



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Otimização de Rotas de Transporte Público
Utilizando um Sistema Híbrido de Colônias de
Formigas: Um Estudo de Caso para o Campus Darcy
Ribeiro da UnB**

Gabriel Amaro Baxto da Silva

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Marcelo Ladeira

Brasília
2025



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Otimização de Rotas de Transporte Público
Utilizando um Sistema Híbrido de Colônias de
Formigas: Um Estudo de Caso para o Campus Darcy
Ribeiro da UnB**

Gabriel Amaro Baxto da Silva

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Marcelo Ladeira (Orientador)
Departamento de Ciência da Computação, UnB

Prof. Dr. Claus de Castro Aranha Prof. Dr. Yuri Cossich Lavinas
Universidade de Tsukuba Université Toulouse 1 Capitole

Prof. Dr. Guilherme Novaes Ramos
Universidade de Brasília

Prof.a Dr. Marcelo Grandi Mandelli
Coordenadora do Bacharelado em Ciência da Computação

Brasília, 31 de julho de 2025

Dedicatória

Aos meus pais, Rosana e Welinton, pelo amor incondicional e pela força silenciosa em cada passo da minha caminhada.

Agradecimentos

Agradeço, em primeiro lugar, ao meu orientador, Prof. Dr. Marcelo Ladeira, pela paciência, apoio e pelas discussões sempre instigantes que enriqueceram este trabalho. Sua orientação foi essencial para que este projeto alcançasse a maturidade técnica e conceitual necessária.

Aos membros da banca — Prof. Dr. Claus de Castro Aranha, Prof. Dr. Yuri Cossich Lavinas, Prof. Dr. Guilherme Novaes Ramos — pela generosidade em dedicar seu tempo, pelas contribuições valiosas e pelos olhares atentos e críticos que elevaram a qualidade deste trabalho.

Aos colegas e professores do Departamento de Ciência da Computação da UnB, que contribuíram, direta ou indiretamente, com minha formação ao longo da graduação.

À Universidade de Brasília e ao Prof. Dr. Pastor Willy, pelo suporte acadêmico e pela oportunidade de desenvolver pesquisa científica no âmbito do PIBITI, que teve papel fundamental na construção da base metodológica deste trabalho.

Resumo

Este trabalho desenvolve, implementa e analisa uma arquitetura de otimização para o Problema de Roteamento de Veículos (VRP) com múltiplos objetivos, baseada em um sistema de Otimização por Colônia de Formigas (ACO) com múltiplas colônias. Aplicada a um estudo de caso de planejamento de rotas de ônibus no campus Darcy Ribeiro da Universidade de Brasília (UnB), a metodologia é estruturada em três estágios. Primeiramente, uma etapa de pré-processamento de dados geográficos modela a rede viária e formaliza restrições operacionais, como a de pares de paradas opostas. Em segundo lugar, um meta-grafo é construído como uma estratégia de redução de complexidade, encapsulando as restrições diretamente em sua estrutura para criar um espaço de busca válido. O núcleo da contribuição reside no terceiro estágio: um otimizador que emprega duas colônias de formigas paralelas e especialistas, ACS-Time e ACS-Vehicle, focadas em minimizar, respectivamente, a distância total e o número de veículos. Um controlador central gerencia a interação entre as colônias através de uma tabela de feromônios compartilhada, guiando a busca por soluções de compromisso (*trade-off*). Os resultados indicam que a arquitetura proposta é capaz de gerar rotas viáveis e logisticamente coerentes para o problema complexo e com restrições, demonstrando o potencial da cooperação entre colônias especialistas.

Palavras-chave: Otimização de rotas; Sistema Multi-Colônia; Otimização por Colônia de Formigas; Problema de Roteamento de Veículos; Meta-grafo.

Abstract

This work develops, implements, and analyzes a multi-objective optimization architecture for the Vehicle Routing Problem (VRP), based on a multi-colony Ant Colony Optimization (ACO) system. Applied to a bus route planning case study at the Darcy Ribeiro campus of the University of Brasília (UnB), the methodology is structured in three stages. First, a geographic and topological data preprocessing step formalizes the road network and its operational constraints, such as forbidden pairs of opposite stops. Second, a meta-graph is constructed as a complexity reduction strategy, embedding these constraints directly into its structure to create a valid search space. The core contribution lies in the third stage: an optimizer that employs two specialized and parallel ant colonies, ACS-Ti me and ACS-Vehicle, focused on minimizing total distance and fleet size, respectively. A central controller manages their interaction via a shared pheromone table, guiding the search towards trade-off solutions. The results indicate that the proposed architecture is capable of generating feasible and logistically coherent routes for the complex, constrained problem, demonstrating the potential of cooperation between specialized colonies.

Keywords: Route Optimization; Multi-Colony System; Ant Colony Optimization; Vehicle Routing Problem; Meta-graph.

Sumário

1	Introdução	1
1.1	Contextualização e Definição do Problema	1
1.2	Objetivos	2
1.3	Justificativa e Contribuições	3
1.4	Estrutura do Documento	3
2	Fundamentação Teórica	4
2.1	Otimização Combinatória e o Problema de Roteamento de Veículos	4
2.1.1	O Problema do Caixeiro Viajante como Caso Fundamental	5
2.1.2	Variantes do VRP	5
2.2	Abordagens de Solução: Meta-heurísticas	5
2.3	Otimização por Colônia de Formigas (ACO)	6
2.3.1	Inspiração Biológica e Estigmergia	6
2.3.2	O Algoritmo Ant Colony System (ACS)	7
2.4	Estratégias Híbridas e Múltiplas Colônias	8
2.4.1	Abordagem Multi-Objetivo	8
2.4.2	Arquitetura Multi-Colônia com Aptidão Híbrida	8
3	Metodologia e Sistema Proposto: Sistema de Colônia de Formigas	10
3.1	Definição Formal do Problema	10
3.2	Visão Geral da Arquitetura	11
3.3	Pré-processamento e Modelagem do Grafo	11
3.3.1	Aquisição e Limpeza do Grafo <i>OSM</i>	11
3.3.2	Deteção de Paradas Opostas	12
3.4	Construção do Meta-Grafo	13
3.4.1	Definição de Nós e Arestas	13
3.4.2	Filtragem de Arestas para Garantia de Restrições	13
3.5	O Otimizador Híbrido Multi-Colônia	13
3.5.1	O Controlador (<i>ACS-Controller</i>) e a Função de Qualidade Global	15

3.5.2 As Colônias Especialistas: <i>ACS-Time</i> e <i>ACS-Vehicles</i>	16
3.5.3 O Ciclo de Vida de uma Formiga: Construindo uma Solução Completa	16
4 Detalhes da Implementação	17
4.1 Arquitetura de Software e Tecnologias	17
4.1.1 Módulos e Classes Principais	17
4.1.2 Bibliotecas Utilizadas	18
4.2 Fluxo de Execução do Sistema	19
4.3 Verificação da Corretude e Reprodutibilidade	19
4.3.1 Verificação da Implementação	19
4.3.2 Reprodutibilidade	20
5 Experimentos e Análise de Resultados	21
5.1 Configuração Experimental	21
5.1.1 Ambiente e Reprodutibilidade	21
5.1.2 Parâmetros, Justificativas e Condições de Parada	22
5.1.3 Cenários de Teste e Algoritmos de Referência	22
5.2 Análise Comparativa de Desempenho	23
5.2.1 Análise em Cenário de Baixa Complexidade (6 Paradas)	23
5.2.2 Análise em Cenário de Maior Complexidade (13 Paradas)	24
5.2.3 Análise da Dinâmica de Convergência e Qualidade da Solução	24
6 Discussão dos Resultados	28
6.1 Interpretação dos Resultados	28
6.2 Análise da Solução e Implicações Práticas	29
6.3 Contribuições Principais do Trabalho	29
6.4 Limitações do Trabalho	30
7 Conclusão e Trabalhos Futuros	31
7.1 Trabalhos Futuros	32
Referências	33

Lista de Figuras

2.1 Ilustração do mecanismo de estigmergia. O caminho mais curto (B) é percorrido mais vezes em um mesmo intervalo de tempo, acumulando mais feromônio e se tornando mais atrativo.	7
2.2 Diagrama conceitual da arquitetura multi-colônia com função de qualidade híbrida. As colônias especialistas propõem soluções diversas. O controlador as avalia com uma única função de qualidade ponderada e atualiza o feromônio compartilhado com base na melhor proposta.	9
3.1 Fluxograma geral das fases de desenvolvimento do sistema de otimização de rotas. O sistema abrange desde a aquisição e modelagem dos dados geográficos até a otimização das rotas e a análise dos resultados.	12
3.2 Exemplo de Meta-Grafo construído com nós representando paradas de ônibus e arestas representando os caminhos mais curtos e válidos entre elas. . .	15
4.1 Diagrama de classes simplificado da arquitetura do otimizador. O <code>ACSCController</code> gerencia as instâncias das colônias, que por sua vez utilizam funções de módulos utilitários.	18
4.2 Fluxograma do processo de execução do sistema, desde o carregamento dos dados até a geração das saídas.	19
5.1 Visualização da melhor solução encontrada para 13 paradas. As diferentes cores representam as três rotas que compõem a solução de 15.636,58 metros.	25
5.2 Gráfico de convergência mostrando a evolução do valor da função de qualidade da melhor solução global ao longo das iterações.	26
5.3 Gráfico comparativo da convergência do número de rotas (veículos) propostas por cada colônia.	26
5.4 Gráfico comparativo da convergência da distância total das soluções propostas por cada colônia.	27

Lista de Tabelas

4.1 Principais bibliotecas utilizadas no desenvolvimento do sistema.	18
5.1 Parâmetros padrão utilizados nos algoritmos ACO.	22
5.2 Comparativo de desempenho para o cenário de 6 paradas.	23
5.3 Comparativo de desempenho para o cenário de 13 paradas.	24

Capítulo 1

Introdução

1.1 Contextualização e Definição do Problema

A gestão da mobilidade em ecossistemas complexos e de grande escala, como campi universitários, apresenta desafios logísticos significativos. O campus Darcy Ribeiro da Universidade de Brasília (UnB), com sua vasta extensão territorial e intenso fluxo diário de usuários, serve como um microcosmo ideal para o estudo de sistemas de transporte. Um planejamento de rotas eficiente é fundamental não apenas para garantir a agilidade dos usuários, mas também para otimizar a alocação de recursos, minimizar custos operacionais e reduzir o impacto ambiental associado ao consumo de combustível.¹

Este trabalho se inspira e busca contribuir para um esforço mais amplo de transformação do campus em um "Smart Campus", alinhando-se a iniciativas prévias de pesquisa que exploraram o uso de tecnologias como IoT e IA para monitoramento da demanda de passageiros. Ao focar na otimização de rotas, esta monografia complementa esses esforços, fornecendo uma peça fundamental para a criação de um sistema de transporte público mais eficiente e adaptativo para a comunidade acadêmica.

Este trabalho aborda o desafio de projetar rotas de transporte sob o prisma da otimização combinatória. O problema é formalizado como uma variante do Problema de Roteamento de Veículos (VRP), um conhecido problema NP-difícil na literatura de pesquisa operacional [1]. A nossa formulação estende o modelo clássico ao incorporar restrições topológicas e operacionais específicas do cenário, notadamente a regra de pares de paradas opostas. Essa restrição proíbe que um mesmo veículo atenda, em uma única rota, paradas localizadas em lados opostos de uma mesma via, refletindo uma limitação operacional comum para evitar manobras complexas ou perigosas.

¹Para a elaboração deste trabalho, foram utilizadas ferramentas de Inteligência Artificial como assistentes para tarefas auxiliares, incluindo a geração de comentários de código, correção ortográfica e padronização de formatos de referência. Todo o desenvolvimento conceitual, implementação do algoritmo e análise dos resultados são de autoria de Gabriel Amaro Baxto.

A complexidade inerente ao problema demanda o uso de meta-heurísticas, que são estratégias de otimização capazes de explorar eficientemente um vasto espaço de soluções para encontrar resultados de alta qualidade em um tempo computacionalmente viável. Diante disso, este trabalho propõe uma solução baseada na Otimização por Colônia de Formigas (ACO), uma meta-heurística bioinspirada reconhecida por sua adequação a problemas de roteamento em grafos. A abordagem se destaca por não apenas aplicar o ACO, mas por desenvolver uma arquitetura híbrida com múltiplas colônias, onde a inteligência coletiva é direcionada para otimizar múltiplos objetivos conflitantes simultaneamente.

1.2 Objetivos

O objetivo geral deste trabalho é projetar, implementar e avaliar um sistema computacional para a otimização de rotas de transporte com múltiplos objetivos, utilizando uma arquitetura híbrida de colônias de formigas. O sistema visa gerar um conjunto de rotas que equilibre a eficiência operacional, medida pela minimização do número de veículos, e a qualidade do serviço, medida pela minimização da distância total percorrida, respeitando as restrições do problema.

Para alcançar este objetivo principal, foram definidos os seguintes objetivos específicos:

- Modelar o problema de roteamento do campus UnB, incluindo o pré-processamento de dados geográficos para a criação de uma representação de grafo da rede viária.
- Projetar e implementar uma estrutura de meta-grafo que abstraia a rede viária, incorporando a restrição de pares de paradas opostas diretamente em sua construção para reduzir o espaço de busca.
- Desenvolver uma arquitetura de otimização com duas colônias de formigas especialistas: uma focada na minimização da distância total e outra na minimização do número de veículos.
- Implementar um controlador central que gerencie a interação entre as colônias, utilizando uma tabela de feromônios compartilhada e uma função de qualidade para guiar a busca por soluções de compromisso.
- Avaliar o desempenho do sistema proposto através da comparação com algoritmos de referência — incluindo uma abordagem de força bruta, uma heurística construtiva e uma implementação de ACO de colônia única — em cenários de complexidade variável.
- Analisar a dinâmica de convergência do sistema, investigando a interação e a contribuição de cada colônia especialista ao longo do processo de otimização.

1.3 Justificativa e Contribuições

A otimização de rotas de transporte é um campo de pesquisa consolidado. A escolha pela meta-heurística ACO neste trabalho foi motivada por sua natural adequação a problemas de roteamento em grafos, pela vasta literatura que valida sua aplicação em variações do VRP [2], e pela afinidade prévia com a técnica, que permitiu um foco maior no desenvolvimento de uma arquitetura customizada.

A principal contribuição científica deste trabalho reside na concepção e análise de um sistema multiagente cooperativo, onde colônias com objetivos distintos colaboram para encontrar soluções de compromisso. Esta abordagem é particularmente relevante para gestores de transporte, que frequentemente precisam tomar decisões baseadas em *trade-offs* entre custo e nível de serviço.

As contribuições deste trabalho podem ser sumarizadas como:

1. **Modelagem de Restrições Específicas:** A formalização e a implementação do conceito de "pares de paradas opostas" e sua integração na construção de um meta-grafo filtrado.
2. **Arquitetura de Otimização Multi-Colônia:** O projeto e a avaliação de um sistema ACO com colônias especialistas e um controlador central, comparando seu desempenho com uma abordagem de colônia única.
3. **Análise Comportamental:** A avaliação empírica de como duas colônias com objetivos conflitantes interagem por meio de um mecanismo de feromônio compartilhado, fornecendo *insights* sobre o comportamento de meta-heurísticas híbridas.

1.4 Estrutura do Documento

Este documento está organizado em sete capítulos. O **Capítulo 2** apresenta a fundamentação teórica sobre o Problema de Roteamento de Veículos, Meta-heurísticas e a Otimização por Colônia de Formigas. O **Capítulo 3** detalha a metodologia e o sistema proposto, descrevendo a arquitetura geral, o pré-processamento de dados, a construção do meta-grafo e a formulação do otimizador. O **Capítulo 4** aborda os detalhes da implementação e da verificação do sistema. O **Capítulo 5** apresenta o protocolo experimental e a análise dos resultados, com foco na convergência e na comparação de desempenho. A **Discussão** dos resultados é explorada no **Capítulo 6**. Finalmente, o **Capítulo 7** sintetiza as conclusões e aponta direções para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos que sustentam o desenvolvimento deste trabalho. A Seção 2.1 introduz o campo da otimização combinatória e define formalmente o Problema de Roteamento de Veículos (VRP), contextualizando-o como um desafio central em pesquisa operacional. A Seção 2.2 discute a necessidade de meta-heurísticas para resolver problemas dessa natureza e justifica a escolha da abordagem adotada. A Seção 2.3 detalha a Otimização por Colônia de Formigas (ACO) e o algoritmo Ant Colony System (ACS). Por fim, a Seção 2.4 explora a aplicação de estratégias de múltiplas colônias, que constituem a base da arquitetura proposta neste trabalho.

2.1 Otimização Combinatória e o Problema de Roteamento de Veículos

A otimização combinatória é uma área da matemática e da ciência da computação que busca encontrar uma solução ótima a partir de um conjunto finito, porém muitas vezes vasto, de possíveis soluções. O *Problema de Roteamento de Veículos* (*Vehicle Routing Problem* - VRP) é um dos desafios mais estudados nesse campo, com vastas aplicações em logística, transporte e planejamento de serviços [?]. Em sua forma geral, o VRP busca determinar um conjunto ótimo de rotas para uma frota de veículos a partir de um ou mais depósitos, de modo a atender as demandas de um conjunto de clientes, minimizando um ou mais objetivos, como a distância total percorrida ou o número de veículos utilizados.

A complexidade do VRP reside em sua natureza NP-difícil [3], o que significa que encontrar uma solução ótima garantida para instâncias de tamanho realista é, em geral, computacionalmente intratável. Não existe um algoritmo conhecido capaz de resolver todos os casos do problema em tempo polinomial. Isso exige o uso de métodos aproximados para a maioria das aplicações práticas.

2.1.1 O Problema do Caixeiro Viajante como Caso Fundamental

O VRP pode ser visto como uma generalização do *Problema do Caixeiro Viajante* (*Traveling Salesman Problem* - TSP), que é talvez o problema mais icônico da otimização combinatória [4]. O TSP consiste em encontrar a rota mais curta possível que visite um conjunto de cidades exatamente uma vez, retornando ao ponto de partida. Enquanto o TSP lida com uma única rota e um único "veículo", o VRP estende o problema para múltiplas rotas e veículos, frequentemente adicionando restrições como a capacidade dos mesmos. O estudo do TSP pavimentou o caminho para o desenvolvimento de muitas das técnicas aplicadas hoje ao VRP.

2.1.2 Variantes do VRP

O modelo básico do VRP foi estendido para inúmeras variantes que buscam modelar restrições do mundo real com maior fidelidade [5]. O termo "*Rich VRP*" é frequentemente utilizado para descrever problemas que combinam múltiplas dessas restrições [?]. As variantes mais relevantes para este trabalho incluem:

- **CVRP (*Capacitated VRP*)**: Cada veículo possui uma capacidade limitada, e a soma das demandas dos clientes em uma rota não pode exceder essa capacidade. Esta é a variante central adotada neste estudo.
- **VRPTW (*VRP with Time Windows*)**: Cada cliente deve ser atendido dentro de uma janela de tempo específica.
- **Problemas Multi-Objetivo (*Multi-Objective VRP*)**: Problemas que buscam otimizar simultaneamente dois ou mais objetivos conflitantes, como custo e tempo de entrega.

Este trabalho foca em uma variação do CVRP, introduzindo restrições topológicas adicionais que serão detalhadas no Capítulo 3.

2.2 Abordagens de Solução: Meta-heurísticas

Dada a complexidade NP-difícil do VRP, métodos exatos que garantem a otimalidade são viáveis apenas para instâncias pequenas. Para problemas de maior escala, as **meta-heurísticas** surgem como a abordagem predominante para encontrar soluções de alta qualidade em um tempo de computação razoável [6].

Uma meta-heurística é uma estrutura algorítmica de alto nível que pode ser adaptada para uma ampla gama de problemas de otimização. Seu objetivo é guiar de forma inteligente o processo de busca no espaço de soluções, equilibrando duas forças fundamentais:

- **Intensificação (*Exploitation*):** A busca focada em regiões do espaço de soluções que já se mostraram promissoras, a fim de refinar e encontrar ótimos locais.
- **Diversificação (*Exploration*):** A exploração de novas e diversas regiões do espaço de soluções, para evitar a convergência prematura para ótimos locais e aumentar a chance de encontrar o ótimo global.

Em contraste com as meta-heurísticas, existem também as **heurísticas construtivas**, como a do **Vizinho Mais Próximo**. Essa heurística constrói uma solução passo a passo, escolhendo sempre a opção localmente ótima (neste caso, a parada mais próxima ainda não visitada). Embora muito rápidas, essas heurísticas frequentemente convergem para soluções de baixa qualidade e servem como uma importante linha de base (*baseline*) para avaliar a eficácia de algoritmos mais sofisticados.

Exemplos de meta-heurísticas bem conhecidas incluem Algoritmos Genéticos, Recozimento Simulado (*Simulated Annealing*), Busca Tabu e, o foco deste trabalho, a Otimização por Colônia de Formigas. A escolha pelo ACO foi motivada por sua natural adequação a problemas de roteamento em grafos e pela vasta literatura que valida sua aplicação em variações do VRP, permitindo um foco maior no desenvolvimento de uma arquitetura customizada para as restrições do nosso problema.

2.3 Otimização por Colônia de Formigas (ACO)

A *Otimização por Colônia de Formigas* (*Ant Colony Optimization* - ACO) é uma meta-heurística populacional bioinspirada no comportamento de formigas reais em busca de alimento [2]. Proposta originalmente por Dorigo em sua tese de doutorado [7], a técnica se mostrou extremamente eficaz para problemas de otimização combinatória, especialmente os que podem ser representados em grafos.

2.3.1 Inspiração Biológica e Estigmergia

O princípio fundamental do ACO é a *estigmergia*, uma forma de comunicação indireta e não-centralizada. Formigas reais, ao se moverem, depositam uma substância química chamada **feromônio**, marcando seu caminho. Outras formigas são capazes de perceber esse feromônio e tendem a seguir os caminhos com maior concentração. Como o feromônio evapora com o tempo, caminhos mais curtos, que são percorridos mais rapidamente, acumulam feromônio a uma taxa maior, tornando-se cada vez mais atrativos. Esse mecanismo de *feedback* positivo permite que a colônia, como um todo, convirja para o caminho mais curto, como ilustrado na Figura 2.1.

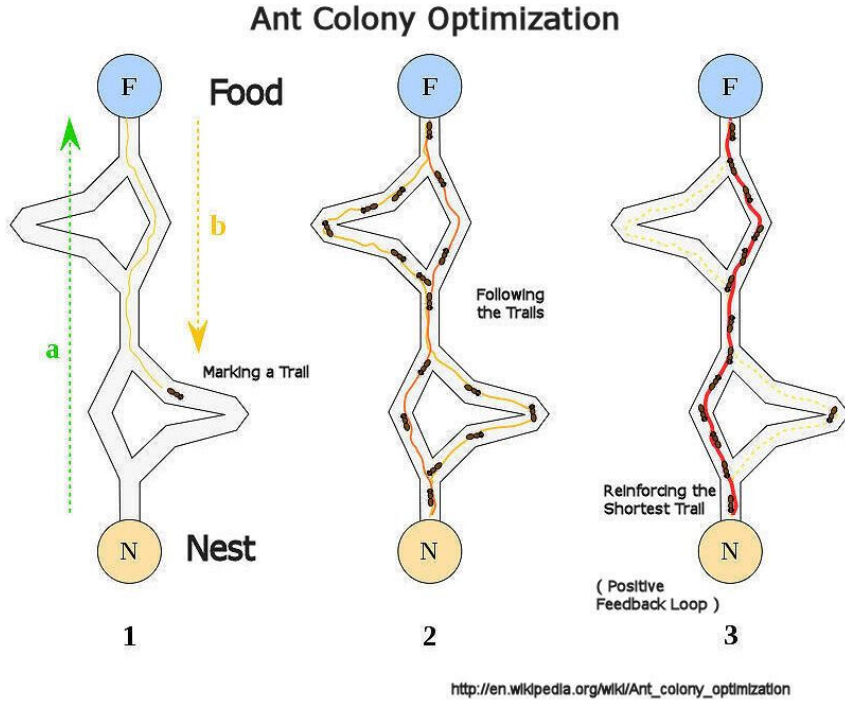


Figura 2.1: Ilustração do mecanismo de estigmergia. O caminho mais curto (B) é percorrido mais vezes em um mesmo intervalo de tempo, acumulando mais feromônio e se tornando mais atrativo.

2.3.2 O Algoritmo Ant Colony System (ACS)

Neste trabalho, utilizamos uma variante popular e eficaz do ACO, o *Ant Colony System* (ACS) [8]. No ACS, "formigas artificiais" constroem soluções de forma probabilística. O processo de decisão de uma formiga pode ser modelado como uma Cadeia de Markov, onde a probabilidade de transitar para o próximo estado (nó) depende apenas do estado atual. A decisão de uma formiga k no nó i de se mover para o nó j é governada pela seguinte regra de transição de estado:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, \quad \text{se } j \in \mathcal{N}_i^k \quad (2.1)$$

Onde cada termo representa:

- $p_{ij}^k(t)$: a probabilidade da formiga k escolher a aresta (i, j) na iteração t .
- $\tau_{ij}(t)$: a quantidade de feromônio na aresta (i, j) na iteração t .
- η_{ij} : a informação heurística da aresta (i, j) , geralmente o inverso da distância $(1/d_{ij})$.
- α, β : parâmetros que controlam a influência relativa do feromônio (aprendizado coletivo) e da heurística (ganância local).

- $\mathcal{N}_i^k$: o conjunto de nós vizinhos ainda não visitados pela formiga k a partir do nó i .

Após cada formiga construir uma solução, os níveis de feromônio são atualizados. O ACS emprega tanto uma atualização local (para diversificação) quanto uma atualização global, que reforça as arestas da melhor solução encontrada até o momento, promovendo a intensificação:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}^{\text{best}} \quad (2.2)$$

Onde $\rho \in (0, 1]$ é a taxa de evaporação do feromônio, e $\Delta\tau_{ij}^{\text{best}}$ é o reforço depositado, tipicamente proporcional à qualidade da melhor solução encontrada.

2.4 Estratégias Híbridas e Múltiplas Colônias

Embora o algoritmo ACS seja poderoso, sua performance em problemas complexos pode ser aprimorada através de estratégias de hibridização. A literatura recente explora extensivamente a combinação do ACO com outras heurísticas ou a utilização de múltiplas populações de formigas para enriquecer o processo de busca [9, 10, 11].

2.4.1 Abordagem Multi-Objetivo

Problemas de roteamento no mundo real raramente se resumem a uma única métrica. Um gestor de frota precisa, simultaneamente, minimizar custos e maximizar a qualidade do serviço. Para lidar com essa natureza multi-objetivo, uma técnica comum é a **agregação de objetivos** em uma única **função de qualidade** (ou função de aptidão). Esta função calcula um valor escalar que representa a "qualidade" geral (Q) de uma solução, conforme a equação genérica:

$$Q(\mathbf{x}) = \sum_{i=1}^n w_i \cdot f_i(\mathbf{x}) \quad (2.3)$$

Onde $\mathbf{x}$ é uma solução candidata, cada $f_i(\mathbf{x})$ é o valor de um objetivo (e.g., número de rotas, distância total), e cada w_i é um peso que define sua importância relativa. Isso permite explorar diferentes soluções de compromisso (*trade-offs*).

2.4.2 Arquitetura Multi-Colônia com Aptidão Híbrida

A principal inovação metodológica deste trabalho é a combinação de uma arquitetura de múltiplas colônias com uma função de qualidade híbrida. O sistema proposto, ilustrado na Figura 2.2, emprega duas colônias de formigas que, embora operem sob as mesmas re-

gras do ACS, são conceitualmente guiadas por "interesses" distintos, focando em objetivos diferentes durante a construção de suas soluções.

A cada iteração, ambas as colônias propõem uma solução completa. Um **controlador central** avalia as duas propostas utilizando a função de qualidade da Equação 2.3. A solução com o maior *score* é selecionada como a melhor daquela iteração. Este modelo é motivado por duas razões principais:

1. **Diversidade na Busca:** As colônias exploram o espaço de soluções a partir de "vieses" diferentes, aumentando a diversidade das soluções geradas e diminuindo a chance de convergência prematura para ótimos locais.
2. **Aprendizado Compartilhado:** Ambas as colônias atualizam uma *única tabela de feromônios compartilhada*. Isso cria um mecanismo de aprendizado cooperativo: os bons caminhos encontrados por uma colônia (e.g., uma rota curta) podem ser aproveitados pela outra, mesmo que seu objetivo primário seja diferente. Esta abordagem se inspira em trabalhos que demonstram a eficácia de múltiplas populações na resolução de VRPs complexos [10, 12].

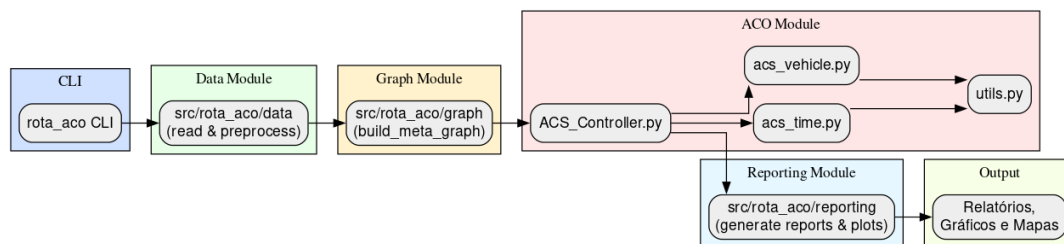


Figura 2.2: Diagrama conceitual da arquitetura multi-colônia com função de qualidade híbrida. As colônias especialistas propõem soluções diversas. O controlador as avalia com uma única função de qualidade ponderada e atualiza o feromônio compartilhado com base na melhor proposta.

Capítulo 3

Metodologia e Sistema Proposto: Sistema de Colônia de Formigas

Este capítulo detalha a metodologia adotada para o desenvolvimento do sistema de otimização de rotas. Inicia-se com uma definição formal do problema, estabelecendo as variáveis, objetivos e restrições. Em seguida, descreve-se o processo de aquisição e tratamento dos dados, a construção do meta-grafo e, por fim, a arquitetura do otimizador híbrido multi-colônia.

3.1 Definição Formal do Problema

O problema abordado neste trabalho é modelado como uma variante do Problema de Roteamento de Veículos com Capacidade (CVRP), definido sobre um grafo $G = (V, E)$, onde V é o conjunto de paradas de ônibus e E é o conjunto de vias. O objetivo é encontrar um conjunto de rotas $R = \{r_1, r_2, \dots, r_k\}$ que minimize uma função de custo, sujeito a um conjunto de restrições.

Variáveis Principais:

- **Frota de Veículos (K):** Um conjunto de veículos homogêneos, onde $|K|$ é o número total de veículos disponíveis.
- **Capacidade do Veículo (C):** A capacidade máxima de passageiros que cada veículo pode transportar.
- **Demanda das Paradas (d_i):** O número de passageiros a serem coletados na parada $i \in V$.
- **Distância ($dist_{ij}$):** O custo (distância ou tempo) para viajar da parada i para a parada j através de uma aresta $(i, j) \in E$.

Objetivos da Otimização: O sistema busca otimizar dois objetivos conflitantes, que são combinados em uma função de qualidade híbrida:

1. **Minimizar o Número de Veículos ($|R|$):** Utilizar a menor quantidade de rotas (veículos) possível para atender a demanda.
2. **Minimizar a Distância Total Percorrida:** Minimizar a soma das distâncias de todas as rotas na solução, $\sum_{r \in R} \sum_{(i,j) \in r} dist_{ij}$.

Restrições Fundamentais: Toda solução válida deve aderir às seguintes restrições:

- **Restrição de Capacidade:** Para cada rota $r \in R$, a soma das demandas das paradas visitadas não pode exceder a capacidade do veículo: $\sum_{i \in r} d_i \leq C$.
- **Restrição de Cobertura:** Cada parada com demanda positiva deve ser visitada por exatamente uma rota.
- **Restrição de Pares Opostos:** Se um par de paradas (u, v) é definido como oposto, elas não podem pertencer à mesma rota r .

3.2 Visão Geral da Arquitetura

O sistema proposto segue uma arquitetura modular em três etapas principais, visando a otimização de rotas de transporte no campus da UnB. Esta abordagem garante uma representação precisa da geografia e infraestrutura de transporte do campus. A Figura 3.1 ilustra o fluxo de processamento, desde a aquisição dos dados brutos até a geração das rotas otimizadas.

3.3 Pré-processamento e Modelagem do Grafo

A fase inicial do sistema concentra-se na coleta e no tratamento dos dados geográficos, transformando-os em um modelo computacional adequado para a otimização.

3.3.1 Aquisição e Limpeza do Grafo *OSM*

O grafo da rede viária do campus Darcy Ribeiro foi extraído da plataforma de dados abertos *OpenStreetMap* (OSM). A biblioteca *OSMnx* foi utilizada para processar esses dados e construir um grafo de ruas (*MultiDiGraph*) que representa a rede viária. O peso de cada aresta no grafo original foi definido como a distância real do segmento de via. Para este estudo, as demandas de cada parada foram simuladas, atribuindo-se valores a partir de uma distribuição uniforme, para permitir um ambiente de teste controlado e reprodutível.

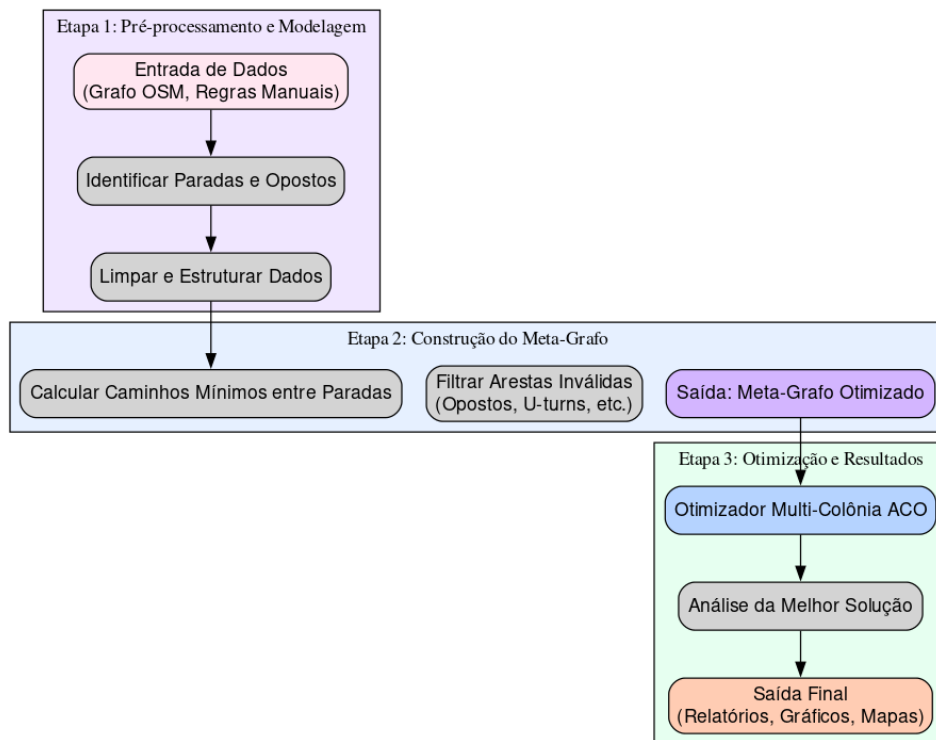


Figura 3.1: Fluxograma geral das fases de desenvolvimento do sistema de otimização de rotas. O sistema abrange desde a aquisição e modelagem dos dados geográficos até a otimização das rotas e a análise dos resultados.

3.3.2 Detecção de Paradas Opostas

Uma das restrições operacionais mais importantes é a existência de paradas em lados opostos de uma mesma via. O acesso a ambas as paradas por um mesmo veículo em uma única rota é logisticamente inviável. O sistema identifica esses **pares de paradas opostas** utilizando uma abordagem combinada:

- **Proximidade Geoespacial:** Paradas muito próximas geograficamente (dentro de uma precisão definida) são consideradas candidatas a opostas.
- **Análise de Vias Comuns:** Verificação se duas paradas candidatas estão conectadas à mesma via principal no grafo OSM.
- **Definição Manual:** Para incorporar conhecimento de especialista e garantir precisão, o sistema permite a definição manual de pares opostos via um arquivo de configuração. Esta abordagem híbrida assegura que o modelo reflita as complexidades do mundo real.

3.4 Construção do Meta-Grafo

A construção do meta-grafo é uma etapa crucial que transforma a complexa rede viária em uma representação mais abstrata e computacionalmente eficiente. Neste grafo simplificado, os nós são apenas as paradas de ônibus, e as arestas representam os caminhos válidos entre elas.

3.4.1 Definição de Nós e Arestas

Os **nós** do meta-grafo representam as paradas de ônibus. As **arestas** do meta-grafo representam os caminhos mais curtos entre cada par de paradas no grafo original. O custo de cada meta-aresta é a distância real do caminho subjacente, calculado com o algoritmo de Dijkstra.

3.4.2 Filtragem de Arestas para Garantia de Restrições

A qualidade do meta-grafo depende da aplicação de filtros rigorosos durante a criação de suas arestas. Para cada par de nós (u, v) , o caminho mais curto entre eles é verificado contra os seguintes critérios:

1. **Arestas entre Opostos Proibidas:** Uma meta-aresta não é criada se u e v forem paradas opostas.
2. **Prevenção de Atalhos sobre Paradas Intermediárias:** Se o caminho entre u e v no grafo original passa por outra parada, a meta-aresta (u, v) é desconsiderada para forçar a visita explícita à parada intermediária.
3. **Filtragem de Caminhos Ineficientes:** Arestas que representam retornos geograficamente ineficientes (*U-turns*) são filtradas. A eficiência do caminho é avaliada comparando seu comprimento com a distância em linha reta (Haversine) entre u e v . Se a razão entre os dois exceder um limiar, a aresta é descartada.

3.5 O Otimizador Híbrido Multi-Colônia

O cerne do sistema é um otimizador ACO que emprega uma arquitetura híbrida de múltiplas colônias. Diferente das abordagens clássicas, este sistema utiliza colônias especializadas para explorar diferentes facetas da otimização, conforme detalhado no Algoritmo 1.

Algorithm 1 Algoritmo de Otimização Híbrido com Múltiplas Colônias

Require: Grafo G , Paradas P , Capacidade C , Parâmetros ACO θ .

Ensure: Melhor conjunto de rotas `melhorSolucaoGlobal`.

```
1: procedure OTIMIZADORMULTICOLONIA
2:   Inicialização:
3:    $\tau_{ij} \leftarrow \tau_0$  para toda aresta  $(i, j) \in G$ ; melhorSolucaoGlobal  $\leftarrow \emptyset$ ;
   melhorQualidadeGlobal  $\leftarrow -\infty$ ;
4:   for  $it = 1$  to MaxIteracoes do
5:     solucoesDistancia  $\leftarrow []$ ; solucoesRotas  $\leftarrow []$ 
6:     for cada formiga  $k$  na colônia ACS-Time do
7:       solucao  $\leftarrow$  CONSTRUIRSOLUCAOCompleta( $G, P, C, \tau, \theta$ )
8:       Adicionar solucao a solucoesDistancia
9:     end for
10:    for cada formiga  $k$  na colônia ACS-Vehicles do
11:      solucao  $\leftarrow$  CONSTRUIRSOLUCAOCompleta( $G, P, C, \tau, \theta$ )
12:      Adicionar solucao a solucoesRotas
13:    end for
14:    melhorSolucaoDist  $\leftarrow$  SELECIONARMELHOR(solucoesDistancia,  $\theta$ )
15:    melhorSolucaoRotas  $\leftarrow$  SELECIONARMELHOR(solucoesRotas,  $\theta$ )
16:    qualidadeDist  $\leftarrow$  CALCULARQUALIDADE(melhorSolucaoDist,  $\theta$ )
17:    qualidadeRotas  $\leftarrow$  CALCULARQUALIDADE(melhorSolucaoRotas,  $\theta$ )
18:    if qualidadeDist  $>$  qualidadeRotas then
19:      melhorDaIteracao  $\leftarrow$  melhorSolucaoDist
20:    else
21:      melhorDaIteracao  $\leftarrow$  melhorSolucaoRotas
22:    end if
23:    if QUALIDADE(melhorDaIteracao)  $>$  melhorQualidadeGlobal then
24:      melhorQualidadeGlobal  $\leftarrow$  QUALIDADE(melhorDaIteracao)
25:      melhorSolucaoGlobal  $\leftarrow$  melhorDaIteracao
26:    end if
27:     $\tau \leftarrow$  ATUALIZARFEROMONIOS( $\tau$ , melhorSolucaoGlobal,  $\theta$ )
28:    if CRITERIODEPARADAATINGIDO( $it$ ) then break
29:    end if
30:  end for
31:  return melhorSolucaoGlobal
32: end procedure
```

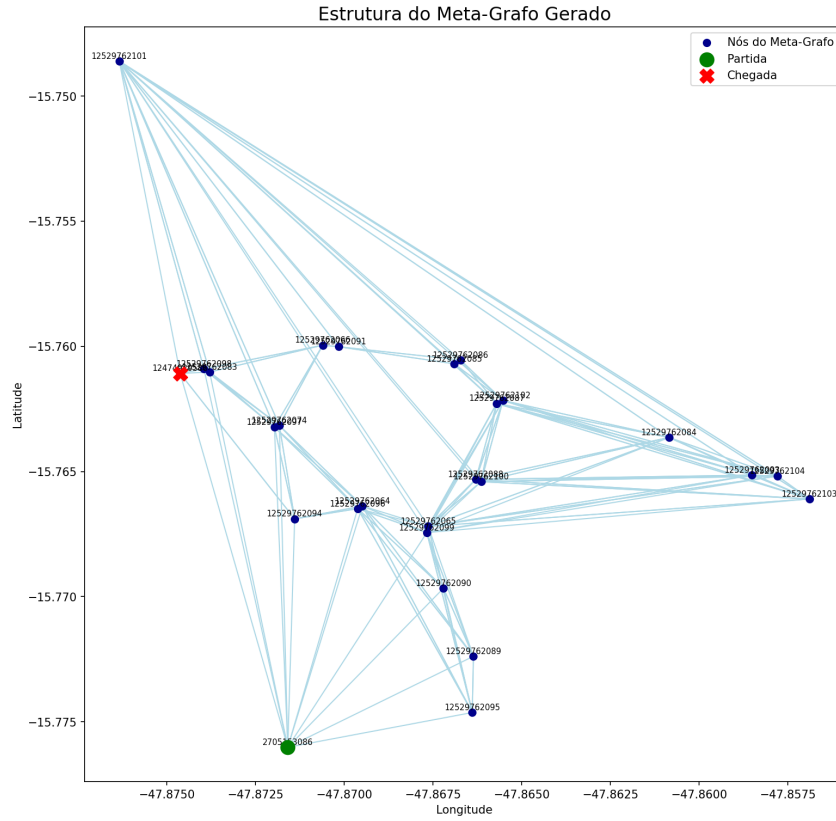


Figura 3.2: Exemplo de Meta-Grafo construído com nós representando paradas de ônibus e arestas representando os caminhos mais curtos e válidos entre elas.

3.5.1 O Controlador (*ACS-Controller*) e a Função de Qualidade Global

O *ACS-Controller* é o módulo central de orquestração do sistema. Sua principal responsabilidade é gerenciar o ciclo iterativo, coordenando a interação entre as colônias.

Função de Qualidade Global (Q): A avaliação das soluções é feita através de uma função de qualidade híbrida, como descrito na Equação 2.3. Esta função combina a cobertura das paradas, o número de rotas e a distância total, ponderando a importância de cada objetivo.

Gerenciamento de Feromônio: O controlador mantém uma única tabela de feromônios compartilhada. Após cada iteração, os feromônios são atualizados globalmente: as arestas pertencentes à melhor solução encontrada (avaliada pela função de qualidade) recebem reforço, enquanto o feromônio nas demais arestas evapora, conforme a Equação 2.2.

CrITÉRIOS de Parada: O controlador monitora o progresso e encerra a otimização com base nos seguintes critérios:

- Atingir um número máximo de iterações.

- Estagnação da solução (a melhor solução não melhora por um determinado número de iterações consecutivas).

3.5.2 As Colônias Especialistas: *ACS-Time* e *ACS-Vehicles*

O sistema emprega duas colônias de formigas paralelas, cada uma com um viés heurístico distinto:

- ***ACS-Time* (antigo *ACS-Time*):** Esta colônia especializa-se em encontrar rotas que minimizem a distância total percorrida. Sua heurística favorece caminhos mais curtos.
- ***ACS-Vehicles* (antigo *ACS-Vehicle*):** Esta colônia foca na minimização do número de veículos. Sua heurística favorece a adição de paradas a uma rota existente, desde que a capacidade não seja violada, em vez de iniciar uma nova rota. A lógica é maximizar a utilização de cada veículo para reduzir o tamanho da frota.

3.5.3 O Ciclo de Vida de uma Formiga: Construindo uma Solução Completa

Cada formiga constrói um conjunto completo de rotas em cada iteração. O processo de construção de uma rota segue os passos do ACS:

1. **Início da Rota:** Cada rota começa no nó de partida.
2. **Seleção da Próxima Parada:** A formiga escolhe probabilisticamente o próximo nó a visitar (Equação 2.1), respeitando as restrições.
3. **Fechamento da Rota:** Uma vez que a capacidade do veículo é atingida ou não há mais paradas válidas para visitar, a rota é finalizada.
4. **Construção Contínua:** A formiga inicia uma nova rota, e o processo se repete até que todas as paradas com demanda tenham sido atendidas.

Parâmetros Principais: Os parâmetros do algoritmo ACO (α , β , ρ) são ajustados para controlar a influência do feromônio e da heurística, bem como a taxa de evaporação. A escolha desses parâmetros impacta diretamente a performance e a qualidade das soluções geradas, sendo sua análise um dos focos dos experimentos.

Capítulo 4

Detalhes da Implementação

Este capítulo descreve os detalhes técnicos da implementação do sistema de otimização de rotas, traduzindo a metodologia apresentada no Capítulo 3 em um artefato de software funcional. A discussão abrange a arquitetura do sistema, os principais módulos e classes, as bibliotecas utilizadas e o fluxo de execução que integra os componentes. Adicionalmente, detalha-se o processo de verificação da corretude do sistema.

4.1 Arquitetura de Software e Tecnologias

O sistema foi desenvolvido integralmente na linguagem de programação **Python** (versão 3.9+), escolhida por seu ecossistema robusto de bibliotecas para ciência de dados, computação científica e manipulação de grafos. A arquitetura do projeto foi estruturada de forma modular para garantir a separação de responsabilidades, a manutenibilidade e a extensibilidade do código.

4.1.1 Módulos e Classes Principais

O código-fonte está organizado em um pacote principal, `rota_aco`, que encapsula as diferentes camadas lógicas do sistema. A Figura 4.1 apresenta um diagrama de classes simplificado, destacando os componentes centrais. O módulo `src/rota_aco/data` é responsável pela aquisição e pré-processamento dos dados. O módulo `src/rota_aco/graph` contém a lógica para a construção do meta-grafo. O núcleo do otimizador reside no pacote `src/rota_aco/aco`, que abriga a classe `ACSController`, orquestradora do processo, e as implementações das colônias especialistas. Finalmente, os pacotes `src/rota_aco/reporting` e `src/rota_aco/cli` implementam, respectivamente, a geração de relatórios e a interface de linha de comando (CLI) para execução dos experimentos.

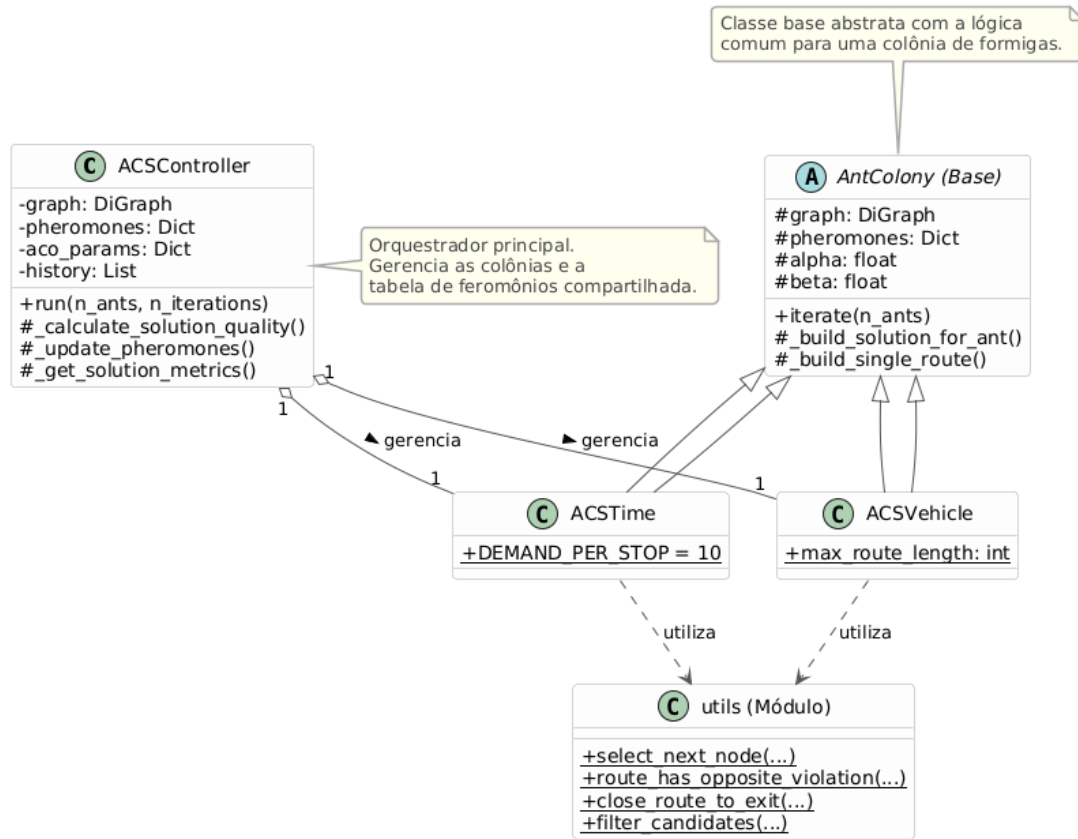


Figura 4.1: Diagrama de classes simplificado da arquitetura do otimizador. O `ACSTController` gerencia as instâncias das colônias, que por sua vez utilizam funções de módulos utilitários.

4.1.2 Bibliotecas Utilizadas

O desenvolvimento do sistema foi suportado por um conjunto de bibliotecas de código aberto do ecossistema Python. A Tabela 4.1 resume as principais dependências e suas funções.

Tabela 4.1: Principais bibliotecas utilizadas no desenvolvimento do sistema.

Biblioteca	Função Principal
NetworkX	Criação, manipulação e estudo de grafos.
Pandas	Análise de dados e manipulação de resultados.
NumPy	Suporte para operações numéricas eficientes.
Matplotlib	Geração de gráficos estáticos para relatórios.
Folium	Criação de visualizações interativas em mapas HTML.
OSMnx	Extração e processamento de dados do OpenStreetMap.

4.2 Fluxo de Execução do Sistema

O fluxo de execução do sistema, orquestrado pela CLI, segue uma sequência lógica que integra os módulos descritos. A Figura 4.2 ilustra este fluxo.

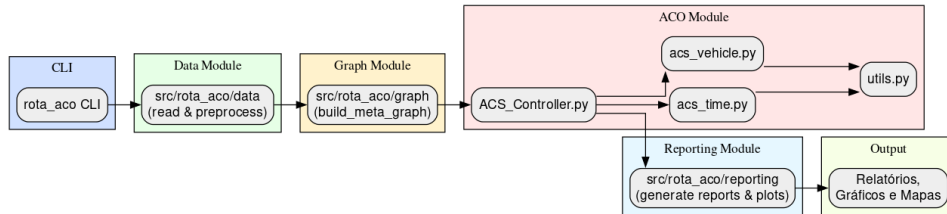


Figura 4.2: Fluxograma do processo de execução do sistema, desde o carregamento dos dados até a geração das saídas.

1. **Carregamento de Dados:** O processo inicia com o carregamento do grafo da rede viária a partir de um arquivo `.graphml`.
2. **Construção do Meta-Grafo:** O módulo de construção do meta-grafo é invocado, recebendo o grafo original e as configurações de restrições.
3. **Instanciação do Otimizador:** O `ACSController` é instanciado com o meta-grafo e os parâmetros do algoritmo.
4. **Loop de Otimização:** O controlador executa o *loop* principal. Em cada iteração, as colônias geram suas propostas, o controlador avalia, atualiza a melhor solução global e gerencia o feromônio.
5. **Geração de Relatórios:** Ao final do *loop*, o controlador retorna a melhor solução, e os módulos de relatório são invocados para gerar os artefatos de saída (relatórios em texto, gráficos, arquivos CSV e mapas interativos).

4.3 Verificação da Corretude e Reprodutibilidade

Garantir a corretude do sistema e a reprodutibilidade dos resultados é fundamental em qualquer trabalho científico. Para isso, foram adotadas as seguintes estratégias.

4.3.1 Verificação da Implementação

Em vez de uma validação formal, que exigiria a comparação com um modelo matemático comprovadamente correto, foi realizado um processo de **verificação** para assegurar que a implementação do software adere à metodologia descrita. Este processo incluiu:

- **Testes Manuais e Asserções:** Funções críticas, como a de detecção de pares opostos e os filtros do meta-grafo, foram verificadas com casos de teste específicos e asserções no código para capturar comportamentos inesperados.
- **Verificação de Conectividade:** Antes de iniciar a otimização, o sistema verifica se o meta-grafo resultante é conexo, garantindo que o problema seja solúvel.
- **Monitoramento de Restrições:** Durante a construção de cada rota, o sistema monitora ativamente a violação de restrições (capacidade e pares opostos). Soluções que violam qualquer restrição são invalidadas.

A combinação dessas verificações com a arquitetura modular permitiu um desenvolvimento iterativo e seguro, aumentando a confiança na corretude do artefato de software.

4.3.2 Reprodutibilidade

Para garantir a transparência e permitir que futuros pesquisadores possam replicar, verificar e estender este trabalho, o código-fonte completo desenvolvido, incluindo os scripts para execução dos experimentos e os dados de entrada utilizados, foi disponibilizado publicamente em um repositório na plataforma GitHub. O repositório pode ser acessado em: <https://github.com/ParaFoxy-xto/busStops>.

Capítulo 5

Experimentos e Análise de Resultados

Este capítulo apresenta a validação empírica do sistema proposto, detalhando o protocolo experimental e a análise aprofundada dos resultados. O objetivo é demonstrar, de forma quantitativa e qualitativa, a eficácia da arquitetura **ACO Híbrido** em comparação com algoritmos de referência representativos. A Seção 5.1 descreve a metodologia dos testes, com foco em rigor, transparência e reprodutibilidade. A Seção 5.2 apresenta uma análise comparativa do desempenho dos algoritmos em cenários de complexidade crescente, estabelecendo o mérito da abordagem proposta. Por fim, a Seção 5.2.3 investiga a dinâmica de convergência do sistema híbrido, fornecendo insights sobre os mecanismos de especialização e cooperação que fundamentam seu desempenho superior.

5.1 Configuração Experimental

Para garantir a validade científica e a comparabilidade dos resultados, todos os testes foram conduzidos sob uma metodologia rigorosa e em um ambiente computacional controlado.

5.1.1 Ambiente e Reprodutibilidade

Os experimentos foram executados em um computador com processador AMD Ryzen 7 5700X, 16 GB de memória RAM DDR4 e sistema operacional Windows 11. O sistema foi implementado em Python 3.12. Para assegurar a **reprodutibilidade** dos resultados, crucial para algoritmos estocásticos, uma semente aleatória fixa (`seed=42`) foi utilizada em todas as execuções. O tempo de execução foi mensurado utilizando a função de alta precisão `time.perf_counter()`, e os resultados de algoritmos estocásticos representam a

média de 10 execuções independentes para garantir a consistência estatística. Em linha com as práticas de ciência aberta, o código-fonte completo e os dados utilizados nos experimentos foram disponibilizados publicamente em um repositório no GitHub.

5.1.2 Parâmetros, Justificativas e Condições de Parada

Os algoritmos baseados em ACO utilizaram os parâmetros de referência detalhados na Tabela 5.1. A escolha desses valores não foi arbitrária; eles foram selecionados com base em recomendações consolidadas na literatura para problemas de VRP e em testes preliminares. O objetivo foi estabelecer uma base de comparação justa, em vez de realizar uma sintonia fina de parâmetros (*parameter tuning*) para cada cenário, o que poderia enviesar os resultados e representa uma direção para trabalhos futuros. A principal **condição de parada** para os algoritmos foi a conclusão de 10 iterações, um número suficiente para observar a dinâmica de convergência nos cenários de teste definidos.

Tabela 5.1: Parâmetros padrão utilizados nos algoritmos ACO.

Parâmetro	Símbolo	Valor e Justificativa
<i>Parâmetros do ACO</i>		
Número de Formigas	m	10 por colônia (padrão comum na literatura).
Influência do Feromônio	α	1.0 (peso igual para o aprendizado coletivo).
Influência da Heurística	β	2.0 (leve prioridade para a ganância local).
Taxa de Evaporação	ρ	0.1 (evaporação lenta para reter memória).
<i>Pesos da Função de Qualidade</i>		
Peso da Cobertura	w_c	10.0 (alta prioridade para atender todas as paradas).
Peso do N. de Rotas	w_r	1.0 (prioridade moderada para minimizar veículos).
Peso da Distância	w_d	0.5 (menor prioridade, pois a distância já é otimizada).
<i>Parâmetros do Problema e Execução</i>		
Capacidade do Veículo	C	70 (valor arbitrário para o cenário).
Demanda por Parada	d_i	10 (demanda simulada e uniforme para controle).
Máx. de Iterações	N_{iter}	10 (condição de parada para os testes).

5.1.3 Cenários de Teste e Algoritmos de Referência

O estudo de caso utilizou um grafo do campus Darcy Ribeiro da UnB, com 27 paradas. A partir deste, foram criados dois cenários de teste com complexidades distintas (6 e 13 paradas). O desempenho do **ACO Híbrido** foi comparado com três algoritmos de referência, cada um representando um paradigma de solução distinto e relevante:

- **Força Bruta:** Algoritmo exaustivo que garante a solução ótima, servindo como a validação teórica da otimalidade em cenários de baixa complexidade.

- **Heurística Gulosa (Vizinho Mais Próximo):** Abordagem determinística e de alta velocidade. Representa uma *baseline* forte e eficiente, sendo uma solução prática e frequentemente utilizada.
- **ACO Simples:** Uma meta-heurística padrão com colônia única. Serve para isolar o ganho de desempenho da arquitetura híbrida e demonstrar as limitações de uma abordagem de otimização mais convencional.

A métrica **Cobertura**, neste contexto, é definida como a capacidade do algoritmo de gerar consistentemente soluções que atendem a 100% das paradas demandadas, respeitando todas as restrições operacionais, como a de pares opostos.

5.2 Análise Comparativa de Desempenho

Esta seção apresenta a análise quantitativa dos resultados, focando em estabelecer a eficácia da abordagem proposta em relação às baselines.

5.2.1 Análise em Cenário de Baixa Complexidade (6 Paradas)

O primeiro teste, em um cenário de escopo reduzido, permitiu uma comparação direta com a solução ótima, estabelecendo um panorama claro do desempenho relativo de cada abordagem. Os resultados estão consolidados na Tabela 5.2.

Tabela 5.2: Comparativo de desempenho para o cenário de 6 paradas.

Algoritmo	Melhor Distância (m)	Tempo Médio (s)	Cobertura
Força Bruta (Ótimo)	8371.59	2.67	100%
Heurística Gulosa	8927.00	0.015	100%
ACO Simples	—	0.077	<100%
ACO Híbrido (Proposto)	8371.59	2.1	100%

A análise revela uma hierarquia de desempenho. O **ACO Simples** falhou em gerar soluções completas, demonstrando que as restrições do problema, notadamente a de pares opostos, criam um espaço de busca "frágil" que uma meta-heurística padrão tem dificuldade em navegar. A **Heurística Gulosa**, por sua vez, provou ser uma baseline robusta, encontrando rapidamente uma solução válida, ainda que sub-ótima (6,6% pior que a ótima).

O resultado mais significativo é que o **ACO Híbrido** foi a **única abordagem heurística capaz de encontrar a solução ótima garantida**, igualando o resultado da

Força Bruta. Este fato valida a eficácia da arquitetura proposta: ela não apenas supera as falhas do ACO Simples, mas também a qualidade da solução da forte e eficiente heurística gulosa.

5.2.2 Análise em Cenário de Maior Complexidade (13 Paradas)

Ao aumentar a complexidade, a inviabilidade da Força Bruta (>60s) reforça a necessidade de heurísticas eficientes. Os resultados (Tabela 5.3) neste cenário mais realista acentuam as vantagens do ACO Híbrido.

Tabela 5.3: Comparativo de desempenho para o cenário de 13 paradas.

Algoritmo	Distância Média (m)	Tempo Médio (s)	Cobertura
Força Bruta	–	> 60s	–
Heurística Gulosa	17775.74	0.062	100%
ACO Simples	–	0.077	<100%
ACO Híbrido (Proposto)	15636.58	5.01	100%

Novamente, o ACO Simples foi incapaz de gerar soluções completas. O nosso **ACO Híbrido encontrou uma solução em média 12% melhor** (mais curta) que a Heurística Gulosa. Esta melhoria substancial justifica a maior complexidade computacional da abordagem híbrida. Ela demonstra que, à medida que a dimensionalidade do problema aumenta, a busca cooperativa entre colônias especialistas torna-se fundamental para escapar de ótimos locais e encontrar soluções de qualidade superior, algo que a abordagem gulosa, por sua natureza, não consegue fazer.

5.2.3 Análise da Dinâmica de Convergência e Qualidade da Solução

A superioridade do ACO Híbrido não é acidental; ela emerge da sua arquitetura de **especialização e cooperação**. A melhor solução para 13 paradas, visualizada na Figura 5.1, não deve ser avaliada por sua "coerência visual" subjetiva, mas sim como a representação geográfica da solução quantitativamente superior, que realiza um *trade-off* inteligente: utiliza 3 rotas para obter uma distância total significativamente menor que a solução de 2 rotas da heurística gulosa.

As figuras de convergência a seguir dissecam este processo. A Figura 5.2 mostra a rápida melhoria da função de qualidade global, indicando que o sistema converge eficientemente para boas soluções.

A Figura 5.3 ilustra a **especialização**: a colônia *ACS-Vehicles* (curva vermelha) explora consistentemente soluções com poucas rotas, cumprindo seu objetivo primário. Em

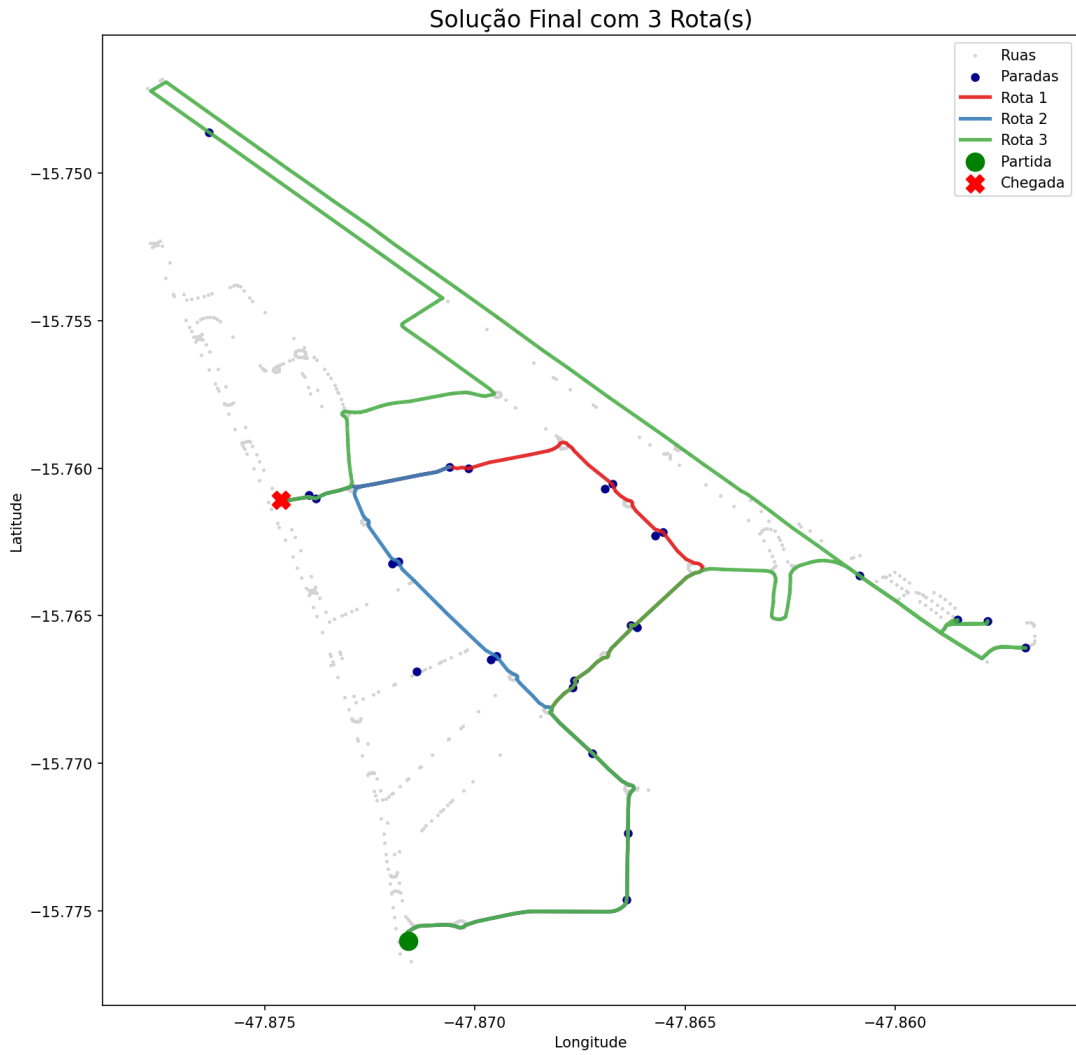


Figura 5.1: Visualização da melhor solução encontrada para 13 paradas. As diferentes cores representam as três rotas que compõem a solução de 15.636,58 metros.

contrapartida, a colônia *ACS-Time* (curva azul) foca em minimizar a distância, sem se preocupar inicialmente com o número de veículos, como mostra a Figura 5.4.

A **cooperação** é o mecanismo que une esses dois comportamentos. Através da tabela de feromônios compartilhada, a colônia *ACS-Vehicles* "aprende" os caminhos curtos descobertos pela *ACS-Time*. Isso é visível na Figura 5.4, onde a performance de distância da colônia vermelha melhora drasticamente ao longo das iterações, aproximando-se da qualidade da colônia especialista. É essa sinergia entre exploração de diferentes objetivos que permite ao sistema como um todo encontrar soluções de compromisso robustas e de alta qualidade, superando as abordagens que otimizam apenas uma faceta do problema.

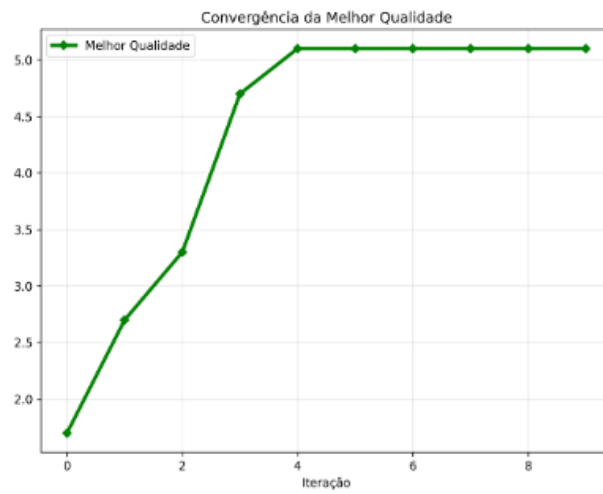


Figura 5.2: Gráfico de convergência mostrando a evolução do valor da função de qualidade da melhor solução global ao longo das iterações.

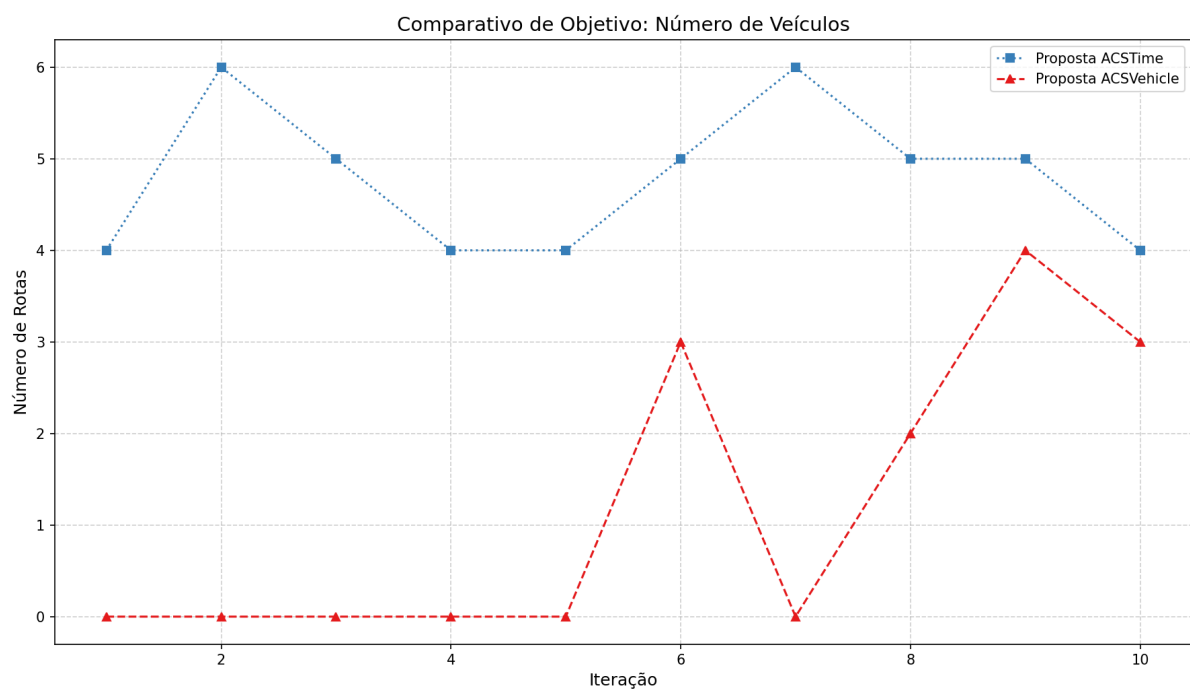


Figura 5.3: Gráfico comparativo da convergência do número de rotas (veículos) propostas por cada colônia.

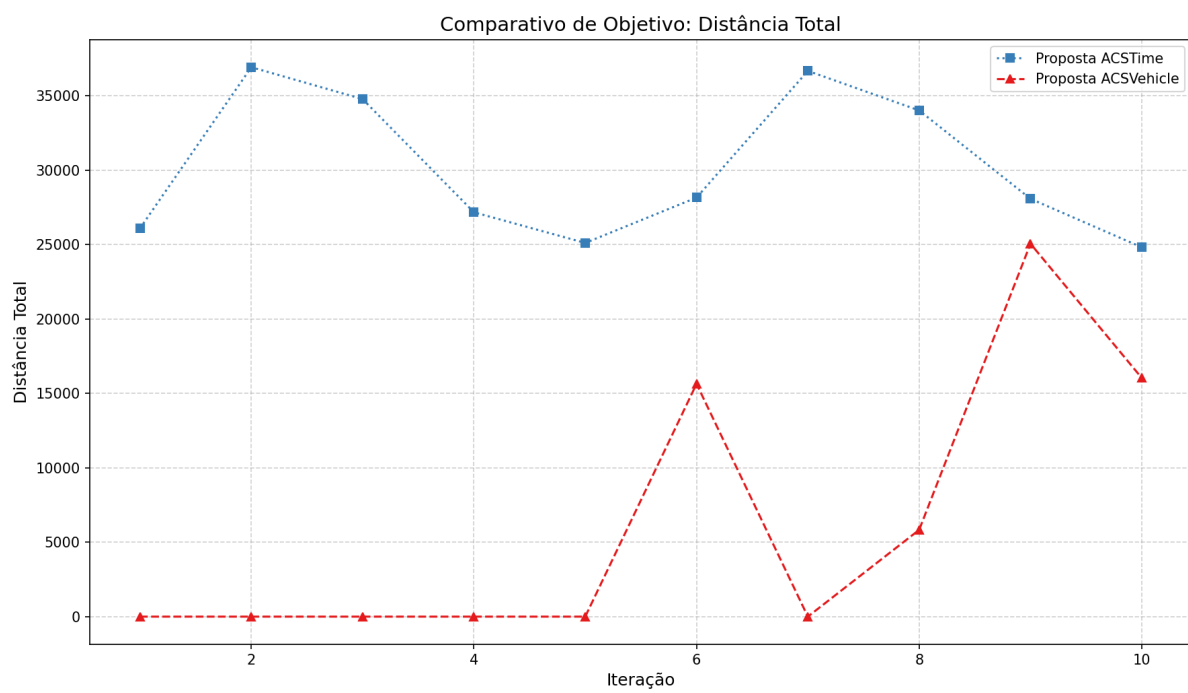


Figura 5.4: Gráfico comparativo da convergência da distância total das soluções propostas por cada colônia.

Capítulo 6

Discussão dos Resultados

Este capítulo aprofunda a análise dos resultados apresentados no capítulo anterior, interpretando seus significados no contexto dos objetivos do trabalho e da literatura da área. A discussão aborda as principais contribuições do sistema proposto, as implicações práticas dos achados e as limitações inerentes à metodologia e aos experimentos realizados.

6.1 Interpretação dos Resultados

Os resultados experimentais sugerem que a arquitetura multi-colônia, combinada com uma função de qualidade híbrida, é uma estratégia promissora para a otimização de rotas com restrições complexas. A análise comparativa apresentada nas Tabelas 5.2 e 5.3 revela um ponto fundamental: a dificuldade do algoritmo ACO Simples em gerar soluções que atendam a todas as paradas. Isso indica que as restrições do problema, especialmente a de pares opostos, criam um espaço de busca "frágil", onde abordagens de otimização que não são especificamente projetadas para lidar com essa complexidade podem não conseguir encontrar soluções viáveis.

A capacidade do ACO Híbrido de gerar soluções completas em 100% das execuções, em contrapartida, pode ser atribuída a duas frentes metodológicas. Primeiro, a construção de um **meta-grafo pré-filtrado** se mostrou crucial, pois oferece ao otimizador um espaço de busca que já respeita as restrições mais difíceis. Segundo, a **arquitetura de múltiplas colônias** demonstrou uma maior capacidade de explorar este espaço de busca complexo em comparação com a abordagem de colônia única.

A análise da dinâmica de convergência (Seção 5.2.3) elucidou o porquê. O comportamento emergente de **especialização e cooperação** parece ser a chave para o desempenho observado. A colônia ACS-Vehicles atua como um "explorador estrutural", identificando configurações de frota eficientes (com poucas rotas). A colônia ACS-Time foca em otimizar os detalhes do trajeto. O mecanismo de feromônio compartilhado permite que a

ACS-Vehicles "aprenda" com os caminhos curtos encontrados pela ACS-Time. Este resultado sugere que a diversidade de "perspectivas" introduzida pelas colônias especialistas previne a estagnação prematura e guia a busca para regiões do espaço de soluções que representam um compromisso de alta qualidade entre os objetivos.

6.2 Análise da Solução e Implicações Práticas

A melhor solução encontrada para o cenário de 13 paradas, composta por 3 rotas, alcançou uma cobertura de 64% da demanda. A cobertura parcial indica que, com os parâmetros atuais, o sistema priorizou a criação de rotas eficientes em termos de custo (distância e número de veículos), em detrimento de atender paradas mais distantes ou logisticamente complexas que exigiriam a criação de uma rota adicional.

Isso revela um *trade-off* importante para um gestor de transporte: para aumentar a cobertura, seria necessário ajustar os pesos da função de qualidade (aumentando w_c , o peso da cobertura) ou simplesmente alocar mais recursos. A solução gerada, portanto, não é apenas um resultado final, mas um **ponto de decisão informado**, que permite ao planejador entender o custo marginal de aumentar a cobertura.

A coerência logística e geográfica das rotas geradas, como pode ser observado na Figura 5.1, é um aspecto prático relevante. Rotas que são visualmente intuitivas e seguem caminhos lógicos geram mais confiança na solução e facilitam sua eventual implementação no mundo real.

6.3 Contribuições Principais do Trabalho

Com base na discussão anterior, as contribuições deste trabalho podem ser consolidadas em três áreas principais:

1. **Contribuição Metodológica:** Foi proposta e avaliada uma arquitetura de otimização que combina múltiplas colônias de formigas especialistas. A comparação empírica com uma abordagem de colônia única indicou um desempenho superior do modelo híbrido na capacidade de encontrar soluções completas para o problema com restrições complexas aqui estudado.
2. **Contribuição na Modelagem do Problema:** O trabalho apresenta um pipeline de pré-processamento que traduz um problema geográfico complexo em um modelo computacional tratável. A formalização e a implementação de filtros no meta-grafo — especialmente para tratar "pares opostos" — representam uma contribuição prática para a modelagem de VRPs com restrições operacionais do mundo real.

3. **Contribuição Empírica:** Foi realizado um estudo de caso detalhado no campus Darcy Ribeiro da UnB. Os resultados gerados, embora limitados ao escopo dos experimentos, fornecem *insights* sobre o comportamento de meta-heurísticas híbridas e sobre os *trade-offs* envolvidos no planejamento de transporte no local.

6.4 Limitações do Trabalho

Apesar dos resultados promissores, é fundamental reconhecer as limitações do presente estudo para contextualizar as conclusões e guiar pesquisas futuras.

- **Escopo dos Experimentos:** Os testes foram realizados em um número limitado de cenários (6 e 13 paradas). Para generalizar as conclusões sobre a superioridade da abordagem híbrida, seria necessário realizar uma bateria de testes mais extensa, em instâncias de diferentes tamanhos e características, e com uma análise estatística mais aprofundada.
- **Modelo de Demanda Estático e Uniforme:** O modelo atual assume uma demanda fixa, conhecida *a priori* e uniforme para todas as paradas. Em um cenário real, a demanda é dinâmica e variável, o que limita a aplicabilidade direta dos resultados para otimização em tempo real.
- **Sensibilidade a Parâmetros:** Como a maioria das meta-heurísticas, o desempenho do ACO é sensível à sua parametrização. Os valores utilizados neste trabalho foram baseados na literatura e em testes preliminares, mas uma sintonia fina sistemática dos parâmetros (e.g., usando técnicas como F-Race ou irace) poderia levar a resultados ainda melhores e não foi o foco deste estudo.
- **Dependência da Qualidade dos Dados:** A eficácia do sistema está intrinsecamente ligada à qualidade e à precisão dos dados do *OpenStreetMap*. Inconsistências ou falta de detalhe no grafo podem levar a rotas sub-ótimas. A inclusão de regras manuais mitiga, mas não elimina completamente, essa dependência.
- **Simplificação do Custo:** O cálculo de custos de aresta é baseado unicamente na distância. Fatores do mundo real como condições de tráfego, semáforos, velocidade da via ou obras não são considerados, o que impacta a precisão do tempo de viagem estimado.

Capítulo 7

Conclusão e Trabalhos Futuros

Este trabalho abordou o Problema de Roteamento de Veículos (VRP) no contexto do transporte do campus Darcy Ribeiro da Universidade de Brasília. Para solucionar este desafio, foi projetado, implementado e avaliado um sistema de otimização baseado em uma arquitetura de múltiplas colônias de formigas. A metodologia desenvolvida, centrada na construção de um meta-grafo pré-filtrado para modelar restrições operacionais complexas, como a de pares opostos, mostrou-se fundamental para a geração de soluções viáveis.

O núcleo da contribuição foi o desenvolvimento de um otimizador híbrido que utiliza duas colônias especialistas — ACS-Time e ACS-Vehicles — focadas em objetivos conflitantes (distância total vs. número de veículos). Gerenciadas por um controlador central e comunicando-se através de uma tabela de feromônios compartilhada, as colônias cooperaram para encontrar soluções de compromisso.

Os resultados experimentais indicaram o potencial da abordagem proposta. Nos testes realizados, o sistema híbrido demonstrou a capacidade de encontrar soluções que atendem a todas as restrições, em cenários onde uma abordagem de ACO de colônia única não obteve o mesmo sucesso. Além disso, em instâncias de baixa complexidade, o sistema foi capaz de encontrar a solução ótima, e em cenários maiores, superou a heurística gulosa em termos de qualidade da solução (distância total). A análise da convergência sugere que a interação entre as colônias especialistas enriquece o processo de busca, permitindo um balanço entre os objetivos conflitantes.

Conclui-se, portanto, que a arquitetura multi-colônia, combinada com uma modelagem cuidadosa do problema, representa uma estratégia promissora para a resolução de VRPs com restrições realistas e múltiplos objetivos. Os resultados deste estudo indicam que a abordagem é capaz de gerar soluções eficientes e logisticamente coerentes para os cenários testados.

7.1 Trabalhos Futuros

O sistema desenvolvido estabelece uma base sólida que pode ser expandida em diversas direções de pesquisa. As seguintes recomendações são propostas para trabalhos futuros:

- **Incorporação de Janelas de Tempo (VRPTW):** A extensão mais natural do modelo seria a inclusão de janelas de tempo para o atendimento das paradas, transformando o problema em uma variante do *Vehicle Routing Problem with Time Windows* (VRPTW), o que aumentaria significativamente o realismo do modelo.
- **Otimização com Dados Dinâmicos:** Integrar o sistema com fontes de dados dinâmicas, como o fluxo de passageiros e as condições de tráfego, permitiria a re-otimização das rotas, adaptando-se a eventos não planejados e aproximando o sistema de uma aplicação em tempo real.
- **Refinamento da Função de Custo:** A função de custo das arestas poderia ser enriquecida para incluir não apenas a distância, mas também o tempo de viagem estimado com base em dados históricos de tráfego, ou até mesmo custos de emissão de carbono para um "VRP Verde".
- **Exploração de Estratégias de Híbridização:** O controlador central poderia ser aprimorado com técnicas de busca local ou algoritmos evolucionários para hibridizar as melhores rotas encontradas por cada colônia, potencialmente gerando soluções de maior qualidade.
- **Análise de Sensibilidade e Sintonia de Parâmetros:** Realizar um estudo sistemático sobre o impacto dos parâmetros do algoritmo no seu desempenho. A utilização de técnicas de sintonia automática de parâmetros, como F-Race ou irace, poderia identificar configurações mais otimizadas para diferentes tipos de instâncias.
- **Desenvolvimento de Ferramenta de Apoio à Decisão:** Construir uma interface gráfica interativa sobre o sistema atual. Tal ferramenta permitiria que gestores de transporte pudessem facilmente configurar cenários, executar simulações e visualizar planos de rotas, transformando a pesquisa em uma ferramenta prática.

Estas sugestões representam caminhos promissores que podem aprofundar tanto a relevância teórica quanto o impacto prático do trabalho iniciado nesta monografia.

Referências

- [1] Toth, P. e D. Vigo (editores): *The Vehicle Routing Problem*. SIAM, 2001. 1
- [2] Dorigo, M. e T. Stützle: *Ant Colony Optimization*. MIT Press, 2004. 3, 6
- [3] Garey, M. R. e D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979. 4
- [4] Applegate, D. L., R. E. Bixby, V. Chvátal e W. J. Cook: *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006. 5
- [5] Braekers, K., K. Ramaekers e I. Van Nieuwenhuysse: *The vehicle routing problem: State of the art classification and review*. Computers & Industrial Engineering, 99:300–313, 2016. 5
- [6] Talbi, E. G.: *Metaheuristics: From Design to Implementation*. John Wiley & Sons, 2009. 5
- [7] Dorigo, M.: *Optimization, Learning and Natural Algorithms*. Tese de Doutorado, Politecnico di Milano, 1992. 6
- [8] Dorigo, M. e L. M. Gambardella: *Ant colony system: A cooperative learning approach to the traveling salesman problem*. IEEE Transactions on Evolutionary Computation, 1(1):53–66, 1997. 7
- [9] Kyriakakis, Nikolaos, Magdalene Marinaki e Yannis Marinakis: *A hybrid clustered ant colony optimization approach for the hierarchical multi-switch multi-echelon vehicle routing problem with service times*. Algorithms, 17(1):1–18, 2024. 8
- [10] Zhang, Tong, Zhinan Zhang, Wenzheng Zhang, Guihuan Zhang e Jing Yuan: *A multipopulation multiobjective ant colony system considering travel and prevention costs for vehicle routing in covid-19-like epidemics*. Applied Sciences, 13(3):1737, 2023. 8, 9
- [11] Yue, Y., S. Wang e J. Zhang: *A hybrid ant colony optimization algorithm for multi-compartment vehicle routing problem*. IEEE Access, 9:16601–16612, 2021. 8
- [12] Liu, Jun, Jialin Liu e Shang Gao: *Vrp problem solving based on adaptive dynamic search ant colony algorithm*. Journal of Physics: Conference Series, 1871(1):012111, 2021. 9