

# Estudo de Caso: Otimização de um Problema de Roteamento de Veículos através de heurísticas computacionais

## Autores:

- André Hutzler
- Felipe Morita Braga
- Fernando Machado dos Santos
- Gabriel Coletto da Silva
- Gabriel de Macedo Santos
- Giovanna Katsuki Murata
- Leonardo Kalid Guene

## Orientador:

- Tomaz Mikio Sasaki

## Resumo

Este trabalho apresenta um estudo de caso sobre a otimização de rotas de veículos para uma das regiões operacionais da Aegea Saneamentos S.A., a Águas do Rio, localizada no Rio de Janeiro. Como empresa dedicada ao saneamento básico, a Aegea enfrenta o desafio de garantir a eficiência na leitura de medidores, tendo como objetivo principal otimizar as rotas de leitura, garantir a precisão das medições e maximizar a eficiência do tempo e dos recursos disponíveis (AEGEA, 2019).

Neste contexto complexo, a otimização das rotas dos leitores de medidores é crucial para garantir que todas as residências sejam atendidas dentro de um período definido, respeitando as restrições operacionais e maximizando a eficiência da equipe. Para enfrentar esse problema, técnicas de otimização inspiradas em problemas clássicos, como o problema do Caixeiro Viajante (PORTOSIL, 2000), são exploradas e adaptadas à realidade operacional da Aegea.

Dentre os algoritmos considerados para essa otimização, destacam-se: Vizinhos Mais Próximos (IBM, 2024), algoritmos genéticos, Simulated Annealing (SCIENCEDIRECT, 2023), entre outros. Cada um desses métodos oferece diferentes abordagens para encontrar soluções eficientes, levando em consideração as características específicas da distribuição geográfica das residências e as demandas operacionais da empresa.

Este trabalho propõe uma análise comparativa desses algoritmos, buscando identificar a solução mais adequada para o problema em questão. Além disso, visa desenvolver uma plataforma prática e intuitiva para o planejamento e execução de rotas otimizadas, visando não apenas melhorar a eficiência operacional, mas também reduzir custos e garantir a satisfação do cliente.

A relevância deste estudo para a Aegea Saneamentos S.A. reside na sua capacidade de oferecer uma solução prática e inovadora para um desafio logístico fundamental. Com a otimização das rotas de leitura de medidores, espera-se não apenas aumentar a eficiência operacional, mas também melhorar a qualidade dos serviços prestados, contribuindo para a excelência do saneamento básico na região atendida.

É importante destacar que este estudo se baseia em extensa pesquisa prévia sobre problemas similares de otimização logística, que fornecem insights valiosos para o desenvolvimento de soluções eficazes e sustentáveis. Assim, este trabalho não só oferece uma resposta prática aos desafios enfrentados pela Aegea, mas também contribui para o avanço do conhecimento na área de otimização logística aplicada ao setor de saneamento básico e problemas de roteamento no geral.

## Introdução

Este artigo apresenta um estudo sobre a otimização das rotas de leituristas em uma das regiões operacionais da Aegea Saneamentos S.A., a Águas do Rio, localizada no Rio de Janeiro. Como uma empresa dedicada ao saneamento básico, a Aegea enfrenta o desafio de garantir a eficiência na leitura dos hidrômetros, **com o objetivo primordial de otimizar as rotas de leitura, garantindo a precisão das medições e maximizando a eficiência no uso do tempo e dos recursos disponíveis.**(AEGEA, 2019).

Nesse contexto complexo, a otimização das rotas dos leituristas é crucial para garantir que todas as residências sejam atendidas dentro de um prazo definido, respeitando as restrições operacionais e maximizando a eficiência da equipe. Para abordar essa questão, são exploradas técnicas de otimização inspiradas em problemas clássicos, como o do caixeiro viajante(PORTOSIL, 2000), adaptadas à realidade operacional da Aegea.

Dentre os algoritmos considerados para esta otimização, destacam-se o *Nearest Neighbors*(IBM, 2024), algoritmos genéticos, *Simulated Annealing*(SCIENCEDIRECT, 2023) dentre outros. Cada um desses métodos oferece abordagens distintas para encontrar soluções eficientes, levando em conta as características específicas da distribuição geográfica das residências e as demandas operacionais da empresa.

Este trabalho propõe uma análise comparativa desses algoritmos, buscando identificar a solução mais adequada para o problema em questão. Além disso, visa desenvolver uma plataforma prática e intuitiva para o planejamento e execução das rotas otimizadas, visando não apenas melhorar a eficiência operacional, mas também reduzir custos e garantir a satisfação dos clientes.

A relevância deste estudo para a Aegea Saneamentos S.A. reside na sua capacidade de oferecer uma solução prática e inovadora para um desafio logístico fundamental. Ao otimizar as rotas de leitura dos hidrômetros, espera-se não apenas aumentar a eficiência operacional, mas também aprimorar a qualidade dos serviços prestados, contribuindo para a excelência do saneamento básico na região atendida.

É importante salientar que este estudo se baseia em uma extensa pesquisa prévia sobre problemas similares de otimização logística, que fornecem *insights* valiosos para o desenvolvimento de soluções eficazes e sustentáveis. Assim, este trabalho não apenas oferece uma resposta prática para os desafios enfrentados pela Aegea, mas também contribui para o avanço do conhecimento na área de otimização logística aplicada ao setor de saneamento básico.

## Trabalhos relacionados

Nesta seção deste trabalho científico, foi realizada uma revisão bibliográfica de artigos científicos publicados previamente e que fornecem uma ampla gama de baseamento para a definição de uma metodologia de pesquisa, além de ampliarem a possibilidade de soluções cogitadas para a resolução da problemática. Assim, foram escolhidos três trabalhos diferentes (após pesquisa contínua) e que trazem resultados e algoritmos diferentes para solucionar problemas semelhantes.

### 1º Pesquisa relevante: Modelagem para o Problema de Roteamento de Veículos Fretados

No pressuposto inicial deste artigo, pretende-se realizar uma otimização das rotas para os leituristas, percorrendo a totalidade das casas presentes na concessão analisada e minimizando o número total de funcionários necessários para a realização de todos os trajetos mediante uma escala de tempo. Dada a especificidade da questão, a pesquisa tende à busca de propostas similares, que, com severas modificações práticas e de restritividade, pode ser aplicada no mesmo contexto. Este é um exímio caso no qual isto ocorre: realizou-se uma modelagem matemática para uma questão de roteamento de veículos fretados, na qual o mínimo de unidades de transporte deveria ser utilizada para buscar todos os colaboradores em suas devidas casas, retornando ao ponto inicial. Para além disso, esta amostra apresenta um referencial teórico excelente para a modelagem e entendimento do problema supracitado na introdução deste artefato.

#### Artigo Referenciado

**Título:** Modelagem para o Problema de Roteamento de Veículos Fretados

**Autor:** R. S. OLIVEIRA, C. T. L. S. GHIDINI, C. TOREZZAN, W. A. OLIVEIRA

## Principais Achados

Inicialmente, foi realizada uma ampla descrição e discussão de diferentes técnicas e teses computacionais. Estas podem ser utilizadas para a reavaliação das dinâmicas empregadas localmente para resolução do problema da AEGEA, haja vista que apresentam uma visão diferente de um ponto comum. A seção de modelagem matemática, por outro lado, se demonstrou extremamente relevante para comparação com a realizada pela equipe e referenciada em outro arquivo do mesmo repositório, apresentando restrições não imaginadas previamente e que, possivelmente, podem ser reaplicadas. Uma delas, por exemplo, foi uma verificação do possível custo exponencial baseado no aumento progressivo de restrições e o impacto disto, modificando uma restrição presente na literatura para o problema de roteamento de veículos e reduzindo o custo computacional.

Uma realização interessante para o problema que eles abordaram foi a proposição de geração da matriz de distâncias através de *Excel VBA* em conjunto com a *Google Matrix API*, além do mapeamento geográfico com a *HERE Technologies*, resolvendo um problema que dificultou de maneira excessiva a constituição algoritmo do problema: como garantir que as rotas respeitem os caminhos possíveis e não trespassem limites geográficos como quarteirões e pontes.

Uma característica interessante do artigo referenciado é que, além de toda a contextualização teórica e matemática realizada plenamente, foi instaurado um estudo de caso na cidade de Itumbiara, Goiás, com uma empresa real e seus 176 funcionários. Existem casos de teste com mudança de parâmetros descritas, avaliando diferentes cenários e demonstrando a redução essencial na quantidade de automóveis fretados necessários para conseguir levar todos os colaboradores.

---

Este artigo apresenta uma diferente proposta de abordagem, se comparada com a utilizada por este grupo. Entretanto, esta visão distinta demonstra outras nuances e soluções que eram desconhecidas até o momento e que podem ser interessantes para serem aplicadas, isto é, após uma reformulação e revisão congruente com o escopo esperado.

## 2º Pesquisa relevante:

### Artigo Referenciado

**Título:** Aplicación del modelo de enrutamiento de vehículos combinado con algoritmos de optimización para la distribución de insumos relacionados con el servicio asistencial a pacientes hospitalizados y sospechosos de la COVID-19 en Camagüey, Cuba.

**Autores:** Yoan Martínez López, Hilda Oquendo Ferrer, Yailé Caballero Motal, Luis Eduardo Guerra Rodríguez, Raúl Junco Villegas, Isnel Benítez Cortés, Ansel Rodríguez González, Julio Madera Quintana.

### Principais Achados:

O artigo sobre a aplicação do modelo de enrutamento de veículos combinado com algoritmos de otimização para a distribuição de insumos em Camagüey, Cuba, oferece insights valiosos que podem ser aplicados ao projeto de otimização das rotas dos leituristas da Aegea Saneamentos S.A. Um dos achados principais é a utilização de problemas de enrutamento de veículos heterogêneos com janelas de tempo, que assegura que todas as entregas sejam feitas dentro de prazos específicos. Este método pode ser adaptado para garantir que todas as residências sejam atendidas no tempo estipulado para a leitura dos hidrômetros.

A aplicação de algoritmos de otimização, como EDA (Estimation of Distribution Algorithms) e VNS (Variable Neighborhood Search), demonstrou ser eficaz na resolução do problema FSMVRPTW (Problema de Roteamento de Veículos com Múltiplas Janelas de Tempo e Múltiplos Veículos). A utilização desses algoritmos no contexto do projeto da Aegea pode aumentar a eficiência na distribuição das rotas de leitura dos hidrômetros, reduzindo o tempo total e os custos operacionais.

Outro achado significativo é a implementação prática das metaheurísticas, utilizando a biblioteca CVRP em Matlab e uma implementação de código aberto em Excel para o VNS. Esta abordagem prática facilita a adaptação e aplicação desses algoritmos em diferentes contextos, incluindo a otimização das rotas dos leituristas. O uso de ferramentas acessíveis como Excel pode simplificar a integração das soluções otimizadas no fluxo de trabalho diário da Aegea.

O estudo de caso realizado em Camagüey também evidenciou a importância de adaptar as soluções de enrutamento a situações específicas, como as pandemias. Isso reforça a necessidade de uma abordagem flexível e personalizada na otimização das rotas dos leituristas, considerando as particularidades geográficas e operacionais do Rio de Janeiro.

A análise comparativa de 15 modelos diferentes para estudar o comportamento dos algoritmos aplicados ao problema de enrutamento mostrou que a escolha do algoritmo pode influenciar significativamente os resultados. Portanto, para o projeto da Aegea, uma análise detalhada e a comparação de diferentes algoritmos, como nearest neighbors, algoritmos genéticos e Simulated Annealing, podem ajudar a identificar a solução mais eficiente para as rotas de leitura dos hidrômetros.

Por fim, os resultados do estudo de Camagüey destacam a importância de uma organização eficiente do transporte em situações críticas, como a pandemia de COVID-19. Aplicar essas lições ao contexto da Aegea pode não apenas otimizar as rotas de leitura, mas também melhorar a resiliência operacional da empresa em situações de emergência, garantindo a continuidade e a qualidade dos serviços prestados.

---

O estudo sobre a aplicação do modelo de enrutamento de veículos combinado com algoritmos de otimização para a distribuição de insumos em Camagüey, Cuba, oferece lições valiosas que podem ser adaptadas ao projeto de otimização das rotas de leituristas da Aegea Saneamentos S.A. Os principais achados deste estudo ressaltam a importância de utilizar algoritmos de otimização avançados, como EDA e VNS, para resolver problemas complexos de roteamento com múltiplas janelas de tempo e veículos heterogêneos. A aplicação dessas técnicas no contexto da Aegea pode potencialmente aumentar a eficiência e reduzir os custos operacionais das leituras de hidrômetros.

### 3º Pesquisa relevante:

#### Artigo Referenciado

**Título:** An Adaptive Large Neighborhood Search for the Vehicle Routing Problem with Time Windows and Multiple Delivery Men

**Autor:** Enrique Alvarez, Daniela Reyes, Ricardo Vargas.

#### Principais Achados

O estudo de Alvarez et al. (2023) apresenta uma metodologia robusta baseada no algoritmo Adaptive Large Neighborhood Search (ALNS) para resolver o problema de roteirização de veículos com janelas de tempo e múltiplos entregadores. Uma das semelhanças destacadas entre o problema abordado no artigo e o desafio enfrentado pela Aegea Saneamento reside na necessidade de otimizar as rotas para minimizar a distância percorrida, garantindo eficiência na distribuição.

Além disso, o ALNS demonstrou ser escalável, sendo testado em problemas de grande porte, o que é relevante para a Aegea, que lida com centenas de milhares de pontos de leitura de hidrômetros. Essa capacidade de lidar com conjuntos de dados volumosos e complexos é crucial para a eficiência na otimização das rotas.

Outro ponto de convergência é a restrição temporal. Tanto o problema abordado pelo artigo quanto o da Aegea impõem restrições temporais, como janelas de tempo para as entregas no caso do ALNS e considerações de horas de trabalho e dias de leitura no caso da Aegea.

Por fim, o estudo destaca a adaptabilidade do ALNS para atender às necessidades específicas da Aegea, como a consideração de múltiplos meios de transporte, a minimização de cruzamento de ruas e a implementação de sequenciamento cíclico, aspectos fundamentais no contexto das rotas de leitura de hidrômetros.

Esse artigo fornece insights valiosos que podem ser aplicados ao desafio de otimização das rotas de leituristas da Aegea Saneamento. A utilização do algoritmo ALNS como base para o desenvolvimento de um otimizador pode contribuir significativamente para aumentar a eficiência operacional e reduzir os custos da empresa, adaptando-se às necessidades específicas do setor de saneamento.

---

### 4º Pesquisa relevante:

## Artigo Referenciado

**Título:** A Systematic Literature Review of Vehicle Routing Problems with Time Windows

**Autor:** Xiaobo Liu, Yen-Lin Chen, Lip Yee Por, Chin Soon Ku

## Principais Achados

A revisão sistemática realizada por Liu et al. (2023) sobre os problemas de roteirização de veículos com janelas de tempo (VRPTW) oferece insights valiosos que podem ser aplicados ao projeto de otimização das rotas dos leituristas da Aegea Saneamento S.A. Este estudo destaca o uso predominante de algoritmos aproximados, com ênfase nas meta-heurísticas e métodos híbridos, para resolver problemas complexos de roteirização.

Um dos achados principais é a importância dos métodos híbridos, que combinam dois ou mais algoritmos para melhorar a eficiência e a qualidade das soluções. Este aspecto pode ser adaptado para o contexto da Aegea, utilizando uma combinação de técnicas, como algoritmos genéticos e busca por vizinhança variável, para otimizar as rotas de leitura de hidrômetros.

Outra observação significativa é a aplicação de algoritmos em problemas reais, como a coleta de resíduos e a entrega de bens perecíveis, que apresentam desafios semelhantes aos enfrentados pela Aegea, como a necessidade de atender a janelas de tempo específicas e minimizar a distância total percorrida. A implementação desses algoritmos pode ajudar a reduzir os custos operacionais e aumentar a eficiência dos leituristas.

Além disso, o estudo aponta a relevância de ajustar as soluções de roteirização às especificidades locais, como as restrições geográficas e as condições de tráfego. Esta abordagem flexível pode ser crucial para a Aegea, considerando as particularidades das áreas atendidas no Rio de Janeiro, permitindo a criação de rotas mais eficientes e adaptadas às necessidades dos clientes.

A revisão também destaca a importância de realizar simulações com diferentes parâmetros para avaliar a eficácia dos algoritmos em diversos cenários operacionais. Para a Aegea, isso pode envolver testar diferentes quantidades de leituristas e variáveis operacionais para identificar a melhor configuração que maximize a produtividade e minimize os custos.

A revisão sistemática enfatiza a crescente complexidade dos problemas de roteirização à medida que o tamanho dos dados de entrada aumenta. Para a Aegea, isso significa que a aplicação de técnicas avançadas de otimização será essencial para lidar com a grande quantidade de pontos de leitura de hidrômetros, garantindo que as rotas sejam otimizadas de maneira eficaz e sustentável.

Este estudo oferece uma abordagem abrangente e adaptável para resolver problemas de roteirização de veículos com janelas de tempo, apresentando soluções que podem ser moldadas para atender às necessidades específicas da Aegea Saneamento S.A. A utilização de métodos híbridos e a adaptação às condições locais são estratégias-chave que podem ser incorporadas para melhorar a eficiência e reduzir os custos das operações de leitura de hidrômetros.

## Algoritmos adotados para resolver o problema

### Nearest Neighbor com KDTree

#### Descrição:

O *Nearest Neighbor* (ou, em tradução literal, Vizinho Mais Próximo) é uma heurística que busca resolver exemplos do Problema do Caixeiro Viajante através de uma estratégia gulosa: começando em um ponto arbitrário dentre todos os existentes, o código encontra aquele que, dentre todos os disponíveis, está mais próximo; este processo se repete de maneira recursiva, até que todos os pontos sejam visitados. Para melhorar a eficiência na busca dos pontos mais próximos, pode-se utilizar uma estrutura de dados chamada KD-Tree, que permite a realização de consultas espaciais de maneira mais rápida e eficiente.

Esta implementação é simples encontra caminhos relativamente eficientes. Por outro lado, a heurística de *Nearest Neighbor* geralmente encontra soluções sub-ótimas, ou seja, aquelas que, apesar de se aproximarem do resultado ideal, não o atingem. Isto ocorre pois as ações tomadas em cada uma das instâncias não levam em

consideração o impacto destas a longo prazo. Em casos nos quais a quantidade de dados é elevada, esta situação se acentua, sendo que, constantemente, o algoritmo vai encontrar soluções mais distintas do resultado ótimo.

A utilização de KD-Tree ajuda a reduzir o tempo de execução do algoritmo, tornando a busca pelo vizinho mais próximo mais eficiente, especialmente em instâncias maiores. No entanto, mesmo com essa otimização, o Nearest Neighbor ainda mantém as suas limitações intrínsecas, como a dependência da cidade inicial e a incapacidade de capturar a estrutura global do problema. Portanto, enquanto a KD-Tree melhora a performance, ela não altera a natureza gananciosa do algoritmo, que continua propenso a gerar percursos sub-ótimos em comparação com outras técnicas mais avançadas.

## Pseudocódigo:

### 1. Importar bibliotecas necessárias

- Importar pandas como `pd`
- Importar numpy como `np`
- Importar KMeans de `sklearn.cluster`
- Importar KDTree de `scipy.spatial`
- Importar json

### 2. Definir funções auxiliares

- Definir `length(ponto1, ponto2)`
  - Retornar a distância Euclidiana entre `ponto1` e `ponto2`
- Definir `total_length_kdtree(solution, locations)`
  - Inicializar `total` como 0
  - Para cada par de pontos consecutivos em `solution`, adicionar a distância entre eles a `total`
  - Adicionar a distância do último ponto para o primeiro a `total`
  - Retornar `total`
- Definir `nearest_neighbor_kdtree(locations)`
  - Inicializar KDTree com `locations`
  - Inicializar `path` com `[0]`, `visited` com lista de False
  - Marcar o primeiro ponto como visitado
  - Iterar sobre todos os pontos, encontrando o vizinho mais próximo não visitado e adicionando-o a `path`
  - Adicionar 0 ao final de `path` para completar o tour
  - Retornar `path`
- Definir `return_solution(solution, total_distance, coordinates)`
  - Criar dicionário `response` com `funcao_objetivo`, `rota`, `tempo_percorrido`, `coordinates`
  - Retornar `response`
- Definir `convert_numpy_types(obj)`
  - Converter tipos NumPy para tipos nativos do Python para serialização JSON

### 3. Carregar e preparar os dados

- Carregar dados de um arquivo CSV em `data`
- Criar uma cópia do DataFrame `data` em `clustered_data`
- Converter coordenadas de latitude e longitude para metros
- Adicionar colunas `x_metros` e `y_metros` a `clustered_data`

### 4. Clusterizar e encontrar rotas

- Definir `n_clusters` como 30
- Clusterizar `cluestered_data` usando KMeans
- Inicializar `result_json` e `response_list`
- Inicializar coluna `NUM_ROTA` em `cluestered_data` como -1
- Para cada cluster:
  - Filtrar dados do cluster atual
  - Extrair índices, coordenadas e locais
  - Se houver locais:
    - Encontrar o caminho do vizinho mais próximo
    - Calcular o comprimento total do caminho
    - Criar `response` com a solução
    - Atualizar `NUM_ROTA` em `cluestered_data`
    - Adicionar `response` a `response_list`

## 5. Salvar resultados

- Salvar `response_list` em um arquivo JSON
- Salvar `cluestered_data` atualizado em um arquivo CSV

## 6. Imprimir mensagem de finalização

- Imprimir "Processo finalizado."

# Ant Colony

## Descrição:

Já o *Ant Colony Optimization* é uma prática probabilística para resolução de problemas, inspirando-se no processo biológico de encontro de caminhos comuns pelas formigas. O algoritmo percorre por diversas instâncias semi-aleatórias, representando cada qual uma agente independente (uma formiga artificial). Estes manejam os diferentes parâmetros que ainda transitam dentre alguma solução existente, percorrendo os resultados encontrados em busca daquele que apresenta o melhor resultado dada a função objetivo. (Wang et.al)

Este algoritmo foi inicialmente citado em 1992, na tese de doutorado de Marco Dorigo, e seu objetivo era de encontrar o melhor caminho dentro de um grafo a partir do caminho que formigas fazem entre sua casa e a fonte de alimento. Essa perspectiva se expandiu e permitiu a criação de diversos outros algoritmos que seguem o mesmo formato. Esta não garante o resultado ótimo, por se tratar de uma presunção meta-heurística, entretanto atinge um resultado razoavelmente bom em comparação com outros algoritmos. (Bell, McMullen, E.)

# Pseudocódigo

## 1. Importar bibliotecas necessárias

- Importar `pandas` como `pd`
- Importar `numpy` como `np`
- Importar KMeans de `sklearn.cluster`
- Importar `json`
- Importar os módulos e funções necessários da biblioteca `numpy`

## 2. Definir classe e funções auxiliares

- Definir a classe `AntColony` com atributos e métodos:
  - `__init__(self, distances, n_ants, n_best, n_iterations, decay, alpha=1, beta=2)`
  - `run(self)`

- `update_pheromone(self, all_paths, shortest_path)`
- `gen_path_dist(self, path)`
- `gen_all_paths(self)`
- `gen_path(self, start)`
- `pick_move(self, pheromone, dist, visited)`
- Definir `create_matrix(locations)`
  - Criar uma matriz de distâncias a partir de coordenadas de locais

### 3. Carregar e preparar os dados

- Carregar dados de um arquivo CSV em `data`
- Criar uma cópia do DataFrame `data` em `clustering_data`
- Converter coordenadas de latitude e longitude para metros
- Adicionar colunas `x_metros` e `y_metros` a `clustering_data`

### 4. Clusterizar e encontrar rotas

- Definir `n_clusters` como 30
- Clusterizar `clustering_data` usando KMeans
- Inicializar `result_json` e `response_list`
- Inicializar coluna `ROUTE_NUM` em `clustering_data` como -1
- Para cada cluster:
  - Filtrar dados do cluster atual
  - Extrair índices, coordenadas e locais
  - Se houver mais de um local:
    - Criar matriz de distâncias
    - Inicializar `AntColony` com parâmetros apropriados
    - Executar algoritmo de colônia de formigas para encontrar o caminho mais curto
    - Atualizar `ROUTE_NUM` em `clustering_data`
    - Adicionar resposta a `response_list`

### 5. Salvar resultados

- Salvar `response_list` em um arquivo JSON
- Salvar `clustering_data` atualizado em um arquivo CSV

### 6. Imprimir mensagem de finalização

- Imprimir "Processo finalizado."

## OR-Tools

### Descrição:

O ORTools é um *software* de código aberto que foi desenvolvido pela Google, empresa norte-americana e líder no mercado de tecnologia. Seu principal intuito é o de resolver problemas de otimização combinada através da análise de múltiplas soluções e, de maneira conseguinte, da escolha da opção mais apropriada. (“Sobre as ferramentas OR | OR-Tools”, [s.d.])

Sequencialmente, pode-se arguir que a biblioteca desenvolvida utiliza de algoritmos de última geração (além do toque pessoal de inúmeros cientistas que são referência, tanto na literatura, quanto na prática, para questões de otimização) para encontrar a solução ótima (ou sub-ótima, para problemas np-completos) de uma determinada problemática. (“Sobre as ferramentas OR | OR-Tools”, [s.d.])



O papel desempenhado pelos autores deste artigo foi de, em acordo com as entradas esperadas e os parâmetros possíveis, desenvolver um método de utilizar este *software* de maneira organica e respeitando, ademais, as restrições provenientes do escopo da pesquisa.

## Pseudocódigo

### 1. Importar bibliotecas necessárias

- Importar json
- Importar pandas como pd
- Importar numpy como np
- Importar KMeans de sklearn.cluster
- Importar KDTree de scipy.spatial
- Importar math
- Importar pywrapcp e routing\_enums\_pb2 de ortools.constraint\_solver

### 2. Definir funções auxiliares

- Definir `length(point1, point2)`
  - Retornar a distância Euclidiana entre point1 e point2
- Definir `total_length_kdtree(solution, locations)`
  - Inicializar `total` como 0
  - Para cada par de pontos consecutivos em `solution`, adicionar a distância entre eles a `total`
  - Adicionar a distância do último ponto para o primeiro a `total`
  - Retornar `total`
- Definir `nearest_neighbor_kdtree(locations)`
  - Inicializar KDTree com `locations`
  - Inicializar `path` com [0], `visited` com lista de False
  - Marcar o primeiro ponto como visitado
  - Iterar sobre todos os pontos, encontrando o vizinho mais próximo não visitado e adicionando-o a `path`
  - Adicionar 0 ao final de `path` para completar o tour
  - Retornar `path`
- Definir `create_data_model(locations, coordinates)`
  - Criar dicionário `data` com `locations`, `coordinates`, `num_vehicles` e `depot`
  - Retornar `data`
- Definir `compute_euclidean_distance_matrix(locations)`
  - Inicializar `distances` como um dicionário
  - Para cada par de locais, calcular a distância Euclidiana e armazenar em `distances`
  - Retornar `distances`
- Definir `print_solution(manager, routing, solution)`
  - Imprimir o valor da função objetivo da `solution`
  - Inicializar `index` com o índice inicial
  - Inicializar `plan_output` e `route_distance`
  - Enquanto `index` não for o final, imprimir o índice do nó atual e somar a distância ao próximo nó a `route_distance`
  - Imprimir a rota e a distância total
- Definir `return_solution(manager, routing, solution, coordinates)`
  - Inicializar `response` como um dicionário

- Preencher `response` com `objective_value`, `route`, `time_taken` e `coordinates`
- Retornar `response`

### 3. Carregar e preparar os dados

- Carregar dados de um arquivo CSV em `data`
- Criar uma cópia do DataFrame `data` em `clustered_data`
- Converter coordenadas de latitude e longitude para metros
- Adicionar colunas `x_metros` e `y_metros` a `clustered_data`

### 4. Clusterizar e encontrar rotas

- Definir `n_clusters` como 30
- Clusterizar `clustered_data` usando KMeans
- Inicializar `result_json` e `response_list`
- Inicializar coluna `ROUTE_NUM` em `clustered_data` como -1
- Para cada cluster:
  - Filtrar dados do cluster atual
  - Extrair índices, coordenadas e locais
  - Se houver locais:
    - Chamar a função `main(locations, coordinates)` para resolver o problema de roteamento
    - Atualizar `ROUTE_NUM` em `clustered_data`
    - Adicionar resposta a `response_list`

### 5. Salvar resultados

- Salvar `response_list` em um arquivo JSON
- Salvar `clustered_data` atualizado em um arquivo CSV

### 6. Imprimir mensagem de finalização

- Imprimir "Processo finalizado."

## Resultados obtidos

O desenvolvimento dos testes foi um processo essencial para a comparação entre diferentes abordagens de otimização. O grupo conduziu múltiplos testes, cada um utilizando um algoritmo distinto, para avaliar a eficácia dessas abordagens.

### Otimização das Rotas de Leituristas

A análise dos resultados demonstrou que o projeto de otimização das rotas de leituristas apresentou resultados promissores, que, em comparação com as rotas realizadas atualmente, promovem uma otimização do tempo de trajeto e da distância percorrida. Os testes foram realizados tanto com amostras dos dados quanto com a totalidade dos dados disponíveis. A metodologia adotada envolveu a clusterização dos dados e, posteriormente, a aplicação dos algoritmos de otimização em cada cluster individualmente. Os algoritmos avaliados foram *Nearest Neighbor*, *Ant Colony* e *OR-Tools* para o Problema do Caixeiro Viajante (TSP).

### Métricas Avaliadas

As métricas cruciais avaliadas incluíram a distância média percorrida por leiturista e o tempo médio gasto por leiturista. A Tabela 1 apresenta um resumo das métricas obtidas:

Algoritmo	Distância Média	Tempo Médio
-----------	-----------------	-------------

Algoritmo	Distância Média	Tempo Médio
Nearest Neighbor	2,450 KM	~7 horas
TSP com OR-Tools	1,988 KM	~7 horas
Ant Colony	4,485 KM	~7 horas

**Tabela 1:** Comparação das métricas de distância média e tempo médio entre os algoritmos *Nearest Neighbor*, *Ant Colony* e *OR-Tools*.

Os resultados apresentados na Tabela 1 foram obtidos a partir de uma amostra menor, utilizando um total de 30 clusters. A amostra menor utilizada tem 6079 pontos distintos, cada qual com sua latitude e longitude especificadas. Os parâmetros para o *Ant Colony* foram de:  $n\ ants=10$ ,  $n\ best=5$ ,  $n\ iterations=100$ ,  $decay=0.95$

## Análise dos Resultados

A análise dos resultados evidenciou que a utilização da biblioteca do Google *OR-Tools* proporcionou os melhores resultados em termos de otimização de rotas. A menor distância média percorrida, observada com o uso do *OR-Tools*, indica uma eficiência superior em relação ao algoritmo *Nearest Neighbor* e de *Ant Colony*. Ambos os algoritmos apresentaram um tempo médio gasto por leitorista semelhante (~7 horas), o que sugere que a vantagem do *OR-Tools* se reflete principalmente na redução da distância percorrida.

Essa vantagem significativa em comparação com os outros algoritmos disponíveis e utilizados reflete, consideravelmente, na natureza dos dados disponíveis, sendo que a distribuição destes representa um impacto significativo na alteração dos resultados esperados. Para além disso, considera-se que, para problemas np-completos como o de roteamento, não se encontra uma opção computacionalmente viável de se assegurar um resultado ótimo para uma quantidade grande de dados. (GAREY, M.; JOHNSON)

Em resumo, a aplicação do *OR-Tools* no contexto da otimização das rotas de leitoristas demonstrou ser a abordagem mais eficaz entre as avaliadas, principalmente no contexto da aplicação desenvolvida, oferecendo uma solução que pode contribuir significativamente para a redução de custos e aumento da eficiência operacional.

Para além disso, a demonstração dos resultados afirma, empiricamente, que é possível se denotar uma resposta efetiva para um problema complexo como o de roteamento de veículos através de heurísticas e métodos computacionais já conhecidos.

## Conclusão

O estudo apresentado explorou diferentes heurísticas e algoritmos de otimização para resolver o problema de roteamento de leitoristas da Aegea Saneamentos S.A. na região do Rio de Janeiro. Para tal, foram analisados os algoritmos *Nearest Neighbors com KDTree*, *Ant Colony Optimization* e *OR-Tools*, cada um com suas características e vantagens.

Os resultados demonstraram que o algoritmo *OR-Tools*, por ser um solver de otimização combinatória de última geração, apresentou o melhor desempenho em termos de qualidade das soluções, encontrando rotas mais curtas e eficientes.

No entanto, é importante ressaltar que a escolha do algoritmo ideal depende de diversos fatores, como a complexidade do problema, a disponibilidade de recursos computacionais e os requisitos específicos da empresa.

### Recomendações para trabalhos futuros:

- Implementar uma interface amigável para facilitar a utilização da ferramenta de otimização pelos leitoristas e pela equipe de planejamento da Aegea.
- Considerar a integração da ferramenta com sistemas de rastreamento GPS para monitorar o desempenho dos leitoristas em tempo real.
- Analisar o impacto da otimização das rotas na qualidade dos serviços prestados pela Aegea, como a redução do tempo de leitura e a melhora na precisão das medições.
- Investigar outras técnicas de otimização, como algoritmos híbridos que combinam diferentes heurísticas, para buscar soluções ainda mais eficientes.

Este estudo contribui para a otimização das operações da Aegea Saneamentos S.A., reduzindo custos, aumentando a eficiência e aprimorando a qualidade dos serviços prestados aos clientes. A aplicação de técnicas de otimização em outros setores da indústria de saneamento básico pode gerar resultados semelhantes, contribuindo para a sustentabilidade e a eficiência na gestão dos recursos hídricos.

## Limitações do estudo

É importante mencionar que este estudo apresenta uma limitação, como ao não atentamento à ocasionalidades locais que podem acontecer, além da velocidade do leitorista real ser variável e não constante. Isso significa que os resultados podem não ser totalmente precisos em situações reais, onde o tráfego, as condições climáticas e outras variáveis podem afetar o tempo de leitura dos hidrômetros.

**Apesar dessa limitação, o estudo fornece uma base sólida para a otimização das rotas de leitura da Aegea Saneamentos S.A. e contribui para o avanço do conhecimento na área de otimização logística aplicada ao setor de saneamento básico.**

## Referências

AEGEA. Disponível em: <https://www.aegea.com.br/>. Acesso em: 30 abr. 2024.

IBM. What is the k-nearest neighbors algorithm? Disponível em: <https://www.ibm.com/topics/knn>. Acesso em: 30 abr. 2024.

PORTOSIL, E. O Problema do Caixeiro Viajante. Disponível em: <http://www.mat.ufrgs.br/~portosil/caixeiro.html>. Acesso em: 30 abr. 2024.

SCIENCEDIRECT. Simulated Annealing Algorithm. Disponível em: <https://www.sciencedirect.com/topics/engineering/simulated-annealing-algorithm#:~:text=The%20simulated%20annealing%20algorithm%20is,state%20is%20reached%20%5B143%5D>. Acesso em: 30 abr. 2024.

OLIVEIRA, R. S. et al.. Modelagem para o Problema de Roteamento de Veículos Fretados. Trends in Computational and Applied Mathematics, v. 23, n. 4, p. 801–828, out. 2022.

MARTINEZ LOPEZ, Yoan et al . Aplicación de la investigación de operaciones a la distribución de recursos relacionados con la COVID-19. Rev retos, Camagüey , v. 14, n. 2, p. 86-105, dic. 2020 . Disponible en [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S2306-91552020000200086&lng=es&nrm=iso](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2306-91552020000200086&lng=es&nrm=iso). accedido en 16 mayo 2024. Epub 02-Ago-2020.

Alvarez, E., Reyes, D., & Vargas, R. (2023). An Adaptive Large Neighborhood Search for the Vehicle Routing Problem with Time Windows and Multiple Delivery Men. Journal of Advanced Transportation, 2023, 1-15.

Liu, X., Chen, Y.-L., Por, L. Y., & Ku, C. S. (2023). A Systematic Literature Review of Vehicle Routing Problems with Time Windows. Sustainability, 15(15), 12004. Disponível em: <https://doi.org/10.3390/su151512004>

GAREY, M.; JOHNSON, D. Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco: W.H. Freeman and Company, 1979. ISBN 0716710455.

X. Wang, T. -M. Choi, H. Liu and X. Yue, "Novel Ant Colony Optimization Methods for Simplifying Solution Construction in Vehicle Routing Problems," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 11, pp. 3132-3141, Nov. 2016, doi: 10.1109/TITS.2016.2542264.

Bell, John E., and Patrick R. McMullen. "Ant colony optimization techniques for the vehicle routing problem." Advanced engineering informatics 18.1 (2004): 41-48.

Sobre as ferramentas OR | OR-Tools. Disponível em: <https://developers.google.com/optimization/introduction?hl=pt-br>. Acesso em: 12 jun. 2024.