

Problema de roteamento de veículos capacitados (PRVC) usando agrupamento de rotas por K-Means e Heurística

Capacitated vehicle routing problem (CVRP) using K-Means route clustering and Heuristic

Problema de ruteo de vehículos con capacidad (PRVC) utilizando agrupamiento de rutas por K-Means y Heurística

DOI: 10.55905/oelv23n5-162

Receipt of originals: 4/25/2025

Acceptance for publication: 5/15/2025

Luiza Amalia Pinto

Doutora em Engenharia Elétrica

Instituição: Instituto de Ciência e Tecnologia de Sorocaba da Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP) – Campus de Sorocaba,

Endereço: Sorocaba, São Paulo, Brasil

E-mail: luiza.amalia@unesp.br

Júlio Cezar Estrella

Doutor em Ciência da Computação e Matemática Computacional

Instituição: Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) – Campus de São Carlos

Endereço: São Carlos, São Paulo, Brasil

E-mail: jcezar@icmc.usp.br

RESUMO

Este trabalho apresenta uma abordagem em duas fases para a resolução do Problema de Roteamento de Veículos Capacitados (PRVC), com aplicação em um estudo de caso de coleta seletiva. Na primeira fase, utiliza-se o método de clusterização K-Means para agrupar os pontos de coleta, reduzindo a complexidade do problema. Em seguida, aplica-se uma heurística baseada no algoritmo do vizinho mais próximo para definir a sequência de visitas aos clientes, visando a otimização das rotas com foco na minimização da distância percorrida e das emissões de CO₂. Os experimentos computacionais foram realizados com as instâncias de Auger et al., cujos resultados apresentaram soluções até 20% próximas das ótimas reportadas na literatura, com vantagens significativas em tempo de processamento e simplicidade algorítmica. Esses resultados evidenciam a viabilidade da combinação entre técnicas de clusterização e heurísticas na resolução eficiente e sustentável de problemas logísticos.

Palavras-chave: Problema de Roteamento de Veículos, K-Means, Heurística, Otimização.

ABSTRACT

This study presents a two-phase approach to solving the Capacitated Vehicle Routing Problem (CVRP), applied to a case study in selective waste collection. In the first phase, the K-Means clustering method is employed to group collection points, reducing the problem's complexity. Subsequently, a heuristic based on the Nearest Neighbor algorithm is applied to determine the visit sequence, aiming to optimize routes by minimizing both travel distance and CO₂ emissions. Computational experiments were conducted using the Augeret benchmark instances. The results yielded solutions within 20% of the optimal values reported in the literature, while offering significant advantages in computational time and algorithmic simplicity. These findings highlight the effectiveness of combining clustering techniques with lightweight heuristics for efficient and sustainable logistics problem-solving.

Keywords: Vehicle Routing Problem, K-Means, Heuristic, Optimization.

RESUMEN

Este estudio presenta un enfoque en dos fases para la resolución del Problema de Ruteo de Vehículos con Capacidad (CVRP), aplicado a un estudio de caso de recolección selectiva de residuos. En la primera fase, se utiliza el método de agrupamiento K-Means para agrupar los puntos de recolección, reduciendo así la complejidad del problema. Posteriormente, se aplica una heurística basada en el algoritmo del Vecino Más Cercano para determinar la secuencia de visitas, con el objetivo de optimizar las rutas minimizando tanto la distancia recorrida como las emisiones de CO₂. Los experimentos computacionales se realizaron utilizando las instancias de Augeret. Los resultados obtenidos estuvieron dentro del 20 % de los valores óptimos reportados en la literatura, con ventajas significativas en tiempo de procesamiento y simplicidad algorítmica. Estos hallazgos destacan la efectividad de combinar técnicas de agrupamiento con heurísticas ligeras para la resolución eficiente y sostenible de problemas logísticos.

Palabras clave: Problema de Ruteo de Vehículos, K-Means, Heurística, Optimización.

1 INTRODUÇÃO

O Problema de Roteamento de Veículos Capacitados (PRVC) é uma variação do clássico Problema de Roteamento de Veículos (PRV), no qual a demanda total dos clientes não pode exceder a capacidade de transporte dos veículos utilizados. Além dos desafios operacionais inerentes à otimização das rotas, aspectos ambientais vêm ganhando relevância crescente, especialmente no contexto de transporte urbano e coleta seletiva. Quando o PRV é formulado com restrições ambientais, como a minimização do

consumo de combustível ou da emissão de poluentes, ele passa a ser conhecido como *Green Vehicle Routing Problem* (GVRP).

Neste trabalho, propõe-se a implementação de uma heurística híbrida para resolver o PRVC com foco em critérios ambientais, mais especificamente a minimização da emissão de CO₂. A abordagem adotada combina o agrupamento dos nós utilizando o método K-Means com uma heurística de roteamento, aplicando uma função objetivo que considera não apenas a distância percorrida, mas também a inclinação do terreno entre os pares de nós. Essa função objetivo penaliza rotas que envolvem subidas carregadas, priorizando trajetos nos quais o veículo suba vazio e desça carregado, contribuindo assim para a redução das emissões (Cantão, Yamakami e Cantão, 2022).

A metodologia proposta segue os seguintes objetivos específicos:

1. Realizar a clusterização inicial dos nós do PRVC por meio do algoritmo K-Means, reduzindo a complexidade do problema;
2. Aplicar uma heurística para determinar a melhor sequência de visita aos nós dentro de cada agrupamento, com base na função objetivo que minimiza o custo total;
3. Empregar uma função objetivo que considere a inclinação entre os nós, priorizando rotas que resultem em menor emissão de CO₂;
4. Validar a implementação computacional utilizando as instâncias de Augerat *et al.* (1998) como benchmark;
5. Aplicar o modelo a dados real proveniente de uma cooperativa de materiais recicláveis da cidade de Sorocaba – SP, que inclui coordenadas geográficas (latitude, longitude e altitude), visando gerar rotas otimizadas do ponto de vista ambiental.

Além dos testes com instâncias clássicas, a aplicação em um cenário real de coleta seletiva reforça a relevância prática da abordagem, permitindo avaliar a efetividade do modelo proposto em um contexto em que a eficiência operacional e a sustentabilidade ambiental são igualmente prioritárias.

2 REFERENCIAL TEÓRICO

De acordo com Cordeau et al. (2007), o Problema de Roteamento de Veículos (PRV) é um processo essencial no gerenciamento do transporte e da distribuição de produtos, sendo amplamente estudado na área da Pesquisa Operacional e da Otimização Combinatória. Segundo He *et al.* (2009), o PRV possui inúmeras aplicações práticas em diversos setores logísticos, consolidando-se como um dos problemas clássicos dessa área.

A formulação original do PRV foi proposta por Dantzig e Ramser (1959), e desde então, diferentes variações e modelagens foram desenvolvidas para incorporar restrições realistas. Em sua configuração mais comum, o problema consiste em uma frota de veículos — não necessariamente homogênea — que parte de um depósito (geralmente representado pelo nó 0), visita um conjunto de clientes com demandas específicas e retorna ao depósito, obedecendo a restrições como capacidade de carga dos veículos, tempo de serviço e exclusividade no atendimento a cada cliente. O principal objetivo é a minimização do custo total associado às rotas.

O PRV é classificado como um problema NP-completo, visto que a quantidade de soluções possíveis cresce de forma exponencial com o número de clientes. Para um conjunto de n nós, o número de sequências possíveis de visita é dado por $(n-1)!$, tornando inviável a obtenção de soluções exatas em tempo hábil para instâncias de grande porte (Nallusamy *et al.*, 2009).

Em função dessa complexidade, diversas heurísticas e meta-heurísticas têm sido propostas para obter soluções aproximadas de boa qualidade, como os Algoritmos Genéticos (AG), inspirados no princípio da seleção natural de Darwin (Nallusamy *et al.*, 2009). Mais recentemente, abordagens híbridas vêm sendo exploradas com sucesso, combinando diferentes técnicas para melhorar o desempenho computacional. Um exemplo eficiente é o uso do algoritmo K-Means para o agrupamento inicial dos nós do PRV, reduzindo a complexidade do problema antes da aplicação de heurísticas de roteamento. Essa estratégia é discutida em trabalhos como os de Gocken e Yaktubay (2019), He *et al.* (2009), Putri et al. (2021) e Mohammadian-Semnani e Karkehabadi (s.d.).

Dessa forma, a implementação de um método híbrido utilizando K-Means associado a uma heurística, com programação em linguagem Python, mostra-se promissora para a obtenção de soluções eficientes e com menor custo computacional.

3 METODOLOGIA

3.1 PROBLEMA DE ROTEAMENTO DE VEÍCULOS CAPACITADOS

Dado um conjunto de solicitações de transportes de cargas e uma frota de veículos, o PRV busca encontrar uma rota para cada veículo que atenda a todas as solicitações, partindo de um depósito, com um custo mínimo, e que seja viável (atenda a todas as condições impostas pelo problema), retornando ao depósito (Irnich; Toth; Vigo, 2014).

Matematicamente o PRV consiste na distribuição (ou coleta) de produtos saindo de um único depósito, denotado como ponto 0 , para um conjunto de outros n pontos, ou clientes, $N = \{1, 2, \dots, n\}$.

Um PRV inicia e finaliza a sua rota em um depósito, o ponto 0 . A quantia que deve ser entregue (ou retirada) em cada cliente $i \in N$ é conhecida como demanda, dada por um valor $q_i > 0$. Para o caso do PRVC, denotamos por $Q > 0$ a capacidade do veículo.

Assumimos que a frota $K = \{1, 2, \dots, |K|\}$ é homogênea, ou seja, todos os veículos têm a

mesma capacidade e o mesmo custo operacional e denotamos por $|K|$ o número de veículos. Um veículo que atende um subconjunto de clientes $S \subseteq N$ inicia a rota no depósito, atende apenas uma vez cada cliente de S e finaliza retornando ao depósito. O deslocamento do veículo de i para j tem um custo c_{ij} .

Usando a estrutura de grafos direcionados ou não-direcionados, seja $V = \{0\} \cup N = \{0, \dots, n\}$ o conjunto de vértices (ou nós). No caso de simetria, quando os custos de deslocamento entre i e j são os mesmos de j para i , $G = (V, E)$ é um grafo completo e não-direcionado com o conjunto de arestas $E = \{e = \{i, j\} = \{j, i\}: i, j \in V, i \neq j\}$. Caso contrário, se pelo menos um par de vértices $i, j \in V$ tem custos assimétricos $c_{ij} \neq c_{ji}$, então temos um dígrafo completo $G = (V, A)$ com o conjunto de arcos $A = \{(i, j) \in V \times V: i \neq j\}$. No geral,

um PRVC é definido completamente pelo grafo $G = (V, E, c_{ij}, q_i)$ ou $G = (V, A, c_{ij}, q_i)$ com uma frota de $|K|$ veículos e capacidade Q .

Uma rota (ou tour) é uma sequência $r = \{i_0, i_1, i_2, \dots, i_s, i_{s+1}\}$ onde $i_0 = i_{s+1} = 0$, e $S = \{i_1, i_2, \dots, i_s\} \subseteq N$ representa o conjunto de clientes visitados. A rota r tem custo $c(r) = \sum_{p=0}^s c_{i_p, i_{p+1}}$ e é viável se a restrição de capacidade $q(s) = \sum_{i \in S} q_i \leq Q$, $q_i \leq Q$ é respeitada e cada cliente é visitado apenas uma vez.

A formulação matemática do PRVC empregada neste estudo corresponde à apresentada por Cantão, Yamakami e Cantão (2022), fundamentada no modelo geral descrito por Xiao et al. (2012), com alterações específicas na função objetivo. Para fins de validação do modelo e da estrutura algorítmica, usamos inicialmente uma função objetivo que minimiza a distância total percorrida, como mostrado na equação (1).

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (1)$$

onde:

d_{ij} é a distância do nó i até o nó j

x_{ij} é a variável binária, ou seja, se $x_{ij} = 1$ o veículo faz o percurso de i até j e $x_{ij} = 0$, caso contrário.

Após a etapa de validação, será empregada uma função objetivo voltada à minimização da emissão total de CO₂, conforme proposta por Toro et al. (2017), a qual leva em consideração a inclinação dos arcos. Essa função será aplicada ao estudo de caso de coleta seletiva.

3.2 CLUSTERIZAÇÃO

Segundo Sinaga e Yang (2020) e Shukla e Naganna (2014), o algoritmo K-Means é o mais conhecido e utilizado método de clusterização, particularmente em ciência de

dados, por exemplo, para reconhecimento de padrões. Em Nallusamy et al. (2009), o algoritmo classifica ou agrupa objetos baseado nos seus atributos ou características em k grupos, que é inteiro e positivo. Este agrupamento é realizado minimizando a soma das distâncias ao quadrado entre o dado e o centroide do k -ésimo cluster.

Ainda em Nallusamy et al. (2009), a ideia principal é definir k centroides, um para cada cluster. Os centroides devem ser bem-posicionados, pois diferentes locais resultam em agrupamentos diferentes.

De acordo com Pedregosa *et al.* (2011) a implementação do K-Means da biblioteca Scikit-Learn (Python) consiste em três etapas. Na primeira são escolhidos os centroides iniciais, seguido de um laço de repetição alternando entre a segunda e terceira etapas. Nesta segunda etapa cada amostra é atribuída ao seu centroide mais próximo. A terceira etapa cria-se centroides tomando o valor médio de todas as amostras atribuídas a cada centroide anterior. A diferença entre o centroide antigo e o novo é computada e o algoritmo repete essas duas últimas etapas até que esse valor seja menor que um limite. Em outras palavras, ele se repete até que os centroides não se movam significativamente. Este procedimento também é apresentado em (Nallusamy *et al.*, 2009) e (Shukla; Naganna, 2014).

3.3 PRVC E CLUSTERIZAÇÃO

Vários trabalhos da literatura citam o uso de K-Means como uma primeira fase na resolução de PRVC agrupando os nós por proximidade em relação a um centroide, como (Gocken; Yaktubay, 2019; He et al., 2009; Nallusamy et al., 2009; Putri *et al.*, 2021; Mohammadian-Semnani; Karkehabadi, s.d.). Na segunda fase, é aplicada uma meta-heurística para encontrar a melhor rota gerada na primeira fase (ou melhor ordenação dos nós a serem visitados) minimizando seu custo. Alguns autores usam algoritmos bio-inpirados para executar a segunda fase. Esta estratégia de resolução é conhecida como *cluster first Route second*.

Para a segunda fase, que consiste em encontrar uma rota ótima para cada agrupamento, podemos citar alguns trabalhos, como segue abaixo:

- Kassem et al. (2019) usa-se o *bat* algorithm, ou seja, um algoritmo que imita o comportamento do morcego;
- Moussa (2021) aplica o algoritmo de Dijkstra considerando o agrupamento como um grafo, onde cada nó é um vértice e as arestas, a distância de um nó ao outro;
- Korayem, Khorsid e Kassem (2015) apresenta o algoritmo baseado no comportamento dos lobos cinzentos (*grey wolf algorithm*) para a caça de suas presas;
- Al-Shboul e Myaeng (2009), Gocken e Yaktubay (2019), Mohammadian-Semnani e Karkehabadi (s.d.), Nallusamy et al. (2009), Putri et al. (2021) e Zeebaree et al. (2017) apresentam o Algoritmo Genético para melhorar as rotas encontradas na primeira fase.

3.4 K-MEANS

Em Shukla e Naganna (2014), a clusterização pelo algoritmo K-Means é executada de maneira iterativa, atribuindo n pontos ao k -ésimo cluster mais próximo, por uma medida de similaridade em relação ao seu centroide. Os passos a seguir resumizam este procedimento.

1. Um cluster inicial é criado escolhendo k centroides de forma aleatória;
2. Para cada ponto, é calculada a distância de todos os centroides, atribuindo-o ao centroide mais próximo;
3. O passo 2 é repetido até que os centroides não se alterem (convergência).

Em Python, a biblioteca `sklearn.cluster.KMeans` (Scikit-Learn, 2024) possui uma implementação deste algoritmo.

Segundo Shalev-Shwartz e Ben-David (2014), o algoritmo K-Means define uma função de custo sobre um conjunto de pontos a serem agrupados e o seu objetivo é o de encontrar uma clusterização que minimiza o custo. Assim, este procedimento passa a ser um problema de otimização, cujo objetivo é o de encontrar, dado um conjunto de elementos X e uma função de distância $d: X \times X \rightarrow R^+$, simétrica e $d(x, x) = 0, \forall x \in X$, uma partição do conjunto X em subconjuntos $C = (C_1, \dots, C_k)$, onde $\bigcup_{i=1}^k C_i = X, \forall i \neq j, C_i \cap C_j \neq \emptyset$.

A função objetivo é uma função de pares de uma entrada, (X, d) , e uma solução de agrupamento proposta $C = (C_1, \dots, C_k)$, para números reais positivos. Dada essa função objetivo, que denotamos por G , o objetivo de um algoritmo de agrupamento é o de encontrar, por uma dada entrada (X, d) , um agrupamento C de modo que $G((X, d), C)$ seja minimizado através de um algoritmo de busca apropriado.

Ainda, segundo Shalev-Shwartz e Ben-David (2014), este problema é NP-difícil, e o agrupamento por K-Means é uma aproximação particular de clusterização, tomando a função objetivo, como a distância euclidiana. Seja μ_i o centróide de C_i , assim:

$$\mu_i(C_i) = \arg \min_{\mu \in C_i} \sum_{x \in C_i} d(x, \mu)^2 \quad (2)$$

Logo, a função objetivo k-Means é dada por:

$$G_{\text{K-Means}}((X, d), (C_1, \dots, C_k)) = \sum_{i=1}^k \sum_{x \in C_i} d(x, \mu_i(C_i))^2 \quad (3)$$

Podemos reescrever a equação acima, como:

$$G_{\text{K-Means}}((X, d), (C_1, \dots, C_k)) = \min_{\mu_1, \dots, \mu_k \in X} \sum_{i=1}^k \sum_{x \in C_i} d(x, \mu_i)^2 \quad (4)$$

3.5 ORGANIZAÇÃO DAS ROTAS

Após a execução da função K-Means do Scikit-Learn, o algoritmo retorna quais pontos pertencem aos respectivos clusters. Porém, como estamos aplicando a estratégia de agrupamento para criar rotas em um PRVC, ocorre que alguns clusters possuem mais pontos, ou clientes, cuja demanda ultrapassa a capacidade do veículo. Assim, desenvolvemos uma função com a finalidade de organizar sub-rotas de maneira que a capacidade do veículo seja atendida.

1. Inicialização – recebe
 - a sequência de clientes para cada cluster;

- demanda de cada nó;
 - distância de cada nó em relação a cada centroide.
2. Para cada cluster, atribuímos os clientes deste núcleo até que a capacidade do veículo seja preenchida.
 3. Caso algum nó (ou cliente) não possa ficar neste cluster, é designado para uma lista denominada *sobra*.
 4. Quando todas as rotas forem concluídas, os clientes em *sobra* são designados um a um para as rotas mais próximas, em relação à distância do centroide, e a folga na capacidade do veículo.

O cálculo da distância é realizado através da distância Euclidiana no plano. Dados dois pontos (x_i, y_i) e (x_j, y_j) , a distância entre eles é dada por:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5)$$

3.6 BUSCA LOCAL – VIZINHO MAIS PRÓXIMO

Após a construção das sub-rotas, precisamos organizá-las de maneira que execute um tour (ou trajeto) com menor custo para a função objetivo. Neste momento, a otimização deste *tour* é um Problema do Caxeiro Viajante (PCV), com a diferença de que a inicialização e o término são sempre no depósito (nó 0). Assim, para cada rota, buscamos o nó mais próximo do depósito, entre todos os nós da rota, e o alocamos como o primeiro cliente a ser visitado, posteriormente, o nó mais próximo a este, até que todos os nós (clientes) sejam ordenados por este critério, como descrevem (Cunha; Bonasser; Abrahão, 2002) e (Silva *et al.*, 2013).

4 RESULTADOS E DISCUSSÕES

Os testes computacionais a seguir foram executados em um computador com processador Intel Core i7-2600, 3.4 GHz, com 8 GB de memória RAM e sistema

operacional GNU/Linux. A linguagem de programação usada é Python 3.9.6.

Neste trabalho utilizamos as instâncias de Augerat *et al.* (1998)¹ para os testes computacionais e posteriormente, aplicaremos a um estudo de caso, de uma cooperativa de reciclagem. Estas instâncias são usadas para testar modelos e implementações para PRVC, presentes em muitas publicações científicas desta área de estudo.

4.1 TESTES COMPUTACIONAIS

Na Tabela 1 estão compilados o nome do problema testado, a rota encontrada, o ótimo local desta rota e o ótimo global apresentado nas instâncias de Augerat *et al.* (1998), a diferença percentual entre o valor encontrado e o valor ótimo e o tempo computacional de cada instância (em segundos).

Em relação ao nome dos problemas, por exemplo A-n32-k5, ‘n32’ significa que há 32 nós (ou clientes) e ‘k5’, cinco veículos para atender toda a demanda. Além disso, todos os veículos deste benchmark têm capacidade de 100 unidades de carga.

Tabela 1: Problemas de Augeret, et al, (1998)

PROBLEMA	ROTAS	FUNÇÃO OBJETIVO	SOLUÇÃO ÓTIMA	DIFERENÇA %	TEMPO (S)
A-n32-k5	(0, 20, 5, 29, 15, 10, 25, 0) (0, 30, 13, 21, 31, 19, 17, 0) (0, 6, 3, 2, 23, 0) (0, 24, 18, 8, 28, 4, 11, 9, 22, 0) (0, 26, 16, 12, 1, 7, 14, 27, 0)	900.0553	787	14.36	47.76
A-n33-k5	(0, 28, 16, 3, 9, 17, 0) (0, 20, 4, 13, 8, 7, 26, 32, 0) (0, 22, 23, 18, 29, 15, 2, 0) (0, 12, 10, 30, 25, 27, 5, 0) (0, 11, 6, 24, 19, 14, 21, 1, 31, 0)	754.902	662	13.9	47.91
A-n33-k6	(0, 31, 24, 23, 26, 22, 0) (0, 10, 21, 12, 0) (0, 17, 11, 29, 19, 18, 0) (0, 28, 32, 25, 30, 16, 27, 0) (0, 14, 1, 6, 7, 13, 5, 0) (0, 8, 4, 3, 2, 15, 20, 9, 0)	786.184	742	5.93	46.997
A-n34-k5	(0, 20, 26, 4, 30, 24, 0) (0, 5, 27, 1, 23, 11, 8, 0) (0, 29, 19, 17, 25, 31, 28, 32, 0) (0, 18, 2, 16, 22, 9, 12, 3, 33, 0) (0, 10, 13, 7, 6, 15, 14, 21, 0)	846.663	778	8.74	50.514
A-n36-k5	(0, 34, 6, 3, 12, 31, 19, 4, 0) (0, 9, 28, 14, 23, 2, 35, 8, 15, 0) (0, 21, 22, 32, 18, 30, 17, 29, 33, 13, 0) (0, 26, 11, 25, 27, 24, 1, 0) (0, 10, 7, 20, 5, 16, 0)	893.43	799	11.76	54.51
A-n37-k5	(0, 36, 24, 9, 27, 8, 11, 20, 23, 0) (0, 7, 4, 33, 5, 13, 22, 0) (0, 34, 32, 28, 31, 26, 30, 35, 18, 25, 29) (0, 12, 17, 14, 19, 2, 10, 6, 0) (0, 15, 21, 16, 1, 3, 0)	805.283	669	20.33	48.77
A-n37-k6	(0, 25, 35, 27, 32, 0) (0, 8, 3, 1, 5, 33, 9, 0) (0, 24, 16, 7, 14, 36, 29, 0) (0, 13, 26, 10, 11, 15, 0) (0, 30, 12, 22, 23, 28, 2, 21, 0) (0, 18, 4, 17, 34, 31, 19, 20, 6, 0)	1016.955	949	7.06	52.96
A-n39-k5	(0, 14, 4, 18, 9, 12, 0) (0, 19, 25, 21, 27, 34, 24, 35, 37, 0) (0, 10, 5, 16, 7, 3, 2, 0) (0, 8, 11, 17, 26, 6, 36, 28, 13, 30, 0) (0, 33, 31, 1, 20, 23, 29, 15, 38, 32, 22, 0)	1005.264	829	21.23	52.65
A-n45-k7	(0, 32, 13, 18, 7, 16, 20, 38, 8, 9, 40, 0) (0, 22, 5, 30, 37, 1, 42, 0) (0, 27, 29, 31, 19, 36, 41, 0) (0, 28, 3, 11, 43, 0) (0, 39, 34, 26, 4, 6, 21, 2, 12, 0) (0, 17, 25, 23, 15, 10, 0) (0, 35, 24, 44, 33, 14, 0)	1268.09	1146	10.65	51.1

A-n54-k7	(0, 9, 33, 21, 3, 22, 0) (0, 28, 5, 7, 39, 50, 49, 29, 20, 23, 18, 0) (0, 32, 13, 11, 53, 27, 14, 2, 12, 0) (0, 43, 34, 41, 46, 51, 42, 24, 47, 25, 0) (0, 4, 52, 30, 16, 6, 10, 15, 35, 0) (0, 37, 8, 19, 31, 40, 48, 0) (0, 36, 1, 17, 26, 45, 38, 44, 0)	1309.786	1167	12.17	58.95
A-n63-k10	(0, 61, 46, 55, 33, 0) (0, 7, 21, 23, 54, 16, 37, 24, 0) (0, 59, 10, 5, 18, 45, 39, 3, 42, 19, 0) (0, 60, 8, 35, 27, 6, 0) (0, 20, 26, 13, 4, 30, 0) (0, 40, 29, 56, 25, 15, 49, 53, 0) (0, 32, 44, 47, 11, 34, 51, 58, 22, 0) (0, 14, 1, 41, 2, 9, 12, 0) (0, 17, 62, 50, 28, 57, 31, 0) (0, 48, 36, 52, 43, 38, 0)	1399.497	1314	6.47	58.36
A-n80-k10	(0, 62, 63, 34, 2, 37, 8, 0) (0, 74, 59, 19, 47, 56, 69, 65, 35, 26, 76, 58, 0) (0, 12, 5, 23, 24, 6, 30, 11, 0) (0, 49, 73, 36, 42, 51, 77, 53, 66, 67, 70, 0) (0, 28, 78, 68, 16, 43, 61, 57, 75, 20, 0) (0, 38, 50, 45, 22, 32, 4, 72, 0) (0, 3, 39, 60, 17, 31, 27, 29, 44, 0) (0, 40, 21, 1, 7, 13, 0) (0, 64, 33, 15, 41, 25, 46, 55, 9, 54, 0) (0, 10, 71, 14, 52, 79, 48, 18, 0)	1986.045	1766	12.46	62.96

Fonte: Elaborado pelos autores.

Nenhum problema atingiu a solução global pois a função K-Means estabiliza após algumas execuções, significando que a clusterização encontrada permite encontrar apenas soluções locais.

Com o aumento do número de nós, notamos que alguns pontos não foram atribuídos a nenhuma rota, pois todas estavam com a capacidade de carga próximo da sua limitação e a demanda restante não se encaixava em nenhuma rota pois excedia a capacidade dos veículos. Neste caso, criamos uma função que atribui o valor da carga faltante em cada veículo. Aquela que menos exceder a carga, terá a menor demanda atribuída a outro veículo e a carga faltante entra neste veículo.

Na organização das rotas, a busca pelo vizinho mais próximo daquela carga que ultrapassa o limite do seu cluster, faz uma distribuição balanceada da demanda entre os veículos, deixando pouca margem de manobra para uma demanda mais alta. Assim, um

vetor *sobra* foi criado para esta carga não atribuída. Soma-se este valor a todos os veículos; aquele que menos exceder, retira-se a menor carga, aloca a carga não atribuída e a demanda com menor carga passa para outro veículo com capacidade para recebê-la.

Todas as instâncias foram executadas pelo menos 4 vezes. A menor solução encontrada encontra-se na Tabela 1. Entre os problemas testados, A-n33-k6 (33 nós e 6 veículos) obteve a menor diferença (5.93%) entre a solução encontrada e a solução ótima catalogada, enquanto A-n39-5 (39 nós e 5 veículos) a maior diferença (21.23%).

4.2 ESTUDO DE CASO

O resíduo sólido sempre foi um problema para a população em geral. Nas áreas rurais, os resíduos podem ser utilizados para a compostagem orgânica, por exemplo. Porém, nas regiões urbanas e industriais, o que chamamos de Resíduos Sólidos Urbanos – RSU – tem no poder público a responsabilidade em definir um procedimento adequado para sua destinação. Porém, se esta destinação for inadequada, provoca danos ao meio ambiente e à população.

Neste aspecto, muito se discute e leis são elaboradas com a finalidade de garantir uma destinação correta para os RSU, como a Política Nacional de Resíduos Sólidos – PRNS, Lei nº 12.305, de 2010 (BRASIL, 2010). Um outro conceito que surgiu diante desta problemática são os três R's – Reduzir, Reutilizar e Reciclar. Entre elas, a reciclagem desperta mais interesse tanto da população quanto do poder público.

O processo de *coleta* dos RSU, pode ser facilitada quando os resíduos são coletados separados de alguma forma, ou seja, o resíduo seco é separado do resíduo orgânico, onde o primeiro pode ser destinado para a reciclagem (o que chamamos de coleta seletiva) e o segundo para os aterros sanitários. Portanto, a coleta seletiva é o primeiro estágio do processo de reutilização dos resíduos.

O serviço de coleta seletiva pode ser de responsabilidade da Prefeitura Municipal ou de instituições interessadas diretamente nos resíduos recicláveis, como associações de catadores, cooperativas etc., caracterizando o segundo estágio da cadeia de reutilização como o serviço de logística.

A terceira etapa do processo de reciclagem é a triagem, que consiste no recebimento dos resíduos, sua colocação em mesas de separação ou em uma esteira rolante, a catação e separação manual, lavagem, prensagem e armazenagem.

Os resíduos triados podem seguir destinos diversos, como compostagem (para os orgânicos), reciclagem (plásticos, metais, vidros, papéis etc.) e geração de energia (incineração – por exemplo, lixo hospitalar). Este processo gera o quarto estágio desta cadeia e caracteriza-se por um processo logístico.

Finalmente, o quinto estágio é a reutilização dos resíduos, propriamente dita. Em resumo, a cadeia de reutilização dos RSU pode ser representada a seguir:

Domicílios → logística da coleta → triagem → logística de distribuição dos resíduos selecionados → reutilização

Neste trabalho o foco é a segunda etapa, de logística da coleta, tendo como estudo de caso a cooperativa CORESO da cidade de Sorocaba – SP.

Este processo de logística já foi abordado em Santos, Cantão e Cantão (2019) e Cantão, Yamakami e Cantão (2022), visando a minimização do consumo de combustível e emissão de CO₂, respectivamente.

Os dados usados da cooperativa cobrem aproximadamente 230 ruas, de segunda à sexta-feira, coletando diretamente de residências e geradores coletivos. Como a cooperativa segue uma rotina diária de coleta, os dados foram separados por dias de semana e a cada dia, dois veículos atendem um determinado conjunto de pontos.

Neste estudo, usamos como função objetivo aquela que minimiza a emissão de CO₂, como detalhada em Cantão, Yamakami e Cantão (2022), considerando a inclinação das vias públicas, uma vez que um caminhão descarregado emite menos gases poluentes se subir vazio e descer carregado. Para este cenário, a estratégia do vizinho mais próximo foi modificada, buscando o vizinho com menor custo de emissão de CO₂.

Os endereços contêm as informações geográficas em coordenadas de latitude e longitude, que foram convertidas para metros usando a biblioteca *utm*, do ambiente de programação Python. A função *utm.from_latlon(Lat, Long)* retorna 4 parâmetros, sendo os dois primeiros relativos às coordenadas em metros, e os dois últimos, à zona no globo

terrestre. As coordenadas em metros foram usadas para a execução do programa, assim como para a representação gráfica das Figuras 1, 2, 3; 4 e 5.

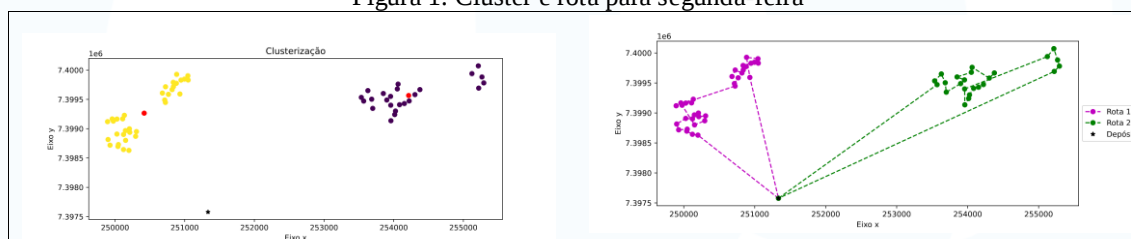
A Tabela 2 apresenta as rotas da cooperativa para cada dia da semana, seu custo total, distância percorrida e o tempo computacional para execução destes testes.

Tabela 2: Custos das rotas para a cooperativa Coreso

Dida da Se- mana	Função Objetivo (CO ₂ – kg)	Distância (m)	Tempo Computacional (s)
Segunda-feira	84.6184	19913.28	37.28
Terça-feira	54.6856	12685.14	50.44
Quarta-feira	52.2463	12148.05	42.01
Quinta-feira	61.3429	14259.12	46.19
Sexta-feira	42.3013	9788.09	47.56

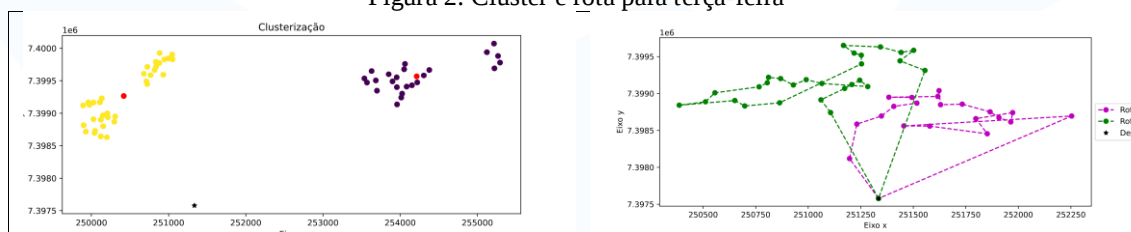
Fonte:Elaborado pelos autores.

Figura 1: Cluster e rota para segunda-feira



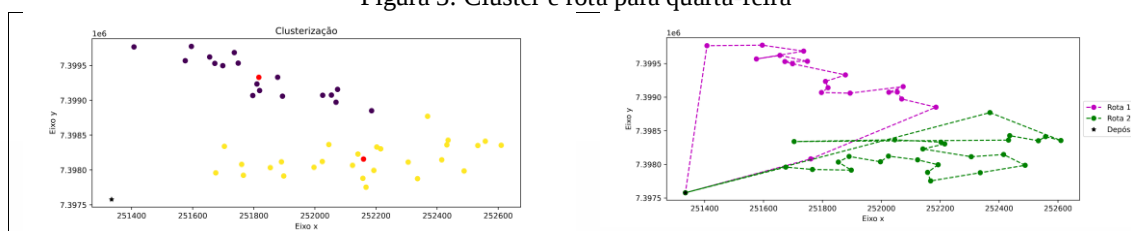
Fonte: Elaborado pelos autores.

Figura 2: Cluster e rota para terça-feira



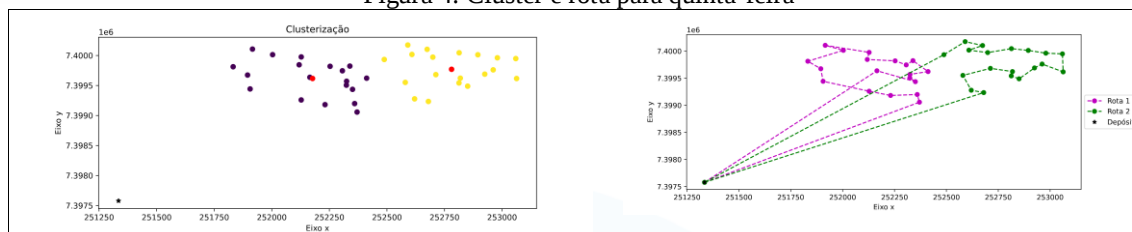
Fonte: Elaborado pelos autores.

Figura 3: Cluster e rota para quarta-feira



Fonte: Elaborado pelos autores.

Figura 4: Cluster e rota para quinta-feira



Fonte: Elaborado pelos autores.

Figura 5: Cluster e rota para sexta-feira



Fonte: Elabora pelos autores.

Por meio das Figuras 1, 2; 3; 4 e 5 percebemos que as rotas elaboradas estão dentro de cada cluster, uma vez que as demandas diárias estão abaixo da capacidade dos caminhões, permitindo que a cooperativa tenha uma margem de folga para recolher mais materiais recicláveis. Mesmo se esta demanda variar, ainda assim dificilmente ultrapassará a sua capacidade.

5 CONCLUSÃO

A finalidade de utilizar o algoritmo K-Means foi a de agregar os pontos (clientes) a serem visitados por proximidade. O número de centroides utilizados é igual ao número de veículos. Posteriormente, uma busca pelo vizinho mais próximo foi empregada para ordenar os clusters de maneira a minimizar a função objetivo. O vizinho mais próximo foi escolhido de acordo com a função objetivo utilizado: nas instâncias testes minimizamos a distância total. E no estudo de caso, minimizamos a emissão de CO₂.

Embora a heurística utilizada seja de busca local, os resultados mostraram-se até 20% próximos da solução global. Esta diferença está relacionada ao fato de que a função da linguagem de programação Python, K-Means, apresentar o melhor agrupamento, sem

grandes variações a partir de um certo número de iterações. O que muda é a ordem que os grupos são apresentados, ou seja, a rota 1 pode ser a rota 3 ou outro valor nas outras gerações. Com isso, quando se executa o processo de balanceamento da carga transportada de uma rota, o ponto relacionado à carga excedente vai para o primeiro cluster mais próximo, de acordo com a distância do centroide do cluster, que em outro processo, este valor pode diferir.

Há softwares, como o CPLEX que resolve este tipo de problema de maneira exata, contudo esta ferramenta é paga e nem sempre é possível obtê-la. Logo, uma implementação baseada em clusterização como primeira etapa e implementação de uma heurística (ou metaheurística) com uma solução aproximada, como uma segunda etapa, tende a ser viável e seu tempo computacional muito menor, como apresentado nas Tabelas 1 e 2.

AGRADECIMENTOS

Ao Instituto de Ciência e Tecnologia de Sorocaba – ICTS, UNESP – campus de Sorocaba, pelo apoio ao pesquisador através do grupo de pesquisa NATEL (Núcleo de Automação e Tecnologias Limpas).

REFERÊNCIAS

- AL-SHBOUL, B.; MYAENG, S.-H. Initializing k-means using genetic algorithms. **World Academy of Science, Engineering and Technology**, v. 54, n. 30, p. 114–118, 2009.
- AUGERAT, P.; BELENGUER, J.-M.; BENAVENT, E.; CORBÉLAN, A.; NADDEF, D. Separating capacity constraints in the cvrp using tabu search. **European Journal of Operational Research**, Elsevier, v. 106, n. 2-3, p. 546–557, 1998.
- BRASIL. Lei Nº 12.305. 2010. Disponível em:
<https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2010/lei/112305.htm>.
- CANTÃO, L.; YAMAKAMI, A.; CANTÃO, R. Capacited vehicle routing problem with co₂ emission minimization considering path slopes. **Trends in Computational and Applied Mathematics**, SciELO Brasil, v. 23, p. 439–469, 2022.
- CORDEAU, J.; LAPORTE, G.; SAVELSBERGH, M.; VIGO, D. **Vehicle routing handbooks in operations research and management science**. [S.l.]: Elsevier, Amsterdam, 2007.
- CUNHA, C. B. da; BONASSER, U. de O.; ABRAHÃO, F. T. M. Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante. In: **XVI Congresso da Anpet**. [S.l.: s.n.], 2002.
- GOCKEN, T.; YAKTUBAY, M. Comparison of different clustering algorithms via genetic algorithm for VRPTW. **International Journal of Simulation Modelling**, v. 18, n. 4, p. 574-585, 2019.
- HE, R.; XU, W.; SUN, J.; ZU, B. Balanced k-means algorithm for partitioning areas in large-scale vehicle routing problem. In: IEEE. **2009 Third International Symposium on Intelligent Information Technology Application**. [S.l.], v. 3, p. 87–90, 2009.
- IRNICH, S.; TOTH, P.; VIGO, D. Vehicle routing: problems methods, and applications. In: _____. Second edition. [S.l.]: SIAM, 2014. cap. The family of vehicle routing problems.
- KASSEM, S.; KORAYEM, L.; KHORSHID, M.; THARWAT, A. A hybrid bat algorithm to solve the capacitated vehicle routing problem. In: IEEE. **2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES)**. [S.l.], v. 1, p. 222–225, 2019.
- KORAYEM, L.; KHORSID, M.; KASSEM, S. Using grey wolf algorithm to solve the capacitated vehicle routing problem. In: **IOP PUBLISHING. IOP conference series: materials Science and engineering**. [S.l.], v. 83, n. 1, p. 012014, 2015.

MOHAMMADIAN-SEMNANI, S.; KARKEHABADI, M. Finding optimized routes in Vehicle routing problem with backhaul by combining genetic algorithm and improved k-means. s.d.

MOUSSA, H. Using recursive kmeans and dijkstra algorithm to solve cvrp. **arXiv preprint arXiv:2102.00567**, 2021.

NALLUSAMY, R.; DURAI SWAMY, K.; DHANALAKSMI, R.; PARTHIBAN, P. Optimization of multiple vehicle routing problems using approximation algorithms. **International Journal of Engineering Science and Technology**, v. 1, n. 3, p. 129–135, 2009.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COUNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Disponível em: <<https://scikit-learn.org/stable/modules/clustering.html#k-means>>.

PUTRI, K. A.; RACHMAWATI, N. L.; LUSIANI, M.; REDI, A. A. N. P. Genetic algorithm with cluster-first route-second to solve the capacitated vehicle routing problem with time windows: A case study. **Jurnal Teknik Industri**, v. 23, n. 1, p. 75–82, 2021.

SANTOS, G. T.; CANTÃO, L. A. P.; CANTÃO, R. F. An ant colony system metaheuristic applied to a cooperative of recyclable materials of sorocaba: A case study. In: **Smart and Digital Cities**. [S.l.]: Springer, 2019. p. 79–97.

SCIKIT-LEARN. *sklearn.cluster.KMeans*. Scikit-learn, 2024. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>. Acesso em: 9 maio 2025.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge university press, 2014.

SHUKLA, S.; NAGANNA, S. A review on k-means data clustering approach. **International Journal of Information and Computation Technology**, v. 4, n. 17, p. 1847–1860, 2014.

SILVA, G. A. N. da; SILVA, F. A. da; RUSSI, D. T. A.; PAZOTI, M. A.; SISCOOTTO, R. A. Algoritmos heurísticos construtivos aplicados ao problema do caixeiro viajante para a definição de rotas otimizadas. In: **Colloquium Exactarum**. ISSN: 2178-8332. [S.l.: s.n.], 2013. v. 5, n. 2, p. 30–46.

SINAGA, K. P.; YANG, M.-S. Unsupervised k-means clustering algorithm. **IEEE access**, IEEE, v. 8, p. 80716–80727, 2020.

TORO, E. M.; FRANCO, J. F.; ECHEVERRI, M. G.; GUIMARÃES, F. G. A multi-objective model for the green capacitated location-routing problem considering environmental impact. **Computers and Industrial Engineering**, v. 110, p. 114 – 125, 2017. ISSN 0360-8352. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0360835217302176>>.

XIAO, Y.; ZHAO, Q.; KAKU, I.; XU, Y. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. **Computers and Operations Research**, v. 39, n. 7, p. 1419 – 1431, 2012. ISSN 0305-0548. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0305054811002450>>.

ZEEBAREE, D. Q.; HARON, H.; ABDULAZEEZ, A. M.; ZEEBAREE, S. Combination of k-means clustering with genetic algorithm: A review. **International Journal of Applied Engineering Research**, v. 12, n. 24, p. 14238–14245, 2017.