



# APLICAÇÃO DE ROTEIRIZAÇÃO DE VEÍCULOS COM MÚLTIPLOS CENTROS DE DISTRIBUIÇÃO EM UMA EMPRESA DE CORTES DE FRANGO

JEFERSON ALMEIDA FERREIRA  
IFES – Campus Cariacica. jefalmeidaferreira@gmail.com.

Prof. GUILHERME GUILHERMINO NETO  
IFES – Campus Cariacica

## 1 PROBLEMA DE PESQUISA

A complexidade da roteirização nas operações logísticas exige abordagens inovadoras e eficientes para a resolução dos problemas de otimização com o intuito de garantir a entrega de produtos e serviços de forma ágil e econômica (Cunha, 2016). No contexto da empresa de cortes de frango estudada, localizada em Marechal Floriano/ES que abate 47 mil aves por dia - equivalente a 143 t/dia - e que possui produção própria dos frangos abatidos e das rações consumidas, essa necessidade é crucial para as operações, em função do tamanho e da dinâmica das suas operações logísticas que é 100% própria executada por cinco caminhões, sendo quatro *trucks* com capacidade de 16 toneladas e um *bi-truck* com capacidade de 22 toneladas. A estrutura da empresa engloba duas fábricas de ração e 146 granjas localizadas ao redor das fábricas, onde cada uma das granjas têm a sua demanda específica variando semanalmente em relação à quantidade e tipo de ração, devido ao ciclo de vida dos frangos. Isso faz com que o desafio logístico se torne ainda mais complexo e crucial.

A competitividade no mercado atual é fortemente influenciada pelas decisões logísticas acertadas. A capacidade de uma empresa de otimizar suas rotas de transporte, reduzir custos operacionais e atender às demandas de forma eficiente pode ser determinante para sua posição no mercado. Para empresas que possuem em suas atividades a dinâmica do processo de criação de animais, cujo sucesso é evidenciado pelo seu significativo controle diário ou semanal e pela autossuficiência em termos de produção e logística, as boas decisões logísticas são vitais para manter sua competitividade e sustentabilidade no longo prazo.

O presente trabalho propõe solucionar a roteirização de entrega de ração da empresa em questão, com o intuito de otimizar as operações logísticas de entrega de ração semanal às 146 granjas. O objetivo principal é reduzir os custos com transporte, garantindo a eficiência na distribuição da ração, considerando as demandas sazonais em função do crescimento e abate dos frangos e variáveis das granjas, além de maximizar a capacidade de produção das fábricas de ração, que já operam na



capacidade máxima. O objetivo específico desse trabalho é desenvolver um modelo de roteirização de caminho mínimo que leve em consideração as demandas dinâmicas das granjas e a capacidade de produção das fábricas de ração.

Nesse sentido, essa pesquisa busca responder a seguinte questão: Como otimizar as rotas entre as fábricas de ração de uma empresa de ração de aves e as cento e quarenta e seis granjas ao seu redor, minimizando os custos de transporte em um ambiente logístico dinâmico, caracterizado por demandas variáveis e sazonais?

A relevância dessa pesquisa se destaca no cenário atual, onde a eficiência na gestão de rotas não apenas impacta diretamente os custos operacionais, mas também influencia a satisfação do cliente e a competitividade no mercado. No caso da empresa de abate de ave em questão, que possui um faturamento anual entre 200 e 300 milhões de reais e mais de 300 funcionários (Econodata, 2024), uma gestão de transporte eficiente é fundamental para manter sua posição de destaque no setor, garantindo a qualidade e a regularidade no abastecimento das granjas, o que impacta diretamente na produção e na rentabilidade do negócio. Este trabalho visa contribuir significativamente para a melhoria dos processos logísticos da empresa, oferecendo uma perspectiva inovadora e eficaz na busca por soluções ótimas na roteirização de sua frota de caminhões.

## 2 PROCESSOS METODOLÓGICOS/MATERIAIS E MÉTODOS

A metodologia adotada neste trabalho foi selecionada com base na escolha dos métodos mais adequados para abordar o problema de roteirização de caminhões com múltiplos centros de distribuição. Esta escolha foi fundamentada na necessidade de utilizar abordagens eficazes para solucionar o problema logístico em questão. A metodologia é composta por várias etapas interrelacionadas, conforme descrito a seguir:

- I. **Entendimento do Problema:** Inicialmente, foi realizado um levantamento detalhado das operações logísticas da empresa, incluindo as características das 2 fábricas de ração, as demandas das 146 granjas, as capacidades de produção e a capacidade dos caminhões, entre outros aspectos relevantes. Esse processo permitiu um entendimento completo do problema em questão e a identificação das principais variáveis e restrições envolvidas.
- II. **Coleta e Análise de Dados:** Em seguida, foram coletados os dados necessários para alimentar o modelo de roteirização. Esses dados foram fornecidos pela empresa e enviados por e-mail em arquivos .csv. Incluíram informações detalhadas sobre as localizações das fábricas de ração e das granjas, as demandas de ração de cada granja, as capacidades de produção



das fábricas, as características dos caminhões, entre outros aspectos relevantes.

- III. **Desenvolvimento do Algoritmo Computacional:** Com base no entendimento detalhado do problema e na análise dos dados coletados, foi desenvolvido um algoritmo computacional para representar adequadamente o problema de roteirização. Este algoritmo incluiu variáveis de decisão, funções objetivo e restrições que capturaram todas as nuances do problema, levando em consideração as características específicas das operações logísticas da empresa.
- IV. **Implementação Computacional:** O modelo matemático foi implementado em software de otimização adequado, permitindo a resolução eficiente do problema de roteirização. Foi utilizada a linguagem de programação Python para desenvolver o sistema computacional necessário para realizar as simulações e análises.
  - a. Para a implementação do algoritmo, foi utilizado o compilador Python na versão 3.x. A escolha do Python se deu por sua robustez, facilidade de uso e ampla disponibilidade de bibliotecas específicas para otimização e algoritmos genéticos, como a biblioteca DEAP.
  - b. Recursos Computacionais: Para garantir a eficiência e a rapidez na execução do algoritmo, foram utilizados os seguintes recursos computacionais:
    - i. Processamento Paralelo e Threads: A execução do algoritmo foi otimizada utilizando processamento paralelo e threads, o que permitiu a divisão das tarefas em múltiplos núcleos do processador, acelerando o tempo de processamento das simulações.
    - ii. Placa de Vídeo (GPU): Em casos onde o processamento paralelo exigia maior desempenho, a GPU foi utilizada para realizar cálculos intensivos, aproveitando a capacidade de processamento massivamente paralelo oferecida pelas placas de vídeo modernas.
    - iii. Ambiente de Desenvolvimento: O desenvolvimento e a execução do algoritmo foram realizados em um ambiente de desenvolvimento configurado com um processador de alto desempenho (como Intel Core i7 ou superior), 16GB de RAM e uma placa de vídeo NVIDIA GeForce GTX 1080 ou equivalente.



- V. **Validação e Testes:** O modelo implementado foi validado e testado utilizando dados reais. Foram realizados testes para avaliar a robustez e eficácia do modelo na resolução do problema de roteirização, comparando os resultados obtidos com soluções conhecidas ou ótimas, quando disponíveis.

### 3 RESULTADOS

Para resolver o problema de roteirização da empresa considerando a complexidade do problema e as limitações de recursos computacionais, optou-se por uma abordagem heurística em vez de uma otimização exata. Uma heurística comumente usada para esse tipo de problema é o Algoritmo de Clarke-Wright (1964).

O Algoritmo das economias de Clarke-Wright é uma técnica clássica utilizada em problemas de roteirização para encontrar soluções eficientes e econômicas. Ele foi proposto por Clarke e Wright em 1964 e é amplamente aplicado em contextos logísticos, como a distribuição de produtos, entregas de mercadorias e coleta de resíduos. O algoritmo busca identificar economias de escala ao combinar rotas individuais de forma inteligente, reduzindo a distância total percorrida pelos veículos e, conseqüentemente, os custos operacionais. Isso é alcançado ao considerar as distâncias entre os pontos de entrega e ao avaliar as economias potenciais ao combinar diferentes rotas. O algoritmo é uma ferramenta valiosa para empresas que buscam otimizar suas operações logísticas, melhorar a eficiência dos recursos e oferecer um serviço de entrega mais rápido e econômico para seus clientes.

O modelo para empresa foi elaborado seguindo o seguinte do pseudocódigo abaixo passos baseados no algoritmo das economias de Clarke-Wright (Oliveira et al, 2015):

**Entrada:** Conjunto de vértices  $V$  e conjunto de arestas  $E$ , capacidade máxima  $Q$  dos veículos

**Saída:** Conjuntos de rotas

```
1 routes ← create_initial_routes(V);
2 savings_list = compute_savings_list(V);
3 foreach  $i, j \in$  savings_list do
4     if feasible_merge( $i, j, Q$ ) then
5         routes.add(merge_route( $i, j$ ));
6     end
7 end
8 return routes
```



- a) **Inicialização:** Cada caminhão começa na fábrica e visita uma única granja, depois retorna à fábrica (linha 1).
- b) **Cálculo das Economias:** Calcule a economia de custo ao combinar duas rotas em uma, economizando a distância que seria percorrida ao retornar à fábrica (Linha 2).
- c) **Combinação de Rotas:** Combine as rotas que oferecem a maior economia de custo, respeitando as restrições de capacidade dos caminhões (linhas 4 a 6).
- d) **Iteração:** Continue combinando rotas até que não seja possível obter mais economias significativas (linhas 3 a 7).

O modelo para solucionar o problema foi desenvolvido em python versão 3.x (Apêndice I). O objetivo deste algoritmo é otimizar as rotas de entrega de ração da empresa, garantindo uma distribuição eficiente e econômica da ração para as 146 granjas ao redor das 2 fábricas de ração. A otimização das rotas visa minimizar a distância total percorrida pelos caminhões, levando em consideração a demanda variável de cada granja e a capacidade dos caminhões.

### 3.1 Explicação do algoritmo

- a) Leitura dos dados

Primeiramente, o algoritmo faz a leitura das bases de dados da empresa:

`Distances.csv` – A matriz de distâncias entre as granjas é lida de um arquivo CSV, representando as distâncias de cada granja para todas as outras granjas (apêndice I).

`Factory_distances.csv` – A matriz de distâncias das fábricas para as granjas também é lida de um arquivo CSV, representando as distâncias das fábricas de ração para cada granja (apêndice I).

`Necessidade.csv` – Por fim, a demanda de ração de cada granja é lida de um arquivo CSV (apêndice I).

- b) Inicialização das rotas

Após a leitura, é feita a inicialização das rotas onde cada caminhão começa com uma rota individual para cada granja. Essa manobra garante o caminho mais longo para os caminhões. As rotas iniciais dos caminhões são calculadas considerando a demanda de cada granja e a capacidade dos caminhões.



### c) Cálculo das Economias

É calculada a função de economia de Clarke-Wright para todas as combinações possíveis de granjas. A função de economia representa a economia de distância ao combinar duas granjas em uma única rota.

### d) Cálculo da demanda total

A função `route_demand(route)` calcula a demanda total de uma rota somando as demandas das granjas visitadas.

### e) Combinação das rotas

Com base nas economias de Clarke-Wright calculadas, as rotas dos caminhões são combinadas de forma a maximizar a economia de distância. As combinações são feitas respeitando a capacidade dos caminhões e a demanda das granjas.

### f) Impressão dos resultados

O código deve exibir:

- As rotas que cada caminhão deve seguir, partindo da fábrica designada.
- A quantidade total de ração transportada por cada caminhão.
- A distância percorrida por cada caminhão
- A distância total percorrida.

Após a execução do algoritmo utilizando os dados das tabelas fornecidas pela empresa, foram encontrados os seguintes resultados:

#### Caminhão 1

- parte da Fábrica 0 e visita as granjas: 15, 10, 25, 5, 35, 20, 30, 0, 40, 45, 135, 115, 90, 105, 75, 55, 70, 80, 140, 125, 95, 145, 100, 110, 120, 130, 60, 50, 85, 65
- Total transportado pelo Caminhão 1: 15.29 toneladas
- Distância percorrida pelo Caminhão 1: 798.87 km



### Caminhão 2

- parte da Fábrica 1 e visita as granjas: 6, 21, 16, 11, 26, 36, 1, 31, 41, 46, 116, 56, 136, 76, 111, 66, 121, 61, 96, 91, 86, 51, 71, 101, 131, 141, 81, 126, 106
- Total transportado pelo Caminhão 2: 15.03 toneladas
- Distância percorrida pelo Caminhão 2: 856.68 km

### Caminhão 3

- parte da Fábrica 0 e visita as granjas: 12, 27, 17, 7, 22, 2, 37, 32, 42, 47, 132, 92, 57, 102, 52, 87, 97, 137, 112, 67, 127, 107, 77, 142, 72, 117, 82, 122, 62
- Total transportado pelo Caminhão 3: 14.84 toneladas
- Distância percorrida pelo Caminhão 3: 755.94 km

### Caminhão 4

- parte da Fábrica 1 e visita as granjas: 8, 13, 18, 23, 3, 28, 33, 38, 43, 58, 138, 63, 88, 78, 98, 143, 83, 123, 53, 133, 48, 113, 93, 68, 108, 103, 128, 73, 118

Total transportado pelo Caminhão 4: 14.29 toneladas

Distância percorrida pelo Caminhão 4: 733.35 km

### Caminhão 5

- parte da Fábrica 0 e visita as granjas: 14, 9, 4, 19, 119, 29, 34, 24, 39, 44, 134, 139, 69, 54, 59, 74, 79, 109, 104, 94, 84, 99, 144, 89, 114, 129, 49, 124, 64
- Total transportado pelo Caminhão 5: 14.25 toneladas
- Distância percorrida pelo Caminhão 5: 819.72 km

É possível verificar que todas as granjas foram atendidas com os 5 caminhões disponíveis carregados abaixo da capacidade máxima.



## 4 CONCLUSÃO

Este algoritmo de roteirização procura proporcionar uma solução eficiente para a distribuição de ração da empresa de abate de frango estudada, permitindo uma melhor utilização dos recursos e buscando reduzir os custos operacionais. A otimização das rotas contribui para uma logística mais eficiente e sustentável, almejando o abastecimento adequado das granjas e o sucesso das operações da empresa.

A solução oferece um ponto de partida robusto para a otimização logística na da empresa, trazendo uma proposta que possa ajudar na redução dos custos e no aumento da eficiência das operações. No entanto, é possível implementar diversas outras melhorias para aumentar a eficiência do algoritmo, como por exemplo, a incorporação de técnicas de aprendizado de máquina e inteligência artificial para aprimorar a seleção e combinação das rotas, levando em consideração uma variedade maior de variáveis, como condições de tráfego em tempo real, horários de entrega preferenciais e restrições específicas de cada veículo. Além disso, a integração de algoritmos de otimização multiobjetivo poderia permitir a consideração simultânea de diferentes objetivos, como minimização da distância percorrida, maximização da utilização dos veículos e equilíbrio de carga entre as rotas. Essas melhorias têm o potencial de tornar o Algoritmo das economias de Clarke-Wright ainda mais poderoso e versátil, proporcionando soluções logísticas economicamente mais eficientes e adaptáveis às necessidades em constante mudança das empresas modernas.



## APÊNDICE I – Dados de Entrada

### Matriz das distâncias

0	25.32	10.94	18.46	13.21	10.71	37.37	46.65	16.57	39.48	26.81	27.04	26.17
25.32	0	17.2	32.33	9.68	22.37	25.87	43.5	18.11	40.3	40.03	7.87	17.86
10.94	17.2	0	38.15	22.45	13.95	37.01	44.59	46.07	21.66	33.87	21.73	39.43
18.46	32.33	38.15	0	38.79	39.01	5.04	32.57	19.86	17.17	24.98	31.32	49.48
13.21	9.68	22.45	38.79	0	7.35	43.59	26.21	5.79	12.55	12.49	29.37	22.13
10.71	22.37	13.95	39.01	7.35	0	19.03	44.28	31.67	34.72	43.83	8.67	24.28
37.37	25.87	37.01	5.04	43.59	19.03	0	19.97	15.92	22.53	24.9	35.92	44.31
46.65	43.5	44.59	32.57	26.21	44.28	19.97	0	35.94	39.11	34.58	8.61	35.28
16.57	18.11	46.07	19.86	5.79	31.67	15.92	35.94	0	21.84	47.58	25.08	41.5
39.48	40.3	21.66	17.17	12.55	34.72	22.53	39.11	21.84	0	28.18	10.12	35.6
26.81	40.03	33.87	24.98	12.49	43.83	24.9	34.58	47.58	28.18	0	36.22	39.55
27.04	7.87	21.73	31.32	29.37	8.67	35.92	8.61	25.08	10.12	36.22	0	49.24
26.17	17.86	39.43	49.48	22.13	24.28	44.31	35.28	41.5	35.6	39.55	49.24	0

### Distâncias das granjas para as fábricas

24.0	24.0	15.0	15.0	15.0	15.0	15.0	15.0	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
19.0	19.0	6.0	6.0	6.0	6.0	6.0	6.0	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5

### Demanda das granjas

0.19	0.91	0.06	0.51	0.59	0.7	0.97	0.61	0.28	0.74	0.24	0.19	0.13	0.08	0.82	0.43	0.37	0.52	0.22	0.4	0.51	0.27
------	------	------	------	------	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	-----	------	------

## APÊNDICE II – Código em Python

```
import pandas as pd
from itertools import combinations

# Leitura da matriz de distâncias do arquivo CSV
distance_matrix = pd.read_csv('distances.csv', header=None).values

# Leitura da matriz de distâncias das fábricas para as granjas do arquivo CSV
factory_distances = pd.read_csv('factory_distances.csv', header=None).values

# Leitura das demandas das granjas do arquivo CSV
demandas = pd.read_csv('necessidade.csv', header=None).values.flatten()

# Definição dos parâmetros
NUM_TRUCKS = 5
NUM_GRANJAS = distance_matrix.shape[0]
NUM_FABRICAS = factory_distances.shape[0]

# Capacidades dos caminhões (em toneladas)
capacities = [16, 16, 16, 16, 22]

# Inicialização das rotas: cada caminhão começa com uma rota individual para uma granja
routes = {i: [[j] for j in range(NUM_GRANJAS) if j % NUM_TRUCKS == i] for i in range(NUM_TRUCKS)}

# Calculando a função de economia de Clarke-Wright
def savings():
    s = {}
    for (i, j) in combinations(range(NUM_GRANJAS), 2):
        for k in range(NUM_FABRICAS):
            save = factory_distances[k % NUM_FABRICAS][i] + factory_distances[k % NUM_FABRICAS][j] - distance_matrix[i][j]
            s[(i, j)] = save
            s[(j, i)] = save
    return s

savings_values = savings()
sorted_savings = sorted(savings_values.items(), key=lambda item: item[1], reverse=True)

# Função para calcular a demanda total de uma rota
def route_demand(route):
    return sum(demandas[gran] for gran in route)

# Função para calcular a distância de uma rota
```



```
def route_distance(route):
    dist = factory_distances[route[0] % NUM_FABRICAS][route[0]]
    for i in range(len(route) - 1):
        dist += distance_matrix[route[i]][route[i + 1]]
    dist += factory_distances[route[-1] % NUM_FABRICAS][route[-1]]
    return dist

# Combinação das rotas baseada nas economias calculadas
for (i, j), save in sorted_savings:
    for k in range(NUM_TRUCKS):
        route_i, route_j = None, None
        for route in routes[k]:
            if i in route:
                route_i = route
            if j in route:
                route_j = route

        if route_i and route_j and route_i != route_j:
            combined_demand = route_demand(route_i) + route_demand(route_j)
            if combined_demand <= capacities[k]:
                routes[k].remove(route_i)
                routes[k].remove(route_j)
                routes[k].append(route_i + route_j)
                break

# Imprimindo as rotas encontradas, a quantidade total e a distância de cada caminhão
total_dist = 0
for k in range(NUM_TRUCKS):
    total_demand = sum(route_demand(route) for route in routes[k])
    dist_k = sum(route_distance(route) for route in routes[k])
    total_dist += dist_k
    print(f' Caminhão {k + 1} parte da Fábrica {k % NUM_FABRICAS} e visita as
    granjas:')
    for route in routes[k]:
        print(f' Rota: {route}')
    print(f' Total transportado pelo Caminhão {k + 1}: {total_demand} toneladas')
    print(f' Distância percorrida pelo Caminhão {k + 1}: {dist_k} km')
    print(f' Capacidade do Caminhão {k + 1}: {capacities[k]} toneladas\n')

# Imprimindo a distância total percorrida
print(f' Distância total percorrida por todos os caminhões: {total_dist} km')
```



## 5 REFERÊNCIAS BIBLIOGRÁFICAS

CLARKE, G.; WRIGHT, J. W. **Scheduling of vehicles from a central depot to a number of delivery points**. Operations Research, Institute for Operations Research and the Management Sciences (INFORMS), v. 12, n. 4, p. 568-581, ago. 1964.

CUNHA, C.B. **Aspectos Práticos da Aplicação de Modelos de Roteirização de Veículos a Problemas Reais**. Revista Transportes. Disponível em <https://revistatransportes.org.br/anpet/article/view/188/170>. Acesso em: 23/05/24

ECONODATA. Disponível em <https://www.econodata.com.br/consulta-empresa/>. Acesso em: 15/05/24

OLIVEIRA, R. A. C.; DELGADO, K. V.; MOREIRA, D. A. M. **Sistema para roteamento de veículos capacitados aplicando Métodos de Monte Carlo**. iSys – Revista Brasileira de Sistemas de Informação, Rio de Janeiro, vol. 8, No. 3, p. 42-63, 2015

Jeferson Almeida Ferreira

**APLICAÇÃO DE ROTEIRIZAÇÃO DE VEÍCULOS COM MÚLTIPLOS  
CENTROS DE DISTRIBUIÇÃO EM UMA EMPRESA DE CORTES DE  
FRANGO**

Trabalho Final de Curso apresentado à  
Coordenadoria do Curso de Engenharia de  
Produção do Instituto Federal do Espírito Santo  
– *campus* Cariacica como requisito parcial para  
obtenção do título de Especialista em  
Engenharia de Produção com Ênfase em  
Ciência de Dados

Aprovado em 24 de junho de 2024

**COMISSÃO EXAMINADORA**

Prof. Guilherme Guilhermino Neto, D.Sc.

Ifes – Instituto Federal do Espírito Santo

Orientador

Prof. Tiago José Menezes Gonçalves, D.Sc.

Ifes – Instituto Federal do Espírito Santo

Membro da banca avaliadora

Prof. Vitor dos Santos Amorim, M.Sc.

Ifes – Instituto Federal do Espírito Santo

Membro da banca avaliadora



---

---

**FOLHA DE APROVAÇÃO-TCC Nº 7/2024 - CAR-CCEP (11.02.19.01.08.03.10)**

**(Nº do Protocolo: NÃO PROTOCOLADO)**

**(Assinado digitalmente em 27/06/2024 16:59 )**

**GUILHERME GUILHERMINO NETO**

PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO

CAR-CCEP (11.02.19.01.08.03.10)

Matrícula: 2151589

**(Assinado digitalmente em 27/06/2024 21:25 )**

**TIAGO JOSE MENEZES GONCALVES**

PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO

CAR-CCEP (11.02.19.01.08.03.10)

Matrícula: 2073974

**(Assinado digitalmente em 01/07/2024 23:14 )**

**VITOR DOS SANTOS AMORIM**

PROFESSOR DO ENSINO BASICO TECNICO E TECNOLOGICO-

SUBSTITUTO

CAR-CCEP (11.02.19.01.08.03.10)

Matrícula: 3322581

Visualize o documento original em <https://sipac.ifes.edu.br/documentos/> informando seu número: 7, ano: 2024, tipo:  
**FOLHA DE APROVAÇÃO-TCC**, data de emissão: 27/06/2024 e o código de verificação: 8147a3823c