

**ANÁLISE DO USO DA FERRAMENTA OR-TOOLS PARA GERAR ROTAS  
OTIMIZADAS NA AGÊNCIA DOS CORREIOS DE RUSSAS-CE TENDO COMO BASE  
O PROBLEMA DO CAIXEIRO VIAJANTE**

**ANALYSIS OF THE USE OF THE OR-TOOLS TOOL TO GENERATE OPTIMIZED  
ROUTES IN THE RUSSAS-CE POST OFFICE BASED ON THE TRAVELING  
SALESMAN PROBLEM**

Éolo Charles Da Silva\*

Diego Rocha Lima\*\*

**RESUMO**

A ECT (Empresa Brasileira de Correios e Telégrafos) é a maior operadora logística do Brasil, sendo a principal fornecedora do serviço de entregas dos produtos adquiridos no comércio eletrônico e responsável por 36% do total de entregas deste mercado em 2021. O grupo formado pelos segmentos encomenda, logística e internacional vem aumentando sua participação na receita total dos Correios, representando 66,8% da receita total em 2021 contra 57,4% em 2020 e 53,3% em 2019. Os Correios entregam em média 14,1 milhões de objetos diariamente, e fazem uso do Traffic Management System (TMS) para elaborar rotas otimizadas para entrega de encomendas na maioria das suas unidades operacionais, contudo ainda existem unidades onde a implantação não foi realizada, neste caso, fica a cargo da experiência dos carteiros elaborar tais rotas. Este trabalho propõe a partir do Problema do Caixeiro Viajante (PCV), um já bem conhecido problema na área de otimização, modelar o distrito especial da agência de correios de Russas-CE, como um PCV, e utilizar o *Google OR-Tools*, para roteirização das entregas diárias e testar sua viabilidade na elaboração de rotas otimizadas para os distritos em comparação com os métodos atualmente utilizados, buscando minimizar os custos e reduzir atrasos na entrega de objetos postais, aumentando-se desse modo a eficiência operacional para ganho de competitividade.

**Palavras-chave:** Otimização. Problema do Caixeiro viajante. Correios. Logística.

**ABSTRACT**

---

\* Autor, Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus Aracati (IFCE) Rodovia CE-040, Km 137,1 s/n Aeroporto, CE – 62.800-000 – Aracati – CE – Brazil, eolo.charles.silva04@aluno.ifce.edu.br .

\*\* Orientador, Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus Aracati (IFCE) Rodovia CE-040, Km 137,1 s/n Aeroporto, CE – 62.800-000 – Aracati – CE – Brazil, diego.rocha@ifce.edu.br.

ECT (Empresa Brasileira de Correios e Telégrafos) is the largest logistics operator in Brazil, being the main provider of delivery service for products purchased in e-commerce and responsible for 36% of the total deliveries in this market in 2021. The group formed by the parcel, logistics and international segments has been increasing its share in the total revenue of Correios, representing 66.8% of total revenue in 2021 against 57.4% in 2020 and 53.3% in 2019. Correios delivers on average 14.1 million objects daily, and make use of the Traffic Management System (TMS) to develop optimized routes for delivery of parcels in most of its operational units, however there are still units where the implementation has not been carried out, in this case, it is up to of the postmen's experience in designing such routes. This work proposes, from the Traveling Salesman Problem (PCV), a well-known problem in the optimization area, to model the special district of the post office in Russas-CE, as a PCV, and use the *Google OR-Tools*, for routing daily deliveries and testing their feasibility in developing optimized routes for the districts compared to the methods currently used, seeking to minimize costs and reduce delays in the delivery of postal items, thus increasing operational efficiency for competitiveness gain.

**Keywords:** Optimization. Traveling salesman Problem. Mail. Logistics.

## 1 INTRODUÇÃO

A ECT (Empresa Brasileira de Correios e Telégrafos) é a maior operadora logística do Brasil. Em 2021, foram entregues mais de 14,2 milhões de objetos postais por dia, sendo 11,8 milhões de correspondências e 2,3 milhões de encomendas nacionais e internacionais em 5556 municípios brasileiros. Para garantir esse volume de entregas, foram rodados mais de 1,7 milhões de quilômetros, usando mais de 23 mil veículos (CORREIOS, 2022).

Segundo o relatório integrado Correios 2021 divulgado em maio de 2022, a participação de mercado de encomendas dos operadores postais varia entre os países, normalmente na faixa de 20% a 40%. Os Correios encontram-se dentro dessa faixa, com 36% de participação em 2021. Ainda segundo o mesmo relatório o grupo formado pelos segmentos encomenda, logística e internacional vem aumentando sua participação na receita total dos Correios (que contempla a receita operacional, as receitas financeiras e outras receitas operacionais), representando 66,8% em 2021 contra 57,4% em 2020 e 53,3% em 2019, o que reduz a vulnerabilidade decorrente de a empresa ter receitas oriundas de serviços substituíveis pela utilização crescente de mídias eletrônicas para comunicação (CORREIOS, 2022).

Para lidar com um volume tão grande de entregas diárias, e sendo estas uma das principais fontes de receita, os Correios utilizam o Traffic Management System (TMS) uma ferramenta proprietária para elaborar rotas otimizadas na entrega de encomendas, contudo ainda há unidades onde não foi implantado o uso deste sistema. Neste caso a experiência dos carteiros é a principal forma de elaborar tais rotas nestas unidades operacionais da empresa. Após receberem a carga proveniente dos Centros de Tratamento de Cartas e Encomendas (CTCE), os carteiros são responsáveis por realizarem a Triagem por Distrito (TD) e Separação por Logradouro (SL), bem

como elaborar a ordem de entrega dos objetos postais do dia.

No contexto atual esse problema tem se tornado ainda mais relevante. Diante do advento do comércio eletrônico, a demanda por serviços de entrega de mercadorias e distribuição de serviços a domicílio só aumenta a cada ano. Apenas no primeiro semestre de 2021 o faturamento em vendas *online* no Brasil cresceu 31% em comparação com o ano anterior (EBIT, 2021). Segundo dados da 6ª rodada da Pesquisa de Impacto no Transporte – Covid-19, realizada pela Confederação Nacional do Transporte (CNT), a maioria das empresas entrevistadas acreditam que não haverá mudanças no cenário nos próximos seis meses. Das 580 organizações de cargas e passageiros de todos os modais de transporte ouvidas, 28,6% estão otimistas e vislumbram um crescimento da demanda no setor (CNT, 2021).

Diante deste cenário e com aumento da concorrência, buscar soluções que tragam ganhos de eficiência é imprescindível para as empresas deste setor se manterem crescendo de forma sustentável. Segundo pesquisa da Capterra os clientes estão cada vez mais buscando por serviços que ofereçam prazos menores e preços mais acessíveis (CAPTERRA, 2022). Sendo essencial o uso da tecnologia para prover estes ganhos.

Em geral os Correios tratam os objetos postais em dois grupos, objetos simples e qualificados, sem e com rastreio respectivamente. Dependendo do porte da cidade pode haver uma AC (Agência de Correios com distribuição), um CDD (Centro de Distribuição Domiciliar) ou um CEE (Centro de Entrega de Encomendas), estes são os responsáveis pela entrega no trecho final e onde são elaboradas as rotas diárias.

Nas ACs, em geral, há os chamados distritos especiais que lidam apenas com objetos qualificados enquanto os distritos convencionais fazem a entrega de ambos os tipos. Levando-se em conta apenas os distritos especiais o número de pontos de entregas diários em média ultrapassa cinquenta entregas, já os distritos convencionais se considerados todos os objetos, simples e registrados, pode passar facilmente de mais de trezentas entregas. Como descrito em (FILIP; OTAKAR, 2011), a complexidade aumenta a medida que cresce o número de clientes a serem visitados, desse modo, a elaboração de rotas postais usando um modelo baseado no Problema do Caixeiro Viajante não é trivial em termos computacionais.

O Problema do Caixeiro Viajante (PCV) faz parte da categoria NP-completo e é muito estudado nos campos de Ciência da Computação além de Pesquisa Operacional (ILAVARASI; JOSEPH, 2014). Proposto em 1934 por Hassler Whitney em uma palestra de um seminário na Universidade de Princeton (FLOOD, 1956), ele recebe esse nome por se assemelhar à situação onde um vendedor precisa encontrar uma rota que comece em sua residência, visite um conjunto prescrito de cidades e retorne ao local de origem de forma que a distância total percorrida seja mínima e cada cidade seja visitada exatamente uma vez (GUTIN; PUNNEN, 2006). Apesar de fácil de formular o PCV é computacionalmente difícil de calcular e tem vasta aplicação em problemas da vida real, como no caso de roteirização para entregas postais.

Os métodos para resolver o Problema do caixeiro Viajante normalmente podem ser divididos em três partes básicas: um ponto de partida, um esquema de geração de solução, e uma regra de parada. Dizemos que o método é exato se e somente se, quando a iteração termina,

obedecida a regra de parada, a rota gerada é a ideal (BELLMORE; NEMHAUSER, 1968).

Utilizar um sistema de otimização de rotas pode representar um diferencial competitivo nesse mercado de crescente concorrência. A economia em processos de entrega pode significar margens de lucro maiores ou a capacidade de oferecer tarifas mais baratas para os consumidores o que pode acarretar na conquista de novos clientes, ademais a otimização desses processos possibilita um menor índice de perda de prazos e a consequente satisfação dos usuários dos serviços de entregas (MELO; FILHO, 2001).

Muitas são as abordagens para resolver o PCV, variando de algoritmos baseados no funcionamento da mente humana (GANESH; NARENDRAN, 2008) até heurísticas menos elaboradas. Além das heurísticas aplicáveis apenas ao PCV, existem meta-heurísticas, como *Simulated Annealing* (SONG; LEE; LEE, 2003), *Threshold Accepting* (DUECK; SCHEUER, 1990), *Genetic Algorithm* (RAZALI; GERAGHTY et al., 2011), *Tabu Search (TS)* (KNOX, 1994) e *Ant Colony Optimization (ACO)* (YANG et al., 2008), úteis em uma variedade de problemas de otimização (ANBUUDAYASANKAR; GANESH; MOHAPATRA, 2014).

Há também uma ferramenta disponibilizada pela *Google* conhecida como *OR-Tools*<sup>1</sup>. Ela é um pacote de *software* de código aberto para otimização. Esta ferramenta tem sido usada para gerar soluções em tempo aceitável de problemas nesta área inclusive para o Problema do Caixeiro Viajante (PERRON; FURNON, 2019-7-19).

A partir do cenário exposto, este trabalho pretende testar a viabilidade do uso do conjunto de *softwares Google OR-Tools* para gerar rotas otimizadas no distrito especial da AC-Russas, a partir do modelo básico do PCV, utilizando dados geográficos reais de entregas a serem efetuadas por esta unidade. Além disso comparar as rotas feitas atualmente com as geradas pela ferramenta, desse modo, verificando possíveis ganhos de eficiência ao utilizar esta abordagem para a roteirização dos distritos postais.

O presente trabalho está estruturado da seguinte maneira: a Seção 2 revisa todos os conceitos chaves relacionados à temática; já a Seção 3 apresenta alguns trabalhos relacionados a este estudo; a Seção 4 descreve a metodologia utilizada para desenvolvimento do estudo proposto; na Seção 5 são apontados os resultados. Na Seção 6 estão contidas as considerações finais, e por fim, as referências bibliográficas.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Definição do Problema do Caixeiro Viajante (PCV) e Variantes Existentes

O Problema do Caixeiro Viajante em sua versão clássica consiste em um vendedor que partindo de qualquer uma das cidades de um dado conjunto de cidades a visitar, deve passar por cada uma delas uma única vez tendo como destino final a cidade de onde partiu, de maneira que a rota realizada seja ótima, a partir de um critério pré-estabelecido, tal como, distância, tempo, custo, etc.

---

<sup>1</sup> <https://developers.google.com/optimization>

Podemos definir o problema matematicamente de duas formas equivalentes:

Dada a matriz de custos  $D = (d_{ij})$ , onde  $(d_{ij}) =$  custo de ir da cidade  $i$  para cidade  $j$ ,  $(i, j = 1, 2, \dots, n)$ , encontrar uma permutação  $P = (i_1, i_2, i_3, \dots, i_n)$  dos inteiros de 1 a  $n$  que minimize o somatório (LIN, 1965).

$$d_{i_1 i_2} + d_{i_2 i_3} + \dots + d_{i_n i_1}. \quad (1)$$

Outra abordagem de Programação Linear para o problema do caixeiro viajante simétrico  $(d_{ij} = d_{ji})$  que considera apenas parte das restrições lineares necessárias foi dada por (DANTZIG; FULKERSON; JOHNSON, 1954) esta técnica se mostrou eficaz em vários casos.

Dado um grafo  $G = (V, A)$ , onde  $V$  e  $A$ , são conjuntos formadores de um sistema em que  $V = \{v_1, \dots, v_n\}$  é o conjunto dos  $n$  vértices/nós do grafo e  $A = \{a_1, \dots, a_p\} \subset V \times V$  o conjunto das  $m$  arestas/arcos do grafo. Os autores definiram a variável  $x_{ij}$  como um arco entre os nós  $i$  e  $j$ , adquirindo apenas dois valores:

$$\begin{cases} x_{ij} = 1 & \text{quando percorrido o arco da cidade } i \text{ para a cidade } j, \\ x_{ij} = 0 & \text{caso não seja percorrido.} \end{cases}$$

Sendo  $c_{ij}$  a distância (ou o custo) relativo ao arco da cidade  $i$  para a cidade  $j$ . O custo total do percurso é dado por

$$\sum_{(i,j) \in A} c_{ij} x_{ij} \text{ (ou } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \text{ assumindo-se um grafo completo)}. \quad (2)$$

Ao acrescentarmos os arcos em falta com custo infinito, podemos considerar o grafo como sendo completo (isto é, considerar que  $(i, j) \in A, \forall i, j \in V$ ) sem perda de generalidade. Nesse caso, o problema é formulado da seguinte forma:

$$\text{minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (3)$$

$$\text{sujeito a } \sum_{j=1}^n x_{ij} = 1, i = 1, \dots, n, \quad (4)$$

$$\sum_{i=1}^n x_{ij} = 1, j = 1, \dots, n, \quad (5)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \forall S \subset V : 1 < |S| < n - 1, \quad (6)$$

$$x_{ij} \in \{0, 1\} \quad (7)$$

A restrição (4) garante que de cada cidade  $i$  partam apenas um arco para alguma cidade  $j$ , a restrição (5) garante que a cada cidade  $j$  chegue apenas um arco vindo de alguma cidade  $i$ . Garantindo desse modo, que toda cidade seja visitada, mas uma única vez. A restrição (6) assegura que não haja um conjunto de sub-rotas admitindo somente uma rota única como solução.

Uma variante importante do PCV é o Problema do Caixeiro Viajante Múltiplo (PCVM), proposto por Miler, Tucker e Zemlin, em que são considerados  $m$  caixeiros para percorrer todas as cidades. O objetivo é então determinar um conjunto de percursos para os  $m$  caixeiros viajantes de forma a percorrer todas as cidades, percorrendo no total a menor distância possível (OLIVEIRA, 2015).

Outra bem conhecida variante desse problema é o PCVDT (Problema do Caixeiro Viajante Dependente de Tempo) no qual são inseridas restrições de tempo para se visitar cada cidade (FOX; GAVISH; GRAVES, 1980). Este problema pode ser aplicado no caso das entregas postais, já que elas devem atender uma janela de tempo referente ao horário comercial e expediente dos carteiros.

Em geral, a utilização de critérios únicos é insuficiente para atender as demandas da vida real. Neste caso devemos utilizar uma abordagem multi-objetivo, desse modo, é possível minimizarmos simultaneamente o custo, tempo e distância, por exemplo. Dando origem a mais uma generalização do problema do caixeiro viajante o PCV multi-objetivo.

O PCV pode ser classificado como Simétrico (PCVS) se o custo relacionado entre as cidades for igual nos dois sentidos, sendo  $c$  o custo, se  $c(a, b) = c(b, a), \forall a, b$ , ou Assimétrico (PCVA) caso não seja. Caso os custos/distâncias entre as cidades cumpram a norma euclidiana, o PCV diz-se euclidiano (OLIVEIRA, 2015).

## 2.2 Métodos de Solução do PCV

Para a solução teórica de um modelo básico basta explorarmos todas as rotas possíveis e descobriremos qual delas nos fornece o menor valor para a função objetivo (por exemplo, distância percorrida em quilômetros). Se visitarmos  $(n - 1)$  cidades em sequência, sendo  $n$  o total de cidades a visitar, então o número total de possíveis rotas (circuitos fechados) é dado por uma permutação de  $(n - 1)!$ . Para um valor baixo de  $n$  essas rotas podem ser definidas explicitamente, mas aumentando-se o número de cidades visitadas o número total de rotas aumenta drasticamente. Por exemplo, para  $(n) = 6$  cidades, há 720 circuitos possíveis, para  $(n) = 12$ , há 479.001.600 e para  $(n) = 18$ , há inimagináveis  $6,402 \times 10^{15}$  possíveis rotas (FILIP; OTAKAR, 2011).

Diante do exposto, devido a complexidade do problema os métodos exatos, em geral, não representam uma alternativa viável, sendo comum o uso de heurísticas e meta-heurísticas, algoritmos que permitem obter soluções quase ótimas em tempo aceitável. Em seguida serão descritas alguns dos métodos mais utilizados para solução do PCV (FILIP; OTAKAR, 2011).

### 2.2.1 *Branch and Bound*

Basicamente o método consiste em dividir o conjunto de rotas em subconjuntos cada vez menores e calcular para eles um limite inferior relativo ao custo dentre eles. Os limites são utilizados para orientar o particionamento dos subconjuntos e, eventualmente, identificam uma rota ideal. Quando encontrado um subconjunto que contém uma única rota cujo custo é menor ou igual aos limites inferiores para todos os outros subconjuntos, essa rota é ideal. Os

subconjuntos de rotas são convenientemente representados como os nós de uma árvore e o processo de particionamento como uma ramificação de uma árvore, por isso o nome (LITTLE et al., 1963).

### 2.2.2 *Algoritmo de Clarke-Wright*

Este é um método heurístico derivado de um algoritmo de roteamento mais geral proposto por Clarke e Wright. Para a solução de um PCV, começamos com uma pseudo rota em que uma cidade escolhida arbitrariamente é o *hub* e o vendedor retorna ao *hub* após cada visita a outra cidade. (Em outras palavras, começamos com um multigrafo em que cada vértice não *hub* é conectado por duas arestas ao *hub*). Para cada par de cidades fora do centro, seja a economia o valor pelo qual a rota seria reduzida se o vendedor fosse diretamente de uma cidade para a outra, ignorando o centro. Agora procedemos de forma análoga ao algoritmo *Greedy*. Percorremos os pares de cidades não *hub* em ordem decrescente de economia, realizando o desvio, desde que não crie um ciclo de vértices não *hub* ou faça com que um vértice não *hub* se torne adjacente a mais de dois outros vértices não *hub*. O processo de construção termina quando apenas duas cidades não *hub* permanecem conectadas ao *hub*, caso em que temos uma rota verdadeira (CHAUHAN; GUPTA; PATHAK, 2012).

### 2.2.3 *Algoritmo de Otimização da Colônia de Formigas (ACO)*

Este algoritmo baseia-se no comportamento de um formigueiro na busca por fontes de alimento. Em geral as formigas buscam alimento de forma totalmente aleatória, quando encontram retornam para a colônia deixando uma trilha de feromônios que as outras formigas podem usar para se orientar, esse feromônio tem um tempo de evaporação e vai ficando mais fraco com o passar do tempo, porém se outra formiga encontra a trilha o feromônio é fortalecido e permite que outras formigas localizem o alimento mais próximo. No ACO este comportamento é replicado para encontrar as rotas mais curtas entre os nós de um grafo. Cada formiga busca por uma rota válida até que todas as possibilidades se esgotem, no final a trilha com a maior quantidade de feromônio é o valor da heurística para rota ótima mais provável. Mais sobre esse algoritmo e sua descrição matemática pode ser encontrado em (JALALI; AFSHAR; MARINO, 2005).

## 2.3 *OR-Tools*

*OR-Tools* é um *software* de código aberto para otimização combinatória, que busca encontrar a melhor solução para um problema a partir de um conjunto muito grande de soluções possíveis. Entre estes problemas podemos destacar: Roteamento de Veículos (encontre rotas ideais para frotas de veículos que coletam e entregam pacotes com restrições (por exemplo, “este caminhão não pode conter mais de 20.000 libras” ou “todas as entregas devem ser feitas dentro de uma janela de duas horas”); Agendamento (encontre o agendamento ideal para um conjunto complexo de tarefas, algumas das quais precisam ser executadas antes de outras, em um conjunto

fixo de máquinas ou outros recursos); Embalagem de caixas (embale o maior número possível de objetos de vários tamanhos em um número fixo de caixas com capacidades máximas).

Na maioria dos casos, problemas como esses têm um grande número de soluções possíveis muitas para um computador pesquisar todas elas. Para superar isso, o *OR-Tools* usa algoritmos no estado da arte para restringir o conjunto de buscas, a fim de encontrar uma solução ótima (ou próxima da ótima) (PERRON; FURNON, 2019-7-19).

Para este trabalho utilizaremos a biblioteca especializada em problemas de roteamento, a qual oferece um conjunto de opções com vários parâmetros configuráveis que serão usados e descritos a seguir.

### 2.3.1 *First Solution Strategy*

O *First Solution Strategy* é o método usado pelo solucionador para encontrar uma solução inicial, ele contém um conjunto de opções, dentre elas as seguintes usadas neste estudo: *AUTOMATIC*, permite que o solucionador detecte qual estratégia usar de acordo com o modelo que está sendo resolvido; *SAVINGS*, algoritmo de poupança (Clarke e Wright)(CLARKE; WRIGHT, 1964); *CHRISTOFIDES*, algoritmo Christofides (na verdade, uma variante do algoritmo Christofides usando uma correspondência maximizada em vez de uma correspondência máxima, o que não garante o fator  $3/2$  da aproximação em um caixeiro viajante métrico); funciona em modelos genéricos de roteamento de veículos, estendendo uma rota até que nenhum nó possa ser inserido nela (CHRISTOFIDES, 1976); *PARALLEL\_CHEAPEST\_INSERTION*, Constrói iterativamente uma solução inserindo o nó mais barato em sua posição mais barata; o custo de inserção é baseado na função de custo do arco; *GLOBAL\_CHEAPEST\_ARC*, conecta iterativamente dois nós que produzem o segmento de rota mais barato (PERRON; FURNON, 2019-7-19).

### 2.3.2 *Local Search Options*

A seguir são apresentadas as opções para as estratégias de busca local (também chamadas de meta-heurísticas) usadas no estudo: *GUIDED\_LOCAL\_SEARCH*, usa a busca local guiada para escapar dos mínimos locais, esta é geralmente a meta-heurística mais eficiente para roteamento de veículos; *TABU\_SEARCH*, usa busca tabu para escapar dos mínimos locais (PERRON; FURNON, 2019-7-19).

## 3 TRABALHOS RELACIONADOS

Nesta seção, são apresentados alguns trabalhos que, no contexto de entregas e logística, se utilizam do PCV para formular problemas reais da área e buscar soluções aplicáveis a essas tarefas, propondo novos métodos ou testando os já existentes, mas ainda não utilizados para o problema alvo deste estudo.

Em (SHI et al., 2020) é proposto um algoritmo baseado no PCV para planejamento de caminho de logística terminal. Devido as particularidades do problema de entrega foram



utilizadas técnicas de Programação Dinâmica, pois essa estratégia permite que o problema seja dividido em várias etapas inter-relacionadas possibilitando que a otimização das decisões ocorram em cada uma dessas fases. Em geral, se tratando de entregas as decisões tomadas têm relação com o tempo, desse modo, o estado atual afeta diretamente o próximo daí o uso de uma abordagem "dinâmica", já que o processo para obter solução está em constante mudança. O trabalho apresenta uma equação que atende às especificidades dos processos de coleta e entrega, e almeja fornecer uma solução que sirva de base teórica para um novo modelo de entrega inteligente, contudo o algoritmo carece de comprovação na prática logística para comprovar sua viabilidade, segundo os próprios autores.

No trabalho de (SANGUANSAT, 2019) o autor faz uso da API<sup>2</sup> de Matriz de distâncias do *Google* para obter dados em tempo real e utiliza esses dados em conjunto com o *Google OR-Tools*, uma biblioteca de código aberto para problemas de otimização, para resolver três tipos de problemas de roteamento de veículos (PVR), o clássico, onde apenas a quantidade de veículos e os endereços são considerados, o PRVC, no qual são acrescentadas restrições de capacidade dos veículos e demanda dos clientes, por fim o PRVC-JT que obedece às mesmas restrições do anterior, porém com restrições de janelas de tempo para cada cliente.

São apresentados os resultados dos experimentos com dados da cidade de Bangkok, para instâncias com trinta endereços. A vantagem da abordagem de dados em tempo real é que permite uma adequação das rotas para evitar vias que não possam ser utilizadas em determinados dias, no caso de estarem em obras, por exemplo. Levando em consideração o tempo para computação das soluções o estudo apresentou um resultado positivo se mostrando viável para aplicação em problemas práticos da vida real, mas está sujeito a melhorias principalmente na aquisição dos dados de forma mais eficiente.

Na linha de aplicação do PCV para otimização no serviço postal e de entregas o trabalho de (FERREIRA et al., 2017) traz a aplicação de um conjunto de algoritmos para otimização do percurso de distribuição de encomendas dos Correios na cidade de Coromandel/MG. Para elaboração do estudo foi utilizado o *Open Street Map* uma base de dados geográficos gratuitos mantida de forma colaborativa, uma heurística baseado em um algoritmo genético em conjunto com o Algoritmo de Dijkstra usando a linguagem MATLAB. A solução proposta apresentou rotas condizentes em custos de percurso com as mesmas elaboradas por profissionais experientes, porém não apresenta a eficiência esperada em termos de desempenho computacional não podendo ser aplicada a um número grande de pontos de entrega.

Ainda no contexto de serviços postais (JI; CHEN, 2007) nos apresenta um estudo sobre otimização de rotas de veículos no serviço postal de Hong Kong. Neste trabalho a partir de um PRV é proposto um modelo de programação linear de inteiros visando a maximização de recursos e minimização de custos de operação. Para lidar com a complexidade é utilizada uma ferramenta comercial *CPLEX*, atualmente na versão 20.1. O trabalho se concentra em elaborar rotas para coleta de correspondências em trinta e quatro pontos de operação espalhados pela ilha,

<sup>2</sup> <https://maps.googleapis.com/maps/api/distancematrix/>

obedecendo as restrições de minimização da distância, capacidade dos veículos e demanda de cada ponto de coleta. Os resultados apresentados sugerem que o modelo proposto traz benefícios aos Correios de Hong Kong, com ganhos em eficiência e produtividade.

## 4 METODOLOGIA

Para esta pesquisa de natureza descritiva e exploratória, foi usada uma abordagem quantitativa para análise dos dados, aplicados ao caso de estudo na agência de correios de Russas-CE. A etapa inicial consistiu na modelagem do problema a partir da base teórica previamente apresentada, usando os recursos disponíveis pelas ferramentas utilizadas, seguindo pelo método de aquisição, preparação e passagem dos dados. Na sequência é apresentada a forma de condução do experimento no caso estudado e por fim, são apresentadas as métricas usadas para mensurar a eficácia em função dos objetivos de minimização escolhidos. A Figura 1 apresenta um fluxo básico da metodologia e os meios usados em cada fase.



Figura 1 – Fluxo básico da metodologia.

### 4.1 Modelagem do Problema

Para Modelagem do Problema levou-se em consideração as características e condições específicas da entrega de encomendas no distrito especial na agência de correios de Russas-CE,

que em geral, representa o funcionamento das entregas nas demais agências de correios com distribuição, podendo haver diferenças pontuais para outros tipos de unidades operacionais.

Na AC-Russas o distrito especial abrange toda a área com distribuição na cidade. O veículo utilizado possui a capacidade de transportar toda a carga diária da agência, desse modo, o carteiro ao elaborar sua rota, o faz de modo a acomodar todos os objetos de uma só vez no compartimento destinado a carga. O tempo para entrega corresponde ao horário comercial com o intervalo para almoço de uma hora, contudo não há a necessidade de levar em consideração a parada para almoço pois o carteiro não leva em conta o local de almoço na elaboração da rota. Os correios ainda não realizam entregas com horário agendado. Dadas essas condições e a agência contando apenas com um veículo ao tempo da coleta dos dados, optou-se por um modelo básico sem necessidade de inserir restrições de tempo tão pouco de capacidade do veículo, sendo o problema do caso em estudo resumido a versão mais geral do PCV.

Para implementação de um programa básico capaz de resolver instâncias do problema foi utilizado o *Google Colab*<sup>3</sup> em conjunto com a linguagem de programação *Python* na versão 3.7 e o pacote *OR-Tools*. A escolha deste conjunto de ferramentas se deu pela facilidade de instalação, configuração e flexibilidade para utilizar o *OR-Tools* e *Python* no ambiente do *colab*, não sendo necessário utilizar um *hardware* dedicado de alto desempenho para a tarefa.

## 4.2 Aquisição e Pré-processamento dos Dados

A coleta dos dados foi realizada na AC-Russas em um intervalo de dez dias úteis consecutivos junto ao carteiro responsável pela distribuição no distrito especial. Os dados coletados consistem em dez rotas, sendo a menor uma sequência de cinquenta e dois endereços e a maior contém um total de oitenta e três endereços. Excluindo-se os endereços repetidos, foram coletados um total de quinhentos e vinte e sete endereços de entrega. O empregado já trabalha há mais de dez anos na entrega de encomendas na referida agência dos Correios, sendo assim, possui vasto conhecimento sobre as vias e endereços a serem visitados, isto aliado a sua experiência fornece condições para que elabore rotas condizentes com a maioria das rotas geradas por outros carteiros.

### 4.2.1 Coleta das Rotas

Os Correios utilizam o Sistema de Rastreamento de Objetos (SRO), após a preparação da rota pelo carteiro cada objeto é lançado por meio do seu endereço e é gerada uma Lista de Objetos Entregues ao Carteiro (LOEC) que é carregada na versão móvel do SRO que roda nos *Smartphones* usados para baixa dos objetos. Contudo a ordem dos objetos na LOEC nem sempre representa a ordem correta de entrega pelos carteiros. Para evitar discordâncias entre as informações coletadas, o registro das rotas foi realizado da seguinte forma: após a triagem da carga, já com os objetos dispostos na Ordem de entrega, utilizou-se um dispositivo de gravação de áudio no qual foi feito o registro dos endereços contidos em cada encomenda, posteriormente

---

<sup>3</sup> <https://colab.research.google.com/>

esses registros foram transcritos para uma planilha no *Google Sheets*<sup>4</sup>, mantendo-se a ordem em que foram coletados (ordem de entrega).

#### 4.2.2 Validação dos Endereços

Por utilizarmos a base de dados do *Google Maps* e por Russas ainda não contar com CEPs diferentes para cada logradouro, houve a necessidade de tratamento nos endereços antes de realizar as consultas para construção da matriz de distâncias, pois os endereços contidos na base de dados muitas vezes diferem dos usados pelos clientes para endereçar os pacotes. Havendo então divergências nos nomes das ruas, números das casas, entre outras, que impedem a aplicação sem o devido tratamento. Para assegurar que os endereços correspondiam aos locais reais de entrega cada endereço coletado foi checado por meio de busca manual na plataforma de mapas usada. Havendo divergência o endereço era pinçado no mapa e confirmado visualmente pelo carteiro no *Google Street view*<sup>5</sup>, sendo então utilizada a designação constante na base de dados ou a coordenada geográfica do ponto na ausência desta. Outro tratamento necessário foi a codificação dos endereços para um formato aceito em URLs, substituindo os caracteres não permitidos ou reservados.

#### 4.3 Construção das Matrizes de Distância e Duração

Na passagem dos dados na implementação do programa é necessário fornecer uma matriz de distâncias como a mostrada abaixo, ou matriz de duração caso o objetivo de minimização escolhido seja o tempo gasto para percorrer a rota. A matriz de distância é uma matriz cuja entrada  $(i, j)$  é a distância da localização  $i$  à localização  $j$  em metros, onde os índices da matriz correspondem às localizações de cada ponto a ser visitado. A matriz de duração segue a mesma linha, mas ao invés de distância traz uma estimativa da duração do percurso de  $i$  à  $j$ .

```

1  data = {}
2  data[ 'distance_matrix' ] = [
3      [0, 2451, 713, 1018, 1631 ],
4      [2451, 0, 1745, 1524, 831 ],
5      [713, 1745, 0, 355, 920 ],
6      [1018, 1524, 355, 0, 700 ],
7      [1631, 831, 920, 700, 0 ],]
8      # yapf: disable
9  data[ 'num_vehicles' ] = 1
10 data[ 'depot' ] = 0
11 return data

```

Neste exemplo, a matriz de distâncias é definida explicitamente no programa. Também é possível usar uma função para calcular distâncias entre locais: por exemplo, a fórmula euclidiana para a distância entre pontos no plano. No entanto, ainda é mais eficiente calcular previamente

<sup>4</sup> <https://docs.google.com/spreadsheets/>

<sup>5</sup> <https://www.google.com/streetview/>

todas as distâncias entre os locais e armazená-las em uma matriz, em vez de computá-las em tempo de execução.

O *Google* disponibiliza uma API de matriz de distâncias, esse serviço oferece a possibilidade de consultarmos a distância entre dois pontos e nos devolve a duração da viagem e a distância em metros. Os dados são disponibilizados em formato JSON como no exemplo abaixo.

```

1 {
2   "destination_addresses": ["San Francisco , CA, USA" ],
3   "origin_addresses": ["Vancouver , BC, Canada" ],
4   "rows": [{ "elements": [{
5     "distance": { "text": "1,723 km", "value": 1723247 },
6     "duration": { "text": "3 days 21 hours", "value": 335356 },
7     "status": "OK",
8   }],
9 }

```

Após a coleta e validação dos endereços, foi utilizado um *script Python* para consulta das distâncias e durações usadas para construção das matrizes para cada uma das dez rotas, abaixo temos um exemplo de requisição feito diretamente para a URL, neste exemplo são usadas as coordenadas, mas podem ser utilizados os endereços de forma literal.

```

1 https://maps.googleapis.com/maps/api/distancematrix/json
2   ?destinations=40.659569%2C-73.933783%7C40.729029%2C-73.851524%7C40
3   .6860072%2C-73.6334271%7C40.598566%2C-73.7527626
4   &origins=40.6655101%2C-73.89188969999998
5   &key=YOUR_API_KEY

```

A API possui uma série de parâmetros que permitem definir o tipo de veículo utilizado, tipos de via a serem evitadas o sistema de medidas para as distâncias, opções de tráfego, entre outros, que permitem um ajuste ao modelo de problema a ser resolvido, na construção das nossas matrizes utilizamos a configuração padrão que vem ajustada para o modo *driving* devolvendo distâncias usando a rede rodoviária, além disso, para as distâncias foi utilizado o sistema métrico de unidades, não foi configurado nenhum parâmetro extra.

#### 4.4 Geração das Rotas

Após as etapas de escolha do modelo de PCV, coleta e validação dos dados, e geração das matrizes de distância e duração, a fase seguinte foi a resolução das instâncias usando o *OR-tools*. Como já mencionado foi desenvolvido um programa para solução do modelo básico do PCV, usando *Python* e rodando no *Google colab*. O programa recebe como entrada a matriz de distância ou duração da rota e retorna como resultado o valor da função objetivo de minimização escolhida, neste caso a distância total da rota ou o tempo estimado para o percurso. O programa também retorna como saída uma rota indicando a sequência de entrega dos clientes para a rota gerada, sendo esta representada pelos índices da ordem dos endereços usados para gerar a matriz de distâncias.

Para solução das dez instâncias, após uma análise prévia a fim de descobrir quais métodos ofereciam as melhores soluções, foram aplicados os métodos com as opções presentes na Seção 2.3 sendo estes:

- *First solution strategy*:
  - *AUTOMATIC*;
  - *SAVINGS*;
  - *CHRISTOFIDES*;
  - *PARALLEL\_CHEAPEST\_INSERTION*;
  - *GLOBAL\_CHEAPEST\_ARC*.
  
- *LocalSearchMetaheuristic*:
  - *GUIDED\_LOCAL\_SEARCH*;
  - *TABU\_SEARCH*.

#### **4.4.1 Objetivo de Minimização**

Na solução do PCV, em geral, utiliza-se como objetivo a minimização de custos monetários, tempo gasto para percorrer a rota ou distância total percorrida ao final da rota. No caso em estudo, por se tratar de um contexto de entregas optamos por adotar como objetivo de minimização a distância total da rota e o tempo para realizar o percurso, que indiretamente pode nos fornecer estimativas de custos.

Para o objetivo tempo gasto para percorrer a rota, aplicamos o método padrão *First solution strategy* na opção *AUTOMATIC* para dez instâncias, já para a distância total da rota foram aplicados todos os métodos descritos anteriormente na Seção 2.3

#### **4.5 Avaliação de Desempenho**

A escolha dos critérios para avaliação do estudo leva em conta os ganhos reais que o uso da ferramenta poderá proporcionar, os mais evidentes são, a diminuição nas distâncias percorridas pelos carteiros e ou o tempo gasto para realizar as entregas. Para escolha foi levado em consideração a disponibilidade e relevância dos dados, desse modo, optou-se por três parâmetros, a otimização da distância total da rota, o tempo estimado para percorrer a rota e por fim a atratividade visual da rota. Todos os dados foram comparados com as rotas geradas pelos carteiros utilizando as Matrizes de distância e duração obtidas na API.

##### **4.5.1 Atratividade Visual**

A atratividade visual é um critério que se provou eficaz para facilitar a implementação prática e a colaboração dos setores de uma organização envolvidos nos níveis de planejamento e operacional no transporte. Como conclui (ROSSIT et al., 2019) em sua revisão de literatura,

onde nos apresenta um conjunto de métricas aplicáveis a vários problemas de roteamento de veículos, elas são classificadas em quatro categorias: compactação; proximidade; sobreposição e cruzamentos de rotas; e complexidade. Desse modo, optamos por utilizar duas das recomendadas pelos autores devido sua facilidade de implementação e adequação ao caso em estudo:

- Métrica de compactação descrita em (MATIS, 2008)

$$COMP_i = \frac{AvgDist_i}{Avg\ max\ Dist_i}, \quad (8)$$

onde  $AvgDist_i$  é a distâncias média entre dois clientes consecutivos na rota  $i$ , e  $AvgMaxDist_i$  é a distância média das 20% maiores distâncias entre dois clientes consecutivos na rota  $i$ . Quanto maior o valor mais compacta é a solução.

- Número de cruzamentos em uma mesma rota como em (POOT; KANT; WAGELMANS, 2002). Número total de cruzamentos entre os arcos pertencentes à mesma rota  $I$ . Neste caso quanto menor o número de cruzamentos melhor é a solução.

## 5 RESULTADOS

Nesta Seção são apresentados os resultados obtidos a partir da execução das instâncias de acordo com a modelagem prévia para cada um dos métodos utilizados, incluindo a solução do carteiro. Apresentamos também os resultados das métricas escolhidas para avaliação da atratividade visual de cada uma das rotas geradas. Na Tabela 1 apresentamos os valores das distâncias totais em metros, e a quantidade de clientes para cada uma das rotas com o respectivo método de solução, em todos os casos houve uma diminuição no percurso total das rotas.

N° Clientes	Carteiro	FIRST SOLUTION STRATEGY					LOCAL SEARCH OPTIONS	
		AUTOMATIC	SAVINGS	CHRISTOFIDES	PARALLEL_CHEAPEST_INSERTION	GLOBAL_CHEAPEST_ARC	GUIDED_LOCAL_SEARCH	TABU_SEARCH
67	42363	33068	31624	32316	31345	32836	<b>30882</b>	30917
57_a	37147	28851	29259	28474	29701	28496	<b>28055</b>	28620
57_b	36933	31188	29053	28998	29792	28625	<b>28521</b>	28851
52	38252	31041	30280	29530	30295	31251	<b>29456</b>	29830
61	37766	27597	28817	28175	30232	27213	<b>27139</b>	27597
64	41236	30069	29726	30665	31401	30128	<b>29470</b>	29453
60	34665	29417	27723	28350	33494	29417	<b>27463</b>	27723
56	35965	28534	26911	26926	27624	27555	<b>25851</b>	26765
83	46533	37015	33897	35897	34887	34231	<b>32706</b>	35166
69	39845	29969	29963	29122	32222	30840	<b>29059</b>	29644

Tabela 1 – Distância total das rotas para cada método de solução.

A Tabela 2 traz a diferença percentual entre as distâncias totais da rota do carteiro e as rotas geradas usando o *OR-tools*, para cada um dos métodos usados. No geral houve uma diminuição média de 23,16% o que pode significar uma economia relevante nos custos operacionais.

Nºclientes	FIRST SOLUTION STRATEGY					LOCAL SEARCH OPTIONS	
	AUTOMATIC	SAVINGS	CHRISTOFIDES	PARALLEL_CHEAPEST_INSERTION	GLOBAL_CHEAPEST_ARC	GUIDED_LOCAL_SEARCH	TABU_SEARCH
66	-21.94	-25.35	-23.72	-26.01	-22.49	<b>-27.10</b>	-27.02
57_a	-22.33	-21.23	-23.35	-20.04	-23.29	<b>-24.48</b>	-22.95
57_b	-15.56	-21.34	-21.48	-19.34	-22.49	<b>-22.78</b>	-21.88
52	-18.85	-20.84	-22.80	-20.80	-18.30	<b>-22.99</b>	-22.02
61	-26.93	-23.70	-25.40	-19.95	-27.94	<b>-28.14</b>	-26.93
64	-27.08	-27.91	-25.64	-23.85	-26.94	<b>-28.53</b>	-28.57
60	-15.14	-20.03	-18.22	-3.38	-15.14	<b>-20.78</b>	-20.03
56	-20.66	-25.17	-25.13	-23.19	-23.38	<b>-28.12</b>	-25.58
83	-20.45	-27.15	-22.86	-25.03	-26.44	<b>-29.71</b>	-24.43
69	-24.79	-24.80	-26.91	-19.13	-22.60	<b>-27.07</b>	-25.60

Tabela 2 – Diferença percentual entre as rotas do carteiro e as geradas usando *OR-tools*.

A Tabela 3 apresenta uma estimativa do tempo, valores em minutos, para percorrer as rotas, tendo como base a matriz de duração fornecida pela API. Neste caso aplicamos apenas o Método *AUTOMATIC*, onde a ferramenta escolhe qual método usar. Os valores apresentam uma economia semelhante em termos percentuais, comparado com os valores obtidos quando o objetivo de minimização escolhido é a distância.

Após análise dos resultados podemos notar que as maiores diferenças se apresentam nas instâncias com uma maior quantidade de pontos de entrega, como evidenciado na tabela 2, cuja instância com oitenta e três clientes usando *GUIDED\_LOCAL\_SEARCH* apresenta uma redução na distância total da rota de 29,71%, em relação à gerada pelo carteiro. Quando o objetivo de minimização é o tempo, foi observado o mesmo comportamento, como visto na Tabela 3, onde usando o método *FirstSolutionStrategy.AUTOMATIC*, a instância com oitenta e três clientes, teve redução de 23,49% na estimativa do tempo total da rota. Podemos inferir que essa diferença se dá pela maior dificuldade do carteiro em lidar com instâncias grandes, já que os métodos apresentam desempenho semelhante para as demais instâncias.

Nºclientes	Carteiro	FIRST SOLUTION STRATEGY .AUTOMATIC	Diferença(%)
66	120	94.76	<b>-21.03</b>
57_a	98.13	83.65	<b>-14.76</b>
57_b	108.06	90.25	<b>-16.48</b>
52	101.56	86.71	<b>-14.62</b>
61	109.85	87.4	<b>-20.44</b>
64	122.53	95.7	<b>-21.90</b>
60	101.15	86.6	<b>-14.38</b>
56	106.43	85.26	<b>-19.89</b>
83	129.2	98.85	<b>-23.49</b>
69	115.38	92.11	<b>-20.17</b>

Tabela 3 – Estimativa da duração das rotas e diferença percentual.

A Tabela 4 traz os valores das métricas de atratividade visual para cada uma das instâncias resolvidas usando como objetivo a minimização das distâncias das rotas, para cada um dos métodos utilizados. Em geral, as métricas apresentaram resultados melhores ou iguais aos das rotas dos carteiros, o que indica que as rotas geradas estão de acordo em termos visuais, tanto



em compactação quanto em quantidades de cruzamentos. A Figura 2 traz lado a lado a rota do carteiro e uma obtida usando *GUIDED\_LOCAL\_SEARCH* para a instância com cinquenta e dois clientes.

N° Clientes	FIRST SOLUTION STRATEGY												LOCAL SEARCH OPTIONS			
	Carteiro		AUTO-MATIC		SAVINGS		CHRIS-TOFIDES		PARALLEL_CHEAPEST_INSERTION		GLOBAL_CHEAPEST_ARC		GUIDED_LOCAL_SEARCH		TABU_SEARCH	
	Cp	Cr	Cp	Cr	Cp	Cr	Cp	Cr	Cp	Cr	Cp	Cr	Cp	Cr	Cp	Cr
67	0.38	9	0.42	8	0.42	5	0.43	8	0.45	4	0.42	5	0.43	5	0.43	9
57_a	0.36	5	0.39	4	0.4	11	0.39	3	0.41	4	0.38	3	0.39	3	0.38	2
57_b	0.38	8	0.43	6	0.44	2	0.45	3	0.43	3	0.43	1	0.43	1	0.43	2
52	0.34	11	0.35	3	0.36	2	0.36	2	0.36	4	0.34	3	0.37	0	0.36	2
61	0.38	2	0.43	2	0.46	4	0.44	4	0.45	4	0.43	2	0.44	2	0.43	2
64	0.37	10	0.42	2	0.41	5	0.41	2	0.42	7	0.42	5	0.44	1	0.43	5
60	0.41	8	0.48	6	0.49	6	0.51	8	0.51	15	0.48	6	0.52	4	0.49	5
56	0.38	11	0.47	8	0.47	1	0.46	6	0.47	3	0.47	4	0.46	1	0.46	3
83	0.33	12	0.36	21	0.36	8	0.33	16	0.36	14	0.35	12	0.35	12	0.36	18
69	0.35	10	0.39	5	0.41	5	0.41	5	0.41	6	0.42	3	0.4	3	0.41	5

Tabela 4 – Métricas de compactação (Cp) e cruzamento (Cr).



(a) Rota do carteiro. Distância total 38252 metros. 11 cruzamentos.



(b) Rota OR-tools. Distância total 29456 metros. 0 cruzamentos.

Figura 2 – Trajeto das rotas.

## 6 CONSIDERAÇÕES FINAIS

A partir do Problema da elaboração de rotas otimizadas para distritos postais, dado o aumento significativo nas últimas décadas da demanda por serviços de entrega, e o crescente uso da tecnologia para gerar soluções para problemas nos mais diversos campos da vida humana, o presente trabalho utilizou-se de um conjunto de *softwares* de código aberto para solução de instâncias reais do Problema do Caixeiro Viajante (PCV). Neste contexto buscou demonstrar de maneira experimental, a viabilidade do uso destes *softwares* para roteirização de entregas de encomendas, limitado ao caso da agência de Correios de Russas-CE, dentro de condições

replicáveis na maioria das unidades operacionais dos Correios e demais serviços de entrega, sendo bem sucedido nesta tarefa.

No decorrer deste estudo foram coletados e validados os dados das rotas geradas pelos carteiros, e desenvolvido um *script* em *Python* capaz de roteirizar os endereços coletados, com auxílio do *Google OR-tools*. Estes foram passos fundamentais, cumpridos com êxito, para provar a hipótese proposta.

Ao analisarmos os dados obtidos por meio dos experimentos que compõem os objetivos específicos desta obra, podemos confirmar que mostra-se possível o emprego da alternativa proposta, demonstrado pelos resultados obtidos na elaboração das dez instâncias resolvidas, que chegaram a resultados melhores que os obtidos pelos métodos atualmente utilizados.

Os resultados deste trabalho indicam a viabilidade técnica e operacional do modelo proposto e demonstra a eficiência no uso de ferramentas *open source* para automatizar a geração de rotas nos Correios do Brasil e demais serviços logísticos, a utilização do método proporciona avanços na prestação destes serviços.

Cabe ressaltar, contudo, que o serviço de entregas possui os cenários mais diversos e que em suma não é possível generalizar os resultados para além das restrições aqui delimitadas. Desse modo, as condições em que o estudo foi realizado são fatores cruciais para avaliarmos os dados obtidos por este trabalho.

Podemos citar como limitações deste trabalho o fato de não ter sido realizado o emprego da metodologia para gerar rotas para um distrito convencional, onde a quantidade de clientes aumenta enormemente, também a aplicação em problemas com mais de um veículo, PCV múltiplo, há ainda a necessidade de comparação com as rotas geradas usando o Traffic Management System (TMS), ficando como trabalhos futuros a realização de tais experimentos. Uma dificuldade encontrada durante o processo foi a aquisição e passagem dos dados de endereços, sendo necessário uma melhora neste aspecto para operacionalizar a implantação.

## REFERÊNCIAS

- ANBUUDAYASANKAR, S.; GANESH, K.; MOHAPATRA, S. Survey of methodologies for tsp and vrp. In: **Models for Practical Routing Problems in Logistics**. [S.l.]: Springer, 2014. p. 11–42.
- BELLMORE, M.; NEMHAUSER, G. L. The traveling salesman problem: a survey. **Operations Research**, INFORMS, v. 16, n. 3, p. 538–558, 1968.
- CAPTERRA. **Quick commerce: 95% dos consumidores gostariam de reduzir os prazos de entrega**. 2022. <https://www.capterra.com.br/blog/2365/quick-commerce>.
- CHAUHAN, C.; GUPTA, R.; PATHAK, K. Survey of methods of solving tsp along with its implementation using dynamic programming approach. **International journal of computer applications**, Citeseer, v. 52, n. 4, 2012.
- CHRISTOFIDES, N. **Worst-case analysis of a new heuristic for the travelling salesman problem**. [S.l.], 1976.

- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations research**, INFORMS, v. 12, n. 4, p. 568–581, 1964.
- CNT. **Pesquisa De Impacto no Transporte – Covid-19, 6ª rodada**. Confederação Nacional do Transporte, 2021. Disponível em: <<https://www.cnt.org.br/pesquisas>>.
- CORREIOS. **relatorio-integrado-correios-2021**. 2022. Disponível em: <<https://www2.correios.com.br/arquivos/PrestacaoDeContasAnuais/2021/relatorio-integrado-correios-2021.pdf>>.
- DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. **Journal of the operations research society of America**, INFORMS, v. 2, n. 4, p. 393–410, 1954.
- DUECK, G.; SCHEUER, T. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. **Journal of computational physics**, Elsevier, v. 90, n. 1, p. 161–175, 1990.
- EBIT. **44ª ed. Webshoppers**. 2021. Realizado pela Ebit desde 2001, o Webshoppers é o relatório de maior credibilidade sobre o comércio eletrônico brasileiro e é considerado a principal referência para os profissionais do segmento. Disponível em: <<https://www.ebit.com.br/webshoppers>>.
- FERREIRA, M. F. et al. Otimização do percurso de distribuição de encomendas dos correios na cidade de coromandel-mg. **Revista Agroveterinária, Negócios e Tecnologias**, v. 2, n. 2, p. 42–55, 2017.
- FILIP, E.; OTAKAR, M. The travelling salesman problem and its application in logistic practice. **WSEAS Transactions on Business and Economics**, v. 8, n. 4, p. 163–173, 2011.
- FLOOD, M. M. The traveling-salesman problem. **Operations research**, INFORMS, v. 4, n. 1, p. 61–75, 1956.
- FOX, K. R.; GAVISH, B.; GRAVES, S. C. An n-constraint formulation of the (time-dependent) traveling salesman problem. **Operations Research**, INFORMS, v. 28, n. 4, p. 1018–1021, 1980.
- GANESH, K.; NARENDRAN, T. Taste: a two-phase heuristic to solve a routing problem with simultaneous delivery and pick-up. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 37, n. 11-12, p. 1221–1231, 2008.
- GUTIN, G.; PUNNEN, A. P. **The traveling salesman problem and its variations**. [S.l.]: Springer Science & Business Media, 2006. v. 12.
- ILAVARASI, K.; JOSEPH, K. S. Variants of travelling salesman problem: A survey. In: IEEE. **International Conference on Information Communication and Embedded Systems (ICICES2014)**. [S.l.], 2014. p. 1–7.
- JALALI, M. R.; AFSHAR, A.; MARINO, M. A. Ant colony optimization algorithm(aco); a new heuristic approach for engineering optimization. **WSEAS Transactions on Information Science and Applications**, Citeseer, v. 2, n. 5, p. 606–610, 2005.
- JI, P.; CHEN, K. The vehicle routing problem: the case of the hong kong postal service. **Transportation Planning and Technology**, Taylor & Francis, v. 30, n. 2-3, p. 167–182, 2007.
- KNOX, J. Tabu search performance on the symmetric traveling salesman problem. **Computers & Operations Research**, Elsevier, v. 21, n. 8, p. 867–876, 1994.

LIN, S. Computer solutions of the traveling salesman problem. **Bell System Technical Journal**, Wiley Online Library, v. 44, n. 10, p. 2245–2269, 1965.

LITTLE, J. D. et al. An algorithm for the traveling salesman problem. **Operations research, INFORMS**, v. 11, n. 6, p. 972–989, 1963.

MATIS, P. Decision support system for solving the street routing problem. **Transport**, Taylor & Francis, v. 23, n. 3, p. 230–235, 2008.

MELO, A. C. d. S.; FILHO, V. J. M. F. Sistemas de roteirização e programação de veículos. **Pesquisa operacional**, SciELO Brasil, v. 21, n. 2, p. 223–232, 2001.

OLIVEIRA, A. F. M. d. A. **Extensões do problema do caixeiro viajante**. Tese (Doutorado) — Universidade d Coimbra, 2015.

PERRON, L.; FURNON, V. **OR-Tools**. 2019–7–19. Disponível em: <<https://developers.google.com/optimization/>>.

POOT, A.; KANT, G.; WAGELMANS, A. P. M. A savings based method for real-life vehicle routing problems. **Journal of the Operational Research Society**, Taylor & Francis, v. 53, n. 1, p. 57–68, 2002.

RAZALI, N. M.; GERAGHTY, J. et al. Genetic algorithm performance with different selection strategies in solving tsp. In: INTERNATIONAL ASSOCIATION OF ENGINEERS HONG KONG, CHINA. **Proceedings of the world congress on engineering**. [S.l.], 2011. v. 2, n. 1, p. 1–6.

ROSSIT, D. G. et al. Visual attractiveness in routing problems: A review. **Computers & Operations Research**, Elsevier, v. 103, p. 13–34, 2019.

SANGUANSAT, P. Developing applications for vehicle routing problems with real time data acquisition. **origins**, v. 13, n. 100.532713&destinations, p. 13–847881, 2019.

SHI, K. et al. The algorithm of terminal logistics path planning based on tsp problem. In: **2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)**. [S.l.: s.n.], 2020. p. 130–133.

SONG, C.-H.; LEE, K.; LEE, W. D. Extended simulated annealing for augmented tsp and multi-salesmen tsp. In: IEEE. **Proceedings of the International Joint Conference on Neural Networks, 2003**. [S.l.], 2003. v. 3, p. 2340–2343.

YANG, J. et al. An ant colony optimization method for generalized tsp problem. **Progress in Natural Science**, Elsevier, v. 18, n. 11, p. 1417–1422, 2008.