UnB

Rotas de veículos - Logística de entregas: Uma Abordagem com Algoritmos

Alunos:

- **Bruno Lirio;**
- **Patrícia Aguiar;**
- **Katia Knychala; e**
- **Jairo Rodrigues.**

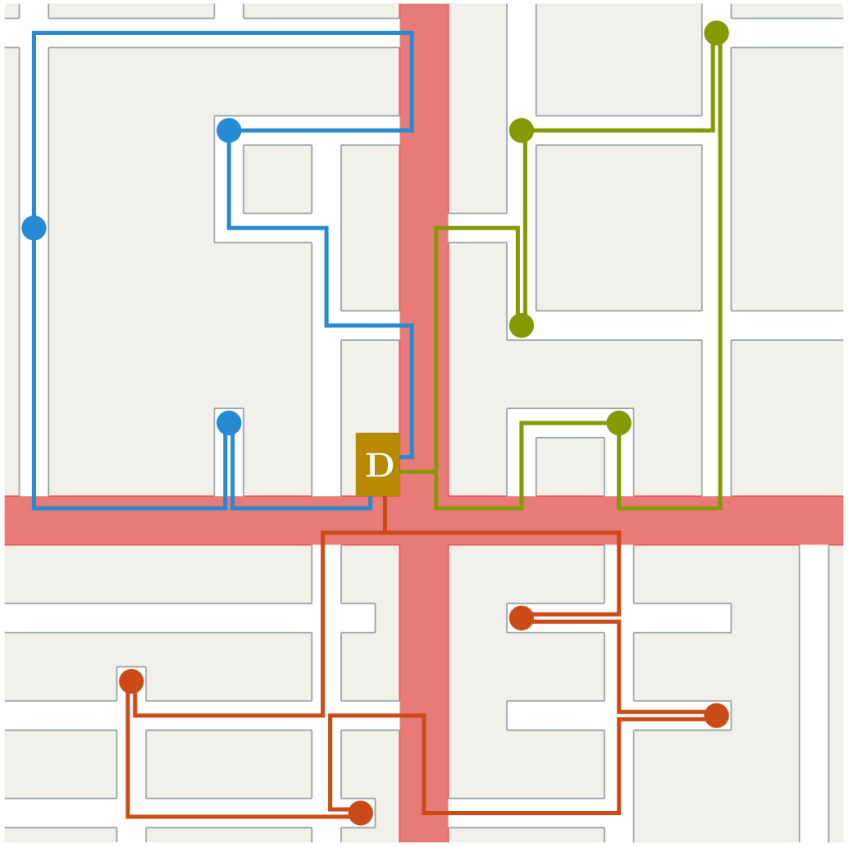
Brasília - DF, 30/08/2025

1. Introdução

Tema: Problema de Roteirização de Veículos (VRP) em Logística de Entregas

- Extensão do Problema do Caixeiro Viajante (TSP), classificado como NP-difícil.
- Exige técnicas de otimização e heurísticas para soluções práticas.
- **Cenário**: Entrega de malotes do depósito do Banco do Brasil (Sede I BB) às agências na Asa Norte e Asa Sul, Brasília.
- **Visualização**: Mapa estilizado do Plano Piloto com rotas otimizadas entre depósito e agências.

Impacto: “Como otimizar a entrega de malotes entre a Sede I BB e as agências, minimizando tempo, custo e emissões?”



2. Motivação e Relevância

Importância Logística:

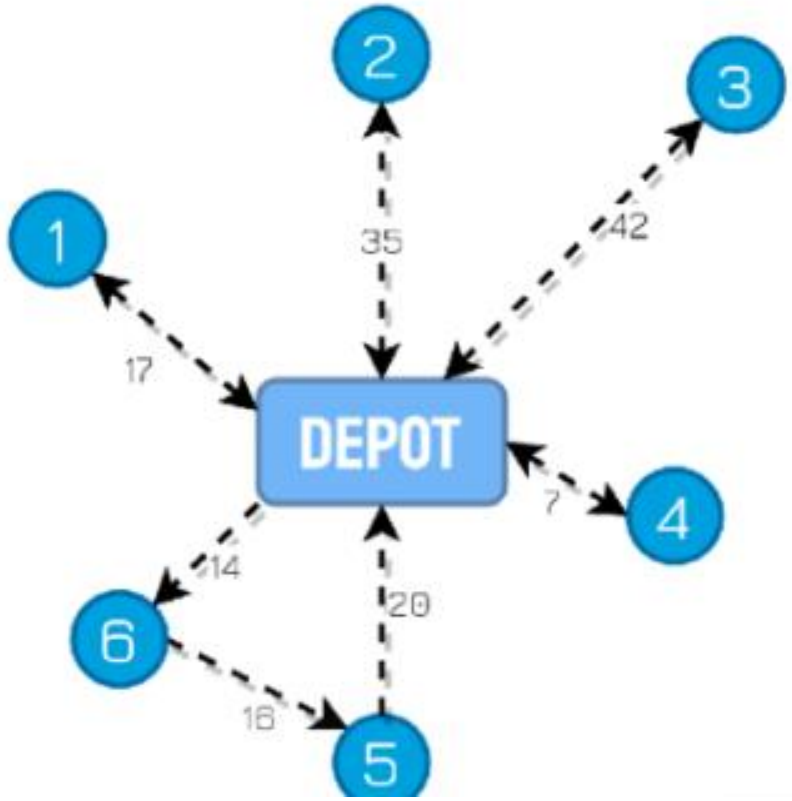
- Empresas como Amazon, Correios e Banco do Brasil lidam com múltiplos pontos de entrega.
- Cada quadra do Plano Piloto representa clientes, agências ou caixas eletrônicos.

Benefícios da Otimização:

- Redução de custos operacionais e tempo de entrega.
- Menor emissão de CO_2 (rotas mais curtas → sustentabilidade).
- Melhor utilização da frota (menos veículos ociosos).

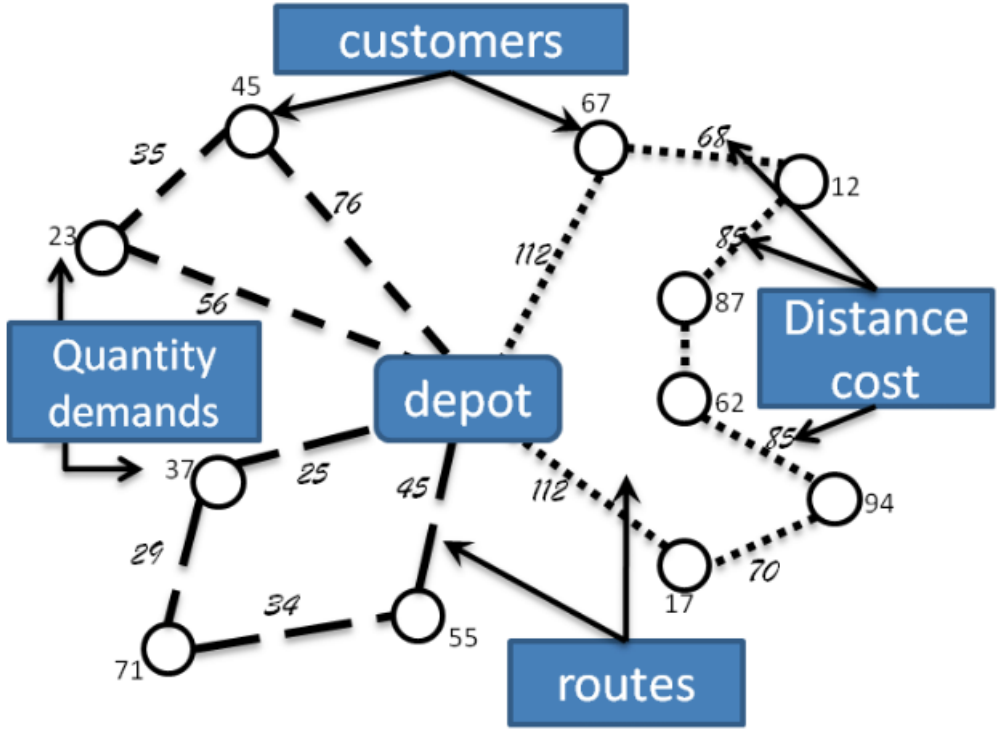
Desafio Computacional:

- Restrições reais incluem capacidade dos veículos, janelas de atendimento e trânsito dinâmico.



3. Objetivos da Pesquisa

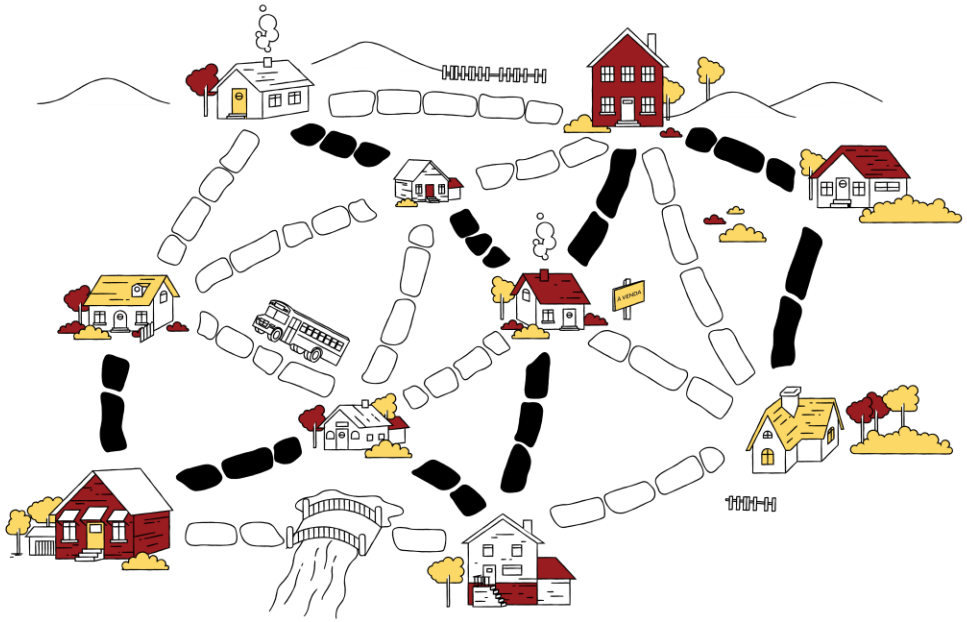
- Desenvolver uma solução em Python para planejar rotas de veículos entre a Sede I BB e agências na Asa Norte e Asa Sul.
- Avaliar a eficiência de heurísticas e algoritmos aproximados.
- Aplicar conceitos de Algoritmos e Estruturas de Dados (AED):
  - Grafos: Representação da malha viária.
  - Complexidade: Análise de problemas NP-difíceis.
  - Algoritmos Aproximados: Uso de metaheurísticas para otimização.



4. Definição do Problema

Entrada:

- Um conjunto de **n agencia/pontos** (quadras 100, 200, 300... da Asa Norte e Asa Sul).
- Um **depósito** (ponto inicial – Ex.: Sede do BB no Setor Bancário Sul para buscar os malotes).
- Frota de **k veículos** com capacidade limitada.



Saída:

- Rotas otimizadas que atendam todas as agências com tempo estimativo

Restrições comuns:

- Capacidade máxima por veículo (ex.: 3 malotes).

Complexidade: heurísticas (análise de complexidade).

5. Modelagem do Problema

Estrutura de Dados:

- Representar a malha viária como **grafo G(V,E)**, onde:
  - **Vértices (V)**: quadras (clientes + depósito).
  - **Arestas (E)**: distâncias entre quadras (usando coordenadas aproximadas de Brasília).
- **Pesos nas arestas**: tempo/distância (pode usar valores simulados).

Exemplo visual: grafo sobreposto ao mapa do Plano Piloto.

Abstração matemática:

- Minimizar ∑ custo(rota) sujeito a capacidade e cobertura de todos os nós.

6. Modelos/Algoritmos utilizados em Python

Objetivo: Resolver o Problema de Roteirização de Veículos (VRP) para entregas no Plano Piloto (Asa Norte e Asa Sul).

Algoritmos aplicados:

1. *Savings* (Clarke-Wright) – para construção inicial das rotas.
2. *2-opt* – melhoria local de rotas para reduzir custos.
3. *Simulated Annealing* – busca global para evitar ótimos locais.

Complexidade: heurísticas foram utilizadas devido à natureza NP-difícil do problema. Ex.: *Savings* é $O(n^2)$, *2-opt* é $O(n^2)$ por rota.

Estruturas de Dados: listas, dicionários e matrizes para armazenar pontos, demandas e custos.

Pseudoalgoritmo simplificado:

1. Ler coordenadas e demandas (depósito + agências).
2. Construir matriz de distâncias (Google Maps API).
3. Criar rotas iniciais (depósito -> agência -> depósito).
4. Mesclar rotas respeitando capacidade.
5. Otimizar cada rota com 2-opt.
6. Refinar com Simulated Annealing (trocas entre rotas).
7. Calcular: tempo, combustível e emissões.
8. Exibir rotas finais e mapa.

```python
# marcadores das agências
for a in agencias:
    folium.Marker(
        [a["lat"], a["lng"]],
        popup=f"{a['nome']} | demanda={a['demanda']}",
        icon=folium.Icon(color="blue", icon="building")
    ).add_to(mc)

route_colors = ["blue", "green", "purple", "orange", "cadetblue", "darkred"]

# Para cada rota, solicitar polyline entre pares consecutivos e desenhar
for idx, r in enumerate(routes):
    color = route_colors[idx % len(route_colors)]
    for u, v in zip(r[:-1], r[1:]):
        src = coords[ids.index(u)]
        dst = coords[ids.index(v)]
        pts = directions_polyline(src, dst)
        if not pts:
            # fallback: desenhar linha reta
            pts = [src, dst]
        folium.PolyLine(pts, weight=5, opacity=0.7, color=color).add_to(m)

# Tabela com rota e custo estimado por rota
rows = []
for r in routes:
    rows.append({
        "rota": " -> ".join(popup_txt(n) for n in r),
        "custo_estimado_min": round(route_cost(r, duration_matrix)/60, 1)
    })

df_rotas = pd.DataFrame(rows)

# Exibir
try:
    display(df_rotas)
except Exception:
    pass

# Em Databricks, para exibir o mapa interativo:
try:
    html = m._repr_html_()
    displayHTML(html)
except Exception:
    # como fallback, salvar em arquivo
    m.save("vrp_brasilia.html")
    print("Mapa salvo: vrp_brasilia.html")
```

7. Descrição dos Dados

Fonte:

- Coordenadas obtidas via Google Maps API.
- Depósito: Sede I BB (SBS).
- 8 agências selecionadas na Asa Norte e Asa Sul.

Atributos principais:

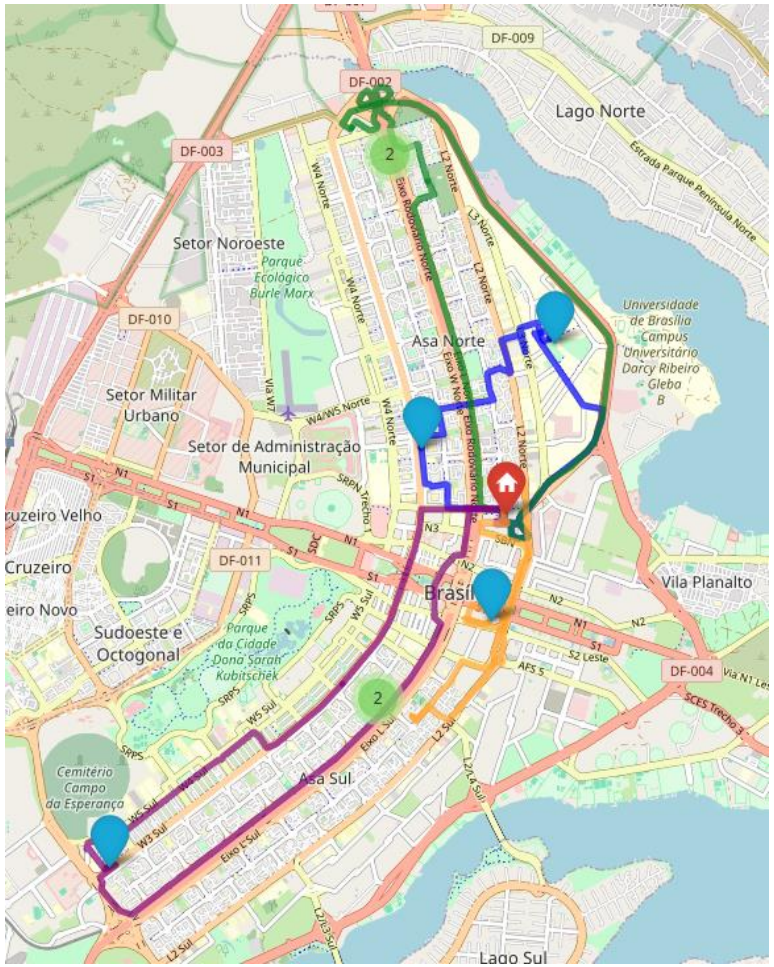
- ID, Nome, Latitude, Longitude.
- Demanda de malotes (1 a 2 por agência).
- Capacidade máxima do veículo: 3 malotes.

Dados calculados:

- Tempo de viagem (minutos).
- Distância (km).
- Custos estimados de combustível (R$ 6,19/L, 12 km/L).
- Emissões estimadas (0,2 kg CO_2/km).

Formato:

- Estruturas em listas e DataFrames (pandas).



7. Descrição dos Dados

```
Consultando Google Distance Matrix...
Rotas finais (IDs):
[0, 1, 7, 0]
[0, 3, 2, 0]
[0, 8, 6, 0]
[0, 4, 5, 0]
Custo total (segundos): 6727
```

Table

	rota	1.2 tempo_estimado_min	1.2 dist_km	1.2 combustivel_litros	1.2 custo_combustivel	1.2 emissoes_kgCO2
1	Sede I BB -> Agência 504 Norte (demanda: 1) -> Agência UnB (demanda: 2) -> Sede I BB	24.7	11.41	0.951	5.89	2.28
2	Sede I BB -> Agência 214 Norte (demanda: 2) -> Agência 316 Norte (demanda: 1) -> Sede I BB	26.9	18.69	1.558	9.64	3.74
3	Sede I BB -> Agência 516 Sul (demanda: 1) -> Agência 504 Sul (demanda: 2) -> Sede I BB	37.5	19.32	1.61	9.97	3.86
4	Sede I BB -> Agência PR (demanda: 1) -> Agência 203 Sul (demanda: 1) -> Sede I BB	23.1	12.27	1.022	6.33	2.45

4 rows | 1.91s runtime

Refre

Table

	1.2 total_tempo_min	1.2 total_distancia_km	1.2 total_combustivel_l	1.2 total_custo_combustivel_R$	1.2 total_emissoes_kgCO2
1	112.2	61.69	5.14	31.83	12.33

UnB

databricks

+

Google Maps

8. Experimentos realizados

Cenário: Frota de 3 veículos partindo da Sede I BB para atender 8 agências.

Execução:

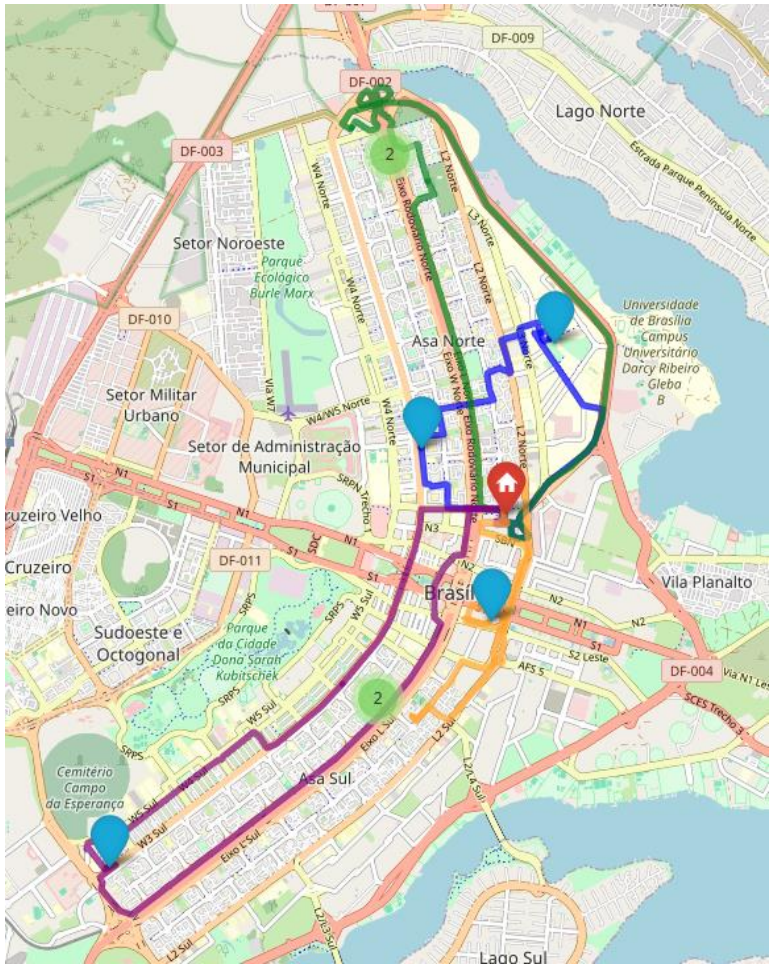
- Geração da matriz de distâncias (Google API).
- Aplicação dos algoritmos (Savings + 2-opt + SA).
- Cálculo do tempo, distância, combustível e emissões por rota.

Resultados (exemplo):

- Rotas finais: 4 rotas otimizadas com custo total ≈ 112 min.
- Combustível total: ≈ R$ 31.
- Emissões: ≈ 12 kg CO_2.

Visualização:

- Tabela estruturada pela DataFrame.
- Mapa interativo com rotas reais.



9. Relevância da disciplina AED

Conceitos Aplicados:

- **Grafos:** Modelagem da rede viária do Plano Piloto.
- **Matrizes e listas:** Armazenamento de dados (distâncias, demandas)
- **Algoritmos Aproximados:** Heurísticas para resolver problemas NP-difíceis.
- **Análise de Complexidade:** deduzir analiticamente a complexidade.

Importância prática: AED fornece a base para estruturar e resolver problemas reais de logística, como o VRP.

"O algoritmo Savings tem complexidade $O(n^2)$ pois precisa calcular e ordenar as 'economias' para todos os pares de n clientes. Já o 2-opt, em cada iteração, testa a troca de todos os pares de arestas, o que também resulta em uma complexidade de $O(n^2)$ por rota."

10. Conclusão e Futuras

Principais aprendizados:

- AED é essencial para modelagem de problemas complexos como VRP.
- Heurísticas combinadas com APIs (Google Maps) permitem soluções práticas.

Resultados obtidos:

- Redução de tempo e custo operacional (~112 min, R$ 31,83).
- Mensuração do impacto ambiental (~12,33 kg CO_2).

Potenciais avanços:

- Rotas dinâmicas com dados de trânsito em tempo real.
- Frota heterogênea (ex.: motos para malotes menores).
- Uso de Machine Learning para prever demandas com base em histórico.

11. Conclusão e Referências

1. **CORMEN, T. H. et al.** *Algoritmos: Teoria e Prática*. 3ª ed. Rio de Janeiro: Campus, 2012.
2. **GERSTING, J. L.** *Fundamentos Matemáticos para a Ciência da Computação*. 3ª ed. Rio de Janeiro: LTC, 2004.
3. **Bibliotecas de Software:** Python, Pandas, Folium, e googlemaps.
4. **GOOGLE.** *Google Maps Platform Documentation*. Disponível em: https://developers.google.com/maps/documentation. Acesso em: 25 ago. 2025.
5. **DATABRICKS.** *Databricks Community Edition*. Disponível em: https://community.cloud.databricks.com/. Acesso em: 25 ago. 2025.
6. **ISHIKAWA, E.** Aulas da disciplina Algoritmos e Estruturas de Dados (PPCA2211). Universidade de Brasília, Brasília, 2º semestre de 2025.

UnB

OBRIGADO!