo; permite apenas que somente arcos associados à melhor solução global recebam feromônio; e as formigas removem parte do feromônio de cada arco (a_{ij}) que percorrem. No ACS existe um critério pseudo-randômico-proporcional de seleção de arcos que atua de forma gulosa

$$s = \begin{cases} \arg\max\limits_{u \notin M} \left\{ \left[\tau_{(r,u)}\right]^{\alpha} \left[\eta_{(r,u)}\right]^{\beta} \right\}, & \text{se } q \leq q_0; \\ \\ \text{expressão (16)}, & \text{caso contrário.} \end{cases} \tag{15}$$

ou probabilística, segundo:

$$p_k(r,s) = \begin{cases} \dfrac{\left[\tau_{(r,s)}\right]^{\alpha} \cdot \left[\eta_{(r,s)}\right]^{\beta}}{\sum\limits_{u \notin M_k} \left[\tau_{(r,u)}\right]^{\alpha} \cdot \left[\eta_{(r,u)}\right]^{\beta}}, & \text{se } s \notin M_k \\ \\ 0, & \text{caso contrário} \end{cases} \tag{16}$$

Sempre a formiga escolher o próximo nó para se mover, um número aleatório q é gerado e comparado ao parâmetro q_0, em que $q, q_0 \in [0,1]$. Se o valor de q for menor que o parâmetro q_0, o algoritmo funciona de forma gulosa, valorizando o aprendizado gerado pela deposição do feromônio. Caso contrário, o algoritmo age de forma probabilística, diversificando o espaço de busca.

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

Para as expressões 15 e 16, tem-se que $p_k(r,s)$ é a probabilidade da formiga k mover-se do nó r para o nó s; $\tau_{(r,s)}$ é a quantidade de feromônio associado ao arco (r,s); $\eta_{(r,s)}$ é a atratividade do arco (r,s); U é o conjunto de arcos não visitados; M_k é a lista tabu da formiga k; α e β são parâmetros de controle da influência de $\tau_{(r,s)}$ e $\eta_{(r,s)}$, situados na faixa $[0,1]$. Na expressão 16, a probabilidade de escolha de um consumidor destino é dada em função do rastro de feromônio e da atratividade do arco até o consumidor em questão. A atratividade (ou visibilidade) de um arco é calculada como o inverso da distância entre os consumidores do arco. Se a formiga já percorreu aquele arco a probabilidade de escolha é zero; caso contrário, é positiva. No ACS, a atualização global do feromônio é feita para privilegiar apenas os arcos pertencentes a melhor solução global. Para Dorigo e Gambardella [1997b], a atualização global do ACS acelera a convergência dos resultados, concentrando as buscas na vizinhança da melhor solução. A atualização global do feromônio é vista na expressão:

$$\tau_{(r,s)} = (1-\rho)\,\tau_{(r,s)} + \rho/J_{\Psi}^{gb}\,, \quad \forall\,(r,s) \in \Psi^{gb} \tag{17}$$

na qual $\tau_{(r,s)}$ é a quantidade de feromônio associado ao arco (r,s); ρ é a taxa de evaporação, tal que $\rho \in [0,1]$; J_{Ψ}^{gb} é a menor distância percorrida, computada desde o início da aplicação do algoritmo; e Ψ^{gb} é a melhor solução computada desde o início da aplicação do algoritmo. Segundo Ellabib et al. [2003], a atualização local do feromônio no ACS é feita durante a construção das soluções. Cada vez que uma formiga se move de um arco para outro a quantidade de feromônio associada ao arco é reduzida, e o arco se torna ligeiramente menos desejável para as demais. O resultado é a modificação dinâmica da escolha do arco pelas formigas. A atualização local do feromônio pode ser vista na expressão:

$$\tau_{(r,s)} = (1-\rho)\,\tau_{(r,s)} + \rho\tau_0 \tag{18}$$

em que $\tau_{(r,s)}$ é a quantidade de feromônio associado ao arco (r,s); ρ é a taxa de evaporação, de modo que $\rho \in [0,1]$; τ_0 é a quantidade de feromônio inicial em todos os arcos. Para Gambardella et al. [1999], um bom valor inicial para τ_0 é dado por $\tau_0 = \dfrac{1}{nJ_{\Psi}^{h}}$, sendo n o número de nós e J_{Ψ}^{h} a distância total percorrida, calculada pela heurística do Vizinho mais Próximo [Flood, 1956]. De acordo com Dorigo et al. [1999], os algoritmos baseados em colônias de formigas possuem, de uma forma geral, as seguintes características: (i) apesar de cada formiga ser capaz de encontrar uma solução, soluções de boa qualidade só emergirão como resultado da interação coletiva entre as formigas; (ii) cada formiga faz uso apenas de suas próprias informações, além das informações locais sobre o nó que visita; (iii) as formigas se comunicam apenas de forma indireta, através da leitura e escrita nas variáveis de rastro de feromônio; (iv) as formigas não são adaptáveis, pelo contrário, modificam a forma como o problema é representado e percebido para as outras formigas. Um procedimento chamado *Daemon Actions* pode ser usado opcionalmente como um processo centralizado de tomada de decisões, como invocar um procedimento de busca local ao final de cada iteração ou depositar uma quantidade extra de feromônio sobre a melhor solução. Esses são exemplos de decisões que não podem ser tomadas pelas formigas individualmente.

4. Metodologia

A metodologia adotada para o desenvolvimento deste trabalho começou com uma revisão concisa da literatura relacionada aos temas centrais: Problema de Roteamento de Veículos com Coleta e Entrega, Janelas de Tempo e Múltiplos Depósitos [Harbaoui Dridi et al., 2020; Sombuntham e Kachitvichyanukul, 2010; Bezerra et al., 2023], Problema de Roteamento de Veículos Capacitado [Ralphs et al., 2003], Problema de Roteamento de Veículos com Janelas de Tempo, Problema de Roteamento de Veículos com Coleta e Entrega [Cordeau et al., 2005; CARABETTI et al., 2010], e metaheurística ACO [Oliveira, 2022].

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

Como não existe um conjunto de instâncias de teste para o PRVCEJTMD, autores diversos propuseram alterações nas instâncias de Li e Lim [2001] para medir o desempenho de algoritmos para o PRVCEJT. As instâncias originais possuem apenas um depósito e, para torná-las multi-depósito, foram seguidos os passos apresentados a seguir. As modificações propostas por Harbaoui Dridi et al. [2020] adicionaram um segundo depósito nas coordenadas (34,32) nas instâncias LC, nas coordenadas (60,40) nas instâncias LR, e nas coordenadas (65,30) nas instâncias LRC. Por outro lado, Alaia et al. [2015] alteraram as instâncias tipo LC1 de Li e Lim [2001] e acrescentou mais dois depósitos nas coordenadas (0,40) e (88,35). Por fim, Alaïa et al. [2015] também modificaram as instâncias LC1 e estabeleceu as coordenadas para cada instância mostradas na Tabela 2. Neste trabalho, para evitar a necessidade de gerar códigos diferentes para cada tipo de instância

Tabela 2: Instâncias e respectivas coordenadas de depósitos extras.

LC101	1 depósito extra em (35, 33);
LC102	2 depósitos extras em (26, 29) e (67, 48);
LC103	1 depósito extra em (33, 33);
LC104	2 depósitos extras em (27, 31) e (71, 57);
LC105	2 depósitos extras em (25, 29) e (66, 49);
LC106	1 depósito extra em (42, 31);
LC107	1 depósito extra em (41, 32);
LC108	3 depósitos extras em (38, 26), (73, 55) e (13, 57);
LC109	[somente um depósito em (40, 50)]

multi-depósito, decidiu-se por modificar as instâncias originais de Li e Lim [2001], inserindo as informações dos depósitos extras. Esses depósitos foram adicionados repetindo a linha inicial de índice zero do depósito original quantas vezes forem necessárias, alterando apenas as coordenadas (x, y) para os novos depósitos. O algoritmo então tratará os depósitos subsequentes como índices $i_{\text{Depósitos}} = \{n + 1, n + 2, n + 3, \ldots\}$, onde n é o número total de consumidores da instância, conforme necessário na hora de mostrar as soluções.

As instâncias de Li e Lim [2001] para o PRVCEJT foram inspiradas nas instâncias de Solomon [1987] para o PRVJT. Estas instâncias são formadas por três grupos distintos, cada um com dois subgrupos. O grupo C consiste em consumidores distribuídos geograficamente em grupos próximos uns dos outros. O grupo R possui consumidores distribuídos aleatoriamente e distantes uns dos outros. Por fim, o grupo RC é uma mistura dos dois grupos anteriores, combinando características de distribuição dos grupos C e R. O subgrupo 1 de cada tipo (C1, R1 e RC1) possui janelas de tempo estreitas e o subgrupo 2 (C2, R2 e RC2) possui intervalos de janelas de tempo maiores. As instâncias são constituídas por 56 problemas-teste, para três conjuntos de 25, 50 e 100 consumidores que se dividem nos grupos citados previamente, totalizando 168 instâncias. Posteriormente, foi definida a forma de representar as soluções geradas, os movimentos de exploração de vizinhança e as funções de avaliação das soluções. Em seguida, iniciou-se o desenvolvimento do software, a calibração dos parâmetros e os testes iniciais para validar a eficácia das abordagens propostas. Por fim, realizou-se os testes finais para avaliar o desempenho e a robustez do software desenvolvido.

4.1. Forma de Representação das Soluções

A representação da solução no PRVCEJTMD consiste em um conjunto de rotas, geradas através do algoritmo desenvolvido para a solução do problema, que possuem um determinado custo associado, e um tamanho do número de rotas/veículos considerados. A estrutura de dados utilizada foi um vetor de vetores. O tamanho do vetor de vetores indica o número de rotas. Foi definida uma função de avaliação lexicográfica, onde o critério mais importante é o número de veículos e a

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

minimização da distância total percorrida em seguida. O tipo de dado associado a distância/tempo foi o *double* (número real de alta precisão).

Tabela 3: Forma de representação de uma solução.

$0-5-10-6-7-0$	$0-1-8-0$	$0-2-3-4-9-0$
Rota 1	Rota 2	Rota 3

Neste trabalho, o *Daemon Actions* consiste na aplicação de busca local na melhor formiga da iteração, variando os movimentos de exploração da vizinhança e utilizando três estratégias de refinamento.

Os movimentos de exploração da vizinhança consistem na realocação e troca de pares de consumidores de coleta e entrega. A realocação envolve remover um par de consumidores de coleta/entrega sua posição original e inseri-lo em outra posição, dentro da mesma rota ou em outra rota. Já a troca de pares implica na remoção de dois pares distintos de suas posições originais e na inserção dos pares com as posições invertidas, podendo ser realizada dentro da mesma rota ou entre rotas diferentes. Esses movimentos são aplicados como parte das estratégias de refinamento para otimizar as soluções nos problemas de roteamento de veículos com janelas de tempo e coleta e entrega. As estratégias de refinamento neste trabalho são três: Elimina Rota, Descida com Realocação e Descida Troca. A estratégia Elimina Rota utiliza o movimento de realocação e o método de refinamento por Descida com Primeira Melhora. Nela, pares de consumidores são removidos das menores rotas e inseridos nas maiores, visando reduzir o número de veículos, sem necessariamente melhorar o valor da função objetivo imediatamente, bastando apenas manter a factibilidade das rotas.

A estratégia Descida com Realocação é similar à Elimina Rota, porém, os movimentos só são aceitos se melhorarem o valor da função objetivo, e não é necessário eliminar toda a rota para aceitar o novo conjunto de rotas. Já a Descida Troca é semelhante à Descida com Realocação, mas utiliza o movimento de troca em vez de realocação. Nessa estratégia, a aceitação do novo conjunto de rotas também é baseada na melhoria do valor da função objetivo. Essas estratégias visam melhorar as soluções dos problemas de roteamento de veículos, utilizando movimentos específicos e critérios de aceitação baseados na otimização da função objetivo.

5. Resultados Computacionais

O sistema foi desenvolvido em C++, utilizando as APIs Boost para uso de threads. Para a compilação, foi utilizado o compilador GCC versão 9.4.0 de 64 bits. O hardware do sistema conta com uma CPU Core i7 8750h operando a 3.8GHz e 32GB de memória RAM. O ambiente operacional empregado foi o Windows 10, com o subsistema Linux WSL (Windows Subsystem for Linux) para execução em um ambiente Ubuntu 22.04 de 64 bits. Durante os testes, foram utilizadas instâncias de Li e Lim [2001] contendo aproximadamente 100 consumidores (varia entre 100 a 110) para o Problema de Roteamento de Veículos com Coleta e Entrega e Janelas de Tempo (PRVCEJT). Cada instância foi executada cinco vezes por uma hora e os melhores resultados obtidos foram selecionados.

Os melhores resultados heurísticos da literatura para o PRVCEJT foram retirados de Sintef [2024], que concentra as melhores soluções de diversos autores e técnicas. Já para o Problema de Roteamento de Veículos com Coleta e Entrega, Janelas de Tempo e Múltiplos Depósitos (PRVCEJTMD), os melhores resultados foram obtidos a partir dos estudos realizados por Harbaoui Dridi et al. [2020]; Alaia et al. [2015]; Alaïa et al. [2015]. É importante notar que os resultados não são diretamente comparáveis, uma vez que cada trabalho adota uma abordagem diferente. No entanto, cada um desses resultados tem sua relevância na comparação, conforme será detalhado adiante. Para

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

cada alteração possível nas instâncias serão apresentados os dados obtidos aqui de forma que sirva para futuras comparações. Os resultados provenientes de Sintef [2024] serão considerados como um limite superior para os resultados apresentados. Isso se deve ao fato de que os consumidores são os mesmos em cada instância, mas a diferença estará na adição de novos depósitos. O uso ou não de depósitos distintos ficará a cargo do algoritmo.

No estudo de Harbaoui Dridi et al. [2020], os autores se concentram exclusivamente na minimização da distância percorrida. Os resultados obtidos estão consideravelmente acima das soluções lexicográficas, que primeiro minimizam o número de veículos e, em seguida, a distância total percorrida. Por outro lado, em Alaïa et al. [2015], foram alcançados bons resultados ao reduzir tanto o número de veículos quanto a distância total percorrida. No entanto, é importante observar que os resultados para a instância LC109, que possui apenas um depósito, superam significativamente os melhores resultados da literatura relatados em Sintef [2024]. Isso levanta suspeitas sobre a validade dos dados apresentados, seja em relação ao número de depósitos ou à validação das soluções.

Finalmente, os resultados apresentados em Alaia et al. [2015] envolvem um modelo multi-objetivo que visa minimizar o número de veículos, a distância total percorrida e o atraso na entrega. No entanto, além de não listarem o nome da instância a que cada solução pertence, as curvas de Pareto não foram apresentadas, o que dificulta a compreensão completa dos resultados, já que apenas a melhor solução é relatada. Ainda assim, os valores apresentados são muito acima das melhores soluções em Sintef [2024], levantando suspeitas sobre a qualidade dos dados apresentados. Nas tabelas, o algoritmo desenvolvido foi chamado de CSC.

Tabela 4: Comparação dos resultados entre CSC, Sintef [2024] e Alaia et al. [2015]

Instância	CSC		Sintef [2024]		Alaia et al. [2015]		
	Número de Veículos	Custo	Número de Veículos	Custo	Número de Veículos	Custo	Atraso
lc101	10	698,89	10	828,94	11	1924,66	71390,07
lc102	10	698,89	10	828,94	9	1713,65	72114,19
lc103	9	1075,33	9	1035,35	6	1837,56	107388,46
lc104	9	840,84	9	860,01	9	1903,19	77750,51
lc105	10	696,50	10	828,94	8	1895,84	79811,35
lc106	10	695,99	10	828,94	11	1889,77	70816,22
lc107	10	712,90	10	828,94	-	-	-
lc108	10	695,99	10	826,44	-	-	-
lc109	10	700,55	9	1000,60	-	-	-

Para o conjunto de instâncias propostos em Alaia et al. [2015] que possui sempre 3 depósitos, os resultados podem ser vistos na Tabela 4. Observa-se que algoritmo CSC ficou abaixo do limite superior conhecido em Sintef [2024] em 2 das 9 instâncias, perdendo apenas na lc103 e lc109. Em relação ao trabalho de Alaia et al. [2015], os resultados obtidos são melhores em 3 das 6 instâncias.

Tabela 5: Comparação dos resultados entre CSC, Sintef [2024] e Alaïa et al. [2015]

Instância	CSC		Sintef [2024]		Alaïa et al. [2015]	
	Número de Veículos	Custo	Número de Veículos	Custo	Número de Veículos	Custo
lc101	10	758,31	10	828,94	6	739,96
lc102	10	691,27	10	828,94	7	765,29
lc103	10	750,96	9	1035,35	5	760,91
lc104	9	847,36	9	860,01	6	762,67
lc105	10	690,17	10	828,94	8	774,07
lc106	10	755,85	10	828,94	4	723,51
lc107	10	776,33	10	828,94	6	742,08
lc108	10	646,52	10	826,44	10	778,96
lc109	10	834,57	9	1000,60	**4**	**734,78**

O resultado para as instâncias que variam de zero a três depósitos extras, propostos em Alaïa et al. [2015], estão na Tabela 5. Neste caso, o algoritmo CSC ficou abaixo do limite superior

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

Tabela 6: Comparação dos resultados entre CSC, Sintef [2024] e Harbaoui Dridi et al. [2020]

Instância	CSC	Harbaoui Dridi et al. [2020]	Sintef [2024]
lc101	758,31	1839,40	828,94
lc102	754,16	1606,80	828,94
lc103	752,66	1638,47	1035,35
lc104	860,90	1658,78	860,01
lc106	758,31	1725,91	828,94
lc108	758,31	1757,38	826,44
lc201	591,56	947,91	591,56
lc202	591,56	1472,29	591,56
lc203	660,38	1885,89	591,17
lc206	645,15	1008,01	588,49
lc208	615,73	1133,53	588,32
lr103	1284,12	1956,60	1292,68
lr104	1023,59	1808,13	1013,39
lr201	1305,90	2060,41	1253,23
lr202	1296,56	2497,55	1197,67
lr208	1016,55	2535,83	734,85
lrc101	1590,66	2513,38	1708,80
lrc104	1099,83	2152,26	1128,40
lrc108	1133,71	2030,55	1147,43
lrc201	1499,75	2937,20	1406,94
lrc202	1541,87	2830,00	1374,27
lrc203	1445,01	2771,56	1089,07
lrc206	1198,47	2933,83	1159,03
lrc208	1097,00	2814,24	852,76
Total	**24280,06**	**48515,89**	**23518,2**

conhecido em Sintef [2024] em 2 das 9 instâncias, perdendo apenas na lc103 e lc109 novamente. Em relação ao trabalho de Alaïa et al. [2015], os resultados obtidos são melhores apenas na instância lc108, ficando bem abaixo em todas as demais. Contudo, parece existir um erro na instância lc109 do outro autor, pois o resultado é significativamente melhor que o melhor resultado da literatura. Ou existem mais depósitos, ou existem outras modificações nos parâmetros da instância que não foram relatados. Por fim, os resultados para o conjunto de instâncias de 2 depósitos do trabalho do Harbaoui Dridi et al. [2020] podem ser vistos na Tabela 6. É possível perceber que os resultados do algoritmo CSC proposto ficaram próximos a compilação dos melhores da literatura. As instâncias C são consideradas de mais fácil resolução e onde foi obtido a maior diferença para as demais soluções. A diferença nos demais grupos foi menor e indica outras técnicas devem ser incorporadas ao algoritmo.

6. Conclusões

Neste trabalho, é apresentado o desenvolvimento de um algoritmo baseado na metaheurística Colônia de Formigas para resolver um problema desafiador conhecido como Problema de Roteamento de Veículos com Coleta e Entrega, Janelas de Tempo e Múltiplos Depósitos (PRVCEJTMD). O PRVCEJTMD é classificado como um problema NP-Difícil, o que significa que sua resolução se torna muito complexa conforme o tamanho do problema aumenta. Portanto, justifica-se o uso de metaheurísticas para lidar com instâncias maiores. A partir da revisão da literatura, é evidente que os algoritmos baseados em Colônia de Formigas (ACO) têm a capacidade de gerar soluções competitivas para problemas desse tipo. Para adaptar esse conceito ao PRVCEJTMD, o algoritmo convencional foi modificado para incorporar dois tipos diferentes de movimentos de exploração na vizinhança e três estratégias distintas de busca local.

O Algoritmo de Colônia de Formigas Contínuo (CSC) demonstrou ser uma abordagem promissora para resolver o Problema de Roteamento de Veículos com Janelas de Tempo (PRVJT), com resultados competitivos em comparação aos melhores algoritmos disponíveis na literatura. Este

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

estudo destaca a importância contínua da pesquisa em metaheurísticas e otimização de algoritmos para lidar com problemas complexos do mundo real, como o Problema de Roteamento de Veículos com Coleta e Entrega em Janelas de Tempo e Múltiplos Depósitos (PRVCEJTMD). Ao ser avaliada em problemas-teste adaptados do estudo de Li e Lim [2001] para o PRVCEJT, os quais foram modificados para incluir múltiplos depósitos da PRVCEJTMD, a abordagem CSC produziu várias soluções abaixo do limite conhecido pelas melhores heurísticas da literatura. Isso sugere que a metodologia é robusta e eficaz. No entanto, em instâncias específicas, como LR e LRC, nosso algoritmo demonstrou que ainda há espaço para melhorias adicionais na exploração da vizinhança e nas estratégias de refinamento.

Uma possível adaptação para lidar com a otimização multiobjetivo permitiria comparar diretamente os resultados com outros estudos da literatura, como o de Alaïa et al. [2015]. A continuidade desses estudos pode levar a desenvolvimentos ainda mais eficazes e eficientes, contribuindo para avanços no campo de otimização e roteamento de veículos.

Referências

Alaia, E. B., Dridi, I. H., Bouchriha, H., e Borne, P. (2015). Genetic algorithm for multi-criteria optimization of multi-depots pick-up and delivery problems with time windows and multi-vehicles. *Acta Polytechnica Hungarica*, 12(8):155–174.

Alaïa, E. B., Dridi, I. H., Bouchriha, H., e Borne, P. (2015). Insertion of new depot locations for the optimization of multi-vehicles multi-depots pickup and delivery problems using genetic algorithm. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)*, p. 695–701. IEEE.

Bezerra, S. N., Souza, M. J. F., e de Souza, S. R. (2023). A variable neighborhood search-based algorithm with adaptive local search for the vehicle routing problem with time windows and multi-depots aiming for vehicle fleet reduction. *Computers & Operations Research*, 149:106016.

CARABETTI, E. G., SOUZA, S., e Fraga, M. C. (2010). Metaheurística colônia de formiga aplicada ao problema de roteamento de veículos com coleta e entrega e janela de tempo. *Anais do XIV Simpósio de Pesquisa Operacional e Logística da Marinha, Escola de Guerra Naval, Urca–RJ*.

Cordeau, J.-F., Ropke, S., e Laporte, G. (2005). Solving the pickup and delivery problem with time windows by branch-and-cut.

Dantzig, G. B. e Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6: 80–91.

Dorigo, M. e Blum, C. (2005). Ant colony optimization theory: a survey. *Theoretical Computer Science*, 344:243–278.

Dorigo, M., Caro, G. D., e Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172.

Dorigo, M. e Gambardella, L. M. (1997a). Ant colonies for the traveling salesman problem. *BioSystems*, 43:73–81.

Dorigo, M. e Gambardella, L. M. (1997b). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1:53–66.

Dorigo, M., Maniezzo, V., e Colorni, A. The ant system: an autocatalytic optimization process. Technical Report 91-016 Revised. Dipartimento di Elettronica, Politecnico di Milano, Itália, 1991.

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br

Ellabib, I., Basir, O. A., e Calamai, P. (2003). A new ant colony system updating strategy for vehicle routing problem with time windows. In *MIC2003: The Fifth Metaheuristics International Conference*, p. 18–1 – 18–6.

Flood, M. M. (1956). The traveling salesman problem. *Operations Research*, 4:61–75.

Gambardella, L. M. e Dorigo, M. (1996). Solving symmetric and asymmetric tsps by ant colonies. In *ICEC96 Proceedings of the IEEE Conference on Evolutionary Computation*, p. 622–627. IEEE Press.

Gambardella, L. M., Taillard, É., e Agazzi, G. (1999). Macs-vrptw: A multiple colony system for vehicle routing problems with time windows. In *New ideas in optimization*. Citeseer.

Harbaoui Dridi, I., Ben Alaïa, E., Borne, P., e Bouchriha, H. (2020). Optimisation of the multi-depots pick-up and delivery problems with time windows and multi-vehicles using pso algorithm. *International Journal of Production Research*, 58(14):4201–4214.

Li, H. e Lim, A. (2001). A metaheuristic for the pickup and delivery problem with time windows. In *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001*, p. 160–167. IEEE.

Oliveira, S. M. d. (2022). *Design configuration for MMAS algorithm applied for the Travelling Salesman Problem with Dynamic Demands*. PhD thesis, CEFET-MG, Belo Horizonte, MG. Tese de Doutorado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, como parte dos requisitos necessários para a obtenção do título de Doutora em Modelagem Matemática e Computacional.

Parragh, S. N., Doerner, K. F., e Hartl, R. F. (2006). A survey on pickup and delivery models part ii: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58: 81–117.

Parragh, S. N., Doerner, K. F., e Hartl, R. F. (2008). A survey on pickup and delivery problems: Part i: Transportation between customers and depot. *Journal für Betriebswirtschaft*, 58:21–51.

Ralphs, T. K., Kopman, L., Pulleyblank, W. R., e Trotter, L. E. (2003). On the capacitated vehicle routing problem. *Mathematical programming*, 94:343–359.

Sintef (2024). Solomon benchmark. URL `https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/`.

Solomon, M. M. (1987). Algorithms for vehicle routing and scheduling problems with time windows constraints. *European Journal of Operational Research*, 35:254–266.

Sombuntham, P. e Kachitvichyanukul, V. (2010). Multi-depot vehicle routing problem with pickup and delivery requests. In *AIP Conference Proceedings*, volume 1285, p. 71–85. American Institute of Physics.

Toth, P. e Vigo, D. (2002). *The Vehicle Routing Problem*. SIAM - Society for Industrial and Applied Mathematics, Philadelphia, 1^a edition.

DOI: 10.59254/sbpo-2024-193479
https://proceedings.science/p/193479?lang=pt-br