

Relatório Final: Simulador de Mobilidade Urbana para Controle de Tráfego e Semáforos

Autores: Lucas de Oliveira Lima e João Vitor Duarte Gonçalves

Disciplina: Estruturas de Dados

Professor(a): Ricardo Sekeff

Data: 22 de Maio de 2025

Resumo

Este relatório detalha o desenvolvimento, a análise e os resultados de um Simulador de Mobilidade Urbana projetado para investigar e comparar a eficácia de diferentes heurísticas de controle de semáforos. Utilizando uma representação da malha viária de uma seção central de Teresina-PI modelada como um grafo, o sistema simula a geração e o movimento de veículos, aplicando três estratégias distintas para a operação de semáforos: Tempos Fixos, Adaptativo por Filas e Economia de Energia. O objetivo principal é avaliar o impacto dessas heurísticas em métricas chave de desempenho, como tempo médio de viagem, tempo médio de espera, consumo de combustível por veículo e níveis de congestionamento da rede, sob diversas condições de demanda, incluindo simulações de horários de pico. São apresentados a arquitetura do sistema, as estruturas de dados implementadas, a lógica das heurísticas de controle, a metodologia de teste e calibração, e uma análise crítica dos resultados obtidos nas simulações.

1. Introdução

O crescimento urbano acelerado e o consequente aumento da frota veicular impõem desafios significativos à mobilidade nas cidades contemporâneas.

Congestionamentos, longos tempos de deslocamento e o impacto ambiental associado ao consumo excessivo de combustível são problemas prementes que afetam a qualidade de vida e a sustentabilidade urbana. Nesse contexto, sistemas de controle de tráfego inteligentes, com ênfase na otimização da operação de semáforos, emergem como ferramentas cruciais para mitigar esses problemas.

Este projeto tem como objetivo central o desenvolvimento de um simulador de mobilidade urbana que sirva como plataforma para modelar, testar e analisar o desempenho de diferentes estratégias (heurísticas) para o controle de semáforos. Através da simulação do fluxo de veículos em uma representação da rede viária de Teresina-PI, busca-se quantificar o impacto de cada heurística em indicadores de eficiência do tráfego, como a redução dos tempos de espera e viagem, a minimização do consumo de combustível e a gestão dos níveis de congestionamento.

2. Contextualização e Justificativa

A gestão eficiente do tráfego urbano é uma necessidade premente nas cidades modernas. Os semáforos, como elementos reguladores primários nos cruzamentos, desempenham um papel vital. No entanto, sistemas tradicionais baseados em temporização fixa frequentemente se mostram inadequados para lidar com a natureza dinâmica e flutuante do tráfego, resultando em atrasos desnecessários e subutilização da capacidade viária.

A simulação computacional oferece um ambiente controlado e de baixo custo para experimentar e validar novas abordagens de controle de tráfego antes de uma custosa e complexa implementação no mundo real. Ao modelar a cidade como um grafo e simular o comportamento de veículos e semáforos, é possível investigar o impacto de diferentes lógicas de controle. Este projeto justifica-se pela necessidade de explorar estratégias mais inteligentes e adaptativas que possam responder dinamicamente às condições de tráfego, visando melhorar a fluidez, reduzir o tempo perdido em congestionamentos e promover um uso mais eficiente dos recursos energéticos.

3. Modelagem do Sistema

O simulador foi construído sobre um modelo que abstrai os componentes chave do ambiente de tráfego urbano.

3.1. Representação da Rede Urbana como Grafo

A infraestrutura viária da área central de Teresina-PI foi modelada como um grafo direcionado $G=(V,E)$, onde:

- **Nós (Vértices - Node.java):** Representam as interseções (cruzamentos) da cidade. Cada nó é caracterizado por um ID único, coordenadas geográficas (latitude e longitude) e um atributo booleano indicando a presença de um semáforo.
- **Arestas (Arcos - Edge.java):** Representam os segmentos de via que conectam duas interseções. Cada aresta possui atributos como ID, nó de origem, nó de destino, comprimento físico (em metros), velocidade máxima permitida (km/h), tempo de travessia nominal (calculado a partir do comprimento e velocidade máxima), e um indicador de mão única.
- **Carregamento do Grafo (JsonParser.java, Graph.java):** Os dados da rede (nós e arestas), incluindo a localização de semáforos, foram carregados a partir de um arquivo JSON (CentroTeresinaPiauiBrazil.json). A classe Graph armazena e

gerencia a estrutura do grafo.

3.2. Modelagem dos Veículos (Vehicle.java)

Os veículos são as entidades dinâmicas que se deslocam pela rede.

- **Atributos:** Cada veículo possui um ID único, nó de origem e destino, uma rota (sequência de nós a percorrer), sua posição atual em uma aresta, tempo de viagem acumulado, tempo de espera acumulado e consumo de combustível.
- **Geração (VehicleGenerator.java):** Veículos são introduzidos na simulação em nós aleatórios da rede, com destinos também aleatórios, a uma taxa configurável (vehicleGenerationRate).
- **Roteamento (Dijkstra.java):** No momento da geração, a rota mais curta entre a origem e o destino do veículo é calculada utilizando o algoritmo de Dijkstra, considerando o tempo de travessia nominal das arestas.
- **Movimentação (Simulator.java):** A cada passo de tempo da simulação (deltaTime), os veículos avançam em suas rotas. Seu movimento é influenciado pela velocidade da via, pela presença de outros veículos (implicitamente, através do congestionamento e filas) e, crucialmente, pelo estado dos semáforos nos cruzamentos.

3.3. Modelagem dos Semáforos (TrafficLight.java)

Os semáforos são componentes essenciais para o controle do fluxo nos cruzamentos e são modelados como Máquinas de Estados Finitos (MEF).

- **Estados Finitos (LightPhase.java):** Os estados representam as diferentes configurações de luzes permitidas para um cruzamento (ex: NS_GREEN_EW_RED - Norte-Sul verde, Leste-Oeste vermelho; NS_YELLOW_EW_RED, etc.). O estado atual é armazenado em TrafficLight.currentPhase.
- **Transições de Estado:**
 - **Evento:** A principal condição para uma transição é a expiração do temporizador da fase atual (TrafficLight.phaseTimer).
 - **Lógica:** Quando o timer expira, o método decideNextPhase() da TrafficLightControlStrategy ativa é invocado. Esta estratégia analisa o estado atual, as filas de veículos e outras condições (como horário de pico) para determinar o próximo estado (fase) e a duração dessa nova fase.
 - **Ação:** O TrafficLight atualiza seu currentPhase e reinicia o phaseTimer com a duração definida pela estratégia.
- **Filas (Queue.java):** Cada semáforo gerencia filas de veículos para as diferentes direções de aproximação, utilizando uma implementação de fila baseada na CustomLinkedList.

4. Estruturas de Dados Implementadas

A escolha e implementação de estruturas de dados adequadas foram fundamentais para o desenvolvimento do simulador:

- **Grafo (Graph.java):** A estrutura central, utilizando HashMap para acesso eficiente aos nós por ID e CustomLinkedList para armazenar as coleções de nós, arestas e semáforos. Cada Node também utiliza uma CustomLinkedList para suas arestas de saída.
- **Lista Encadeada Customizada (CustomLinkedList.java):** Uma implementação própria de lista duplamente encadeada, fornecendo as operações básicas necessárias para gerenciar coleções dinâmicas como rotas de veículos, histórico de congestionamento e a base para a estrutura de Fila.
- **Fila (Queue.java):** Implementada sobre a CustomLinkedList, utilizada para gerenciar os veículos em espera nos semáforos, seguindo o princípio FIFO (First-In, First-Out).
- **HashMap<K, V> (Java Collections):** Utilizada para mapeamentos que exigem acesso rápido, como o mapeamento de IDs de nós para os objetos Node correspondentes no grafo e o mapeamento de nomes de direção para índices de filas dentro da classe TrafficLight.

5. Heurísticas de Controle de Semáforos

Três distintas estratégias de controle de semáforos foram implementadas, cada uma com uma abordagem diferente para otimizar o fluxo de tráfego. Todas implementam a interface TrafficLightControlStrategy.

5.1. Heurística 1: Tempos Fixos (FixedTimeStrategy.java)

- **Descrição:** Esta estratégia opera com durações pré-definidas e constantes para as fases verde e amarela dos semáforos. Uma lógica simples foi implementada para diferenciar os tempos de verde para horários de pico e fora de pico (ou, em algumas configurações de teste, usar o mesmo tempo fixo independentemente do pico para criar um benchmark menos otimizado).
- **Parâmetros Configuráveis (via Configuration.java):** fixedGreenTime, fixedYellowTime.
- **Objetivo:** Servir como um baseline para comparação, representando um sistema de controle de semáforos tradicional.

5.2. Heurística 2: Adaptativo por Fila (AdaptiveQueueStrategy.java)

- **Descrição:** Esta estratégia ajusta dinamicamente a duração da fase verde com base no número de veículos detectados nas filas de aproximação. O objetivo é

alocar mais tempo verde para as direções com maior demanda, visando reduzir os tempos de espera e otimizar a vazão, com uma lógica que tenta "limpar a fila" da direção que recebe o verde.

- **Parâmetros Configuráveis (via Configuration.java):** adaptiveBaseGreen, adaptiveYellowTime, adaptiveMinGreenTime, adaptiveMaxGreen, adaptiveQueueThreshold, adaptiveIncrement, adaptivePeakHourBonus.
- **Objetivo:** Superar a estratégia de tempos fixos, especialmente em horários de pico, adaptando-se dinamicamente às condições de tráfego para minimizar o tempo de espera e melhorar o fluxo.

5.3. Heurística 3: Economia de Energia (EnergySavingStrategy.java)

- **Descrição:** Esta estratégia foca na minimização do consumo de combustível. A abordagem principal é tentar reduzir o número de paradas e partidas, o que pode envolver a concessão de fases verdes mais longas quando há demanda e mais curtas quando a demanda é baixa, especialmente fora do horário de pico.
- **Parâmetros Configuráveis (via Configuration.java):** energySavingBaseGreen, energySavingYellowTime, energySavingMinGreen, energySavingMaxGreenTime, energySavingThreshold.
- **Objetivo:** Alcançar o menor consumo médio de combustível por veículo, particularmente em cenários de tráfego leve a moderado (ex: madrugada).

6. Metodologia de Simulação e Calibração

A avaliação das heurísticas foi conduzida através de uma série de simulações com parâmetros controlados.

- **Parâmetros Gerais da Simulação (Configuration.java):**
 - vehicleGenerationRate: Controla a intensidade da demanda de tráfego.
 - simulationDuration: Define a duração total de cada simulação.
 - vehicleGenerationStopTime: Ponto no tempo em que novos veículos deixam de ser introduzidos.
 - peakHour: Ativa/desativa lógicas específicas para horários de pico.
- **Cenários de Teste:** Foram definidos cenários variando a vehicleGenerationRate e o estado de peakHour.
- **Processo de Calibração:**
 - A Heurística 1 (Fixa) foi configurada com diferentes tempos fixos para servir de benchmark.
 - A Heurística 2 (Adaptativa) passou por um extenso processo iterativo de ajuste de seus múltiplos parâmetros.
 - A Heurística 3 (Economia) foi testada em cenários de baixa demanda.
 - A saturação da rede (picos de congestionamento de 100%) foi um fator

observado, levando à necessidade de testar com taxas de geração variadas.

7. Resultados e Análise

A seguir, são apresentados os resultados comparativos das simulações para as diferentes heurísticas de controle de semáforos.

Cenário de Teste Principal: vehicleGenerationRate = 0.6 veículos/segundo, peakHour = true

Métrica	Heurística 1 (Fixo)	Heurística 2 (Adaptativo)
Tempo Total de Simulação (s)	1884,00	2097,00
Veículos Gerados	722	725
Veículos Chegados	722	725
Tempo Médio Viagem (s)	119,57	140,40
Tempo Médio Espera (s)	11,00	29,97
Consumo Total de Combustível (L)	37,080	40,607
Consumo Médio / Veículo (L)	0,051	0,056
Pico de Congestionamento (%)	100,00%	100,00%
Média de Congestionamento Registrado (%)	88,81%	83,61%

(Nota: Os resultados da Heurística 1 correspondem à sua melhor execução reportada ["H1 - Execução A"]. Os resultados da Heurística 2 correspondem à execução "H2 - Execução Y", que foi a melhor das duas últimas rodadas da H2 fornecidas para este cenário.)

Cenário de Teste Baixa Demanda: vehicleGenerationRate = 0.2 veículos/segundo, peakHour = false

Métrica	Heurística 3 (Economia)
---------	-------------------------

Tempo Total de Simulação (s)	1964,00
Veículos Gerados	291
Veículos Chegados	291
Tempo Médio Viagem (s)	139,15
Tempo Médio Espera (s)	24,93
Consumo Médio / Veículo (L)	0,057
Pico de Congestionamento (%)	100,00%
Média de Congestionamento Registrado (%)	88,47%

Análise Crítica dos Resultados:

- **Comparação H1 vs. H2 (Cenário de Pico com genRate = 0.6):**
 - No cenário de alta demanda, a Heurística 1 (Fixa), em sua melhor execução, apresentou um desempenho superior à Heurística 2 (Adaptativa) com os parâmetros testados.
 - **Tempo de Viagem e Espera:** A H1 alcançou um tempo médio de viagem (119,57s) e, mais notavelmente, um tempo médio de espera (11,00s) significativamente inferiores aos da H2 (140,40s e 29,97s, respectivamente). Isso indica que, para esta configuração específica e nível de carga, a estratégia adaptativa não conseguiu otimizar o fluxo de forma a reduzir os atrasos individuais dos veículos de maneira eficaz em comparação com um sistema fixo bem ajustado (ou que teve uma execução estocástica favorável).
 - **Consumo de Combustível:** O consumo médio por veículo também foi menor para a H1 (0,051 L vs 0,056 L).
 - **Vazão:** Ambas as heurísticas processaram um número similar de veículos, com a H1 tendo uma ligeira vantagem no tempo total para processar seus veículos.
 - **Congestionamento:** A Heurística 2 apresentou uma **média de congestionamento registrado inferior (83,61% vs 88,81%)**. Este é um ponto interessante, sugerindo que, embora a experiência individual do veículo (tempo de espera/viagem) tenha sido pior na H2, o estado geral da rede, em termos de número de veículos parados simultaneamente ou densidade, pode ter sido ligeiramente melhor em média. Isso pode ocorrer se a H2, ao tentar se adaptar, criar fases verdes muito longas para algumas direções, limpando bem essas filas momentaneamente (o que baixa o índice de

congestionamento instantâneo), mas às custas de tempos de espera muito mais longos para as direções transversais.

- **Análise da Heurística 3 (Cenário de Baixa Demanda - $\text{genRate} = 0.2$, $\text{peakHour} = \text{false}$):**
 - **Consumo de Combustível:** Com um consumo médio de 0,057 L/veículo, a Heurística 3 não demonstrou uma vantagem clara em economia de combustível em relação aos resultados da H1 (0,051 L - 0,054 L) ou H2 (0,056 L) nos seus cenários de pico. Para uma heurística focada em economia e em baixa demanda, esperava-se um resultado melhor.
 - **Tempo de Viagem e Espera:** Os tempos de viagem (139,15s) e espera (24,93s) para H3 foram consideravelmente altos para um cenário de baixa demanda.
 - **Pico de Congestionamento:** Atingir 100% de pico de congestionamento com uma taxa de geração de apenas 0.2 e sem horário de pico é um forte indicativo de que os parâmetros da Heurística 3 (possivelmente `energySavingMaxGreenTime` ou `energySavingBaseGreen` muito longos) estão causando bloqueios severos e gerenciando mal o fluxo. A média de congestionamento de 88,47% também é excessivamente alta para este cenário.
 - **Conclusão sobre H3:** A configuração atual da Heurística 3 precisa de uma revisão e calibração significativas, pois não está atingindo seus objetivos e parece induzir congestionamento desnecessário.
- **Impacto da Saturação da Rede:**
 - Os picos de 100% de congestionamento em todos os cenários de alta demanda (e até mesmo no de baixa demanda para H3) indicam que a capacidade da rede é um fator limitante ou que as estratégias de semáforo, como configuradas, não estão prevenindo a saturação em momentos críticos. Isso dificulta a avaliação precisa dos benefícios incrementais das heurísticas adaptativas quando o sistema está operando constantemente no limite.

(Adicione gráficos comparando as heurísticas para as principais métricas. Gráficos de barras são eficazes aqui.)

8. Desafios Encontrados e Soluções Adotadas

Durante o desenvolvimento e teste do simulador, diversos desafios foram encontrados:

- **Calibração da Heurística Adaptativa (H2):** O principal desafio foi encontrar a combinação de parâmetros que permitisse à Heurística 2 superar consistentemente a Heurística 1 em cenários de alta demanda. Isso exigiu um

processo iterativo de múltiplos testes, análise detalhada dos resultados e ajustes finos. Os resultados mostraram grande sensibilidade aos parâmetros e à carga da rede.

- **Desempenho da Heurística de Economia de Energia (H3):** Esta heurística não atingiu o objetivo de menor consumo de combustível de forma clara e, em alguns cenários de baixa demanda, induziu alto congestionamento, indicando necessidade de revisão em sua lógica.
- **Saturação da Rede:** A ocorrência frequente de picos de 100% de congestionamento tornou a diferenciação do desempenho das heurísticas mais sutil e complexa.
- **Definição e Interpretação de Métricas:** A métrica de "Média de Congestionamento" por vezes apresentou tendências diferentes das métricas de experiência do veículo.
- **Variabilidade Estocástica:** Observou-se variabilidade nos resultados de execuções repetidas com os mesmos parâmetros, devido à natureza aleatória da geração de veículos.

9. Conclusão

O desenvolvimento do Simulador de Mobilidade Urbana proporcionou uma plataforma valiosa para a modelagem e análise comparativa de diferentes estratégias de controle de semáforos. As simulações conduzidas na rede viária de Teresina-PI levaram às seguintes conclusões principais:

1. **Heurísticas de Tempos Fixos (H1):** Podem apresentar um desempenho surpreendentemente bom em cenários de alta demanda se os tempos fixos estiverem bem ajustados (mesmo que por acaso). No entanto, sua natureza estática limita sua eficiência em condições de tráfego variáveis.
2. **Heurística Adaptativa por Fila (H2):** A calibração desta heurística para cenários de alta demanda provou-se desafiadora. Embora tenha demonstrado potencial para reduzir a média de congestionamento da rede, os parâmetros testados mais recentemente não resultaram em uma melhoria consistente nos tempos de viagem e espera individuais dos veículos em comparação com a melhor execução da Heurística 1. Isso sugere que, sob forte saturação, as decisões locais da estratégia adaptativa podem não se traduzir em otimizações globais da experiência do usuário, ou que os parâmetros ótimos ainda não foram encontrados.
3. **Heurística de Economia de Energia (H3):** Na sua configuração atual, não atingiu os objetivos de eficiência energética e apresentou problemas de fluxo mesmo sob baixa demanda, indicando a necessidade de uma revisão substancial

em sua lógica e parâmetros.

4. **Impacto da Saturação:** A capacidade da rede viária é um fator limitante crucial. Em cenários onde a demanda excede consistentemente a capacidade, mesmo estratégias de controle otimizadas podem ter dificuldade em evitar picos de congestionamento.

O projeto cumpriu o objetivo de implementar e comparar diferentes heurísticas, fornecendo insights valiosos sobre a complexidade do controle de tráfego e a importância crítica da calibração de parâmetros e da consideração das condições de carga da rede.

Trabalhos Futuros:

As possibilidades de extensão deste trabalho incluem:

- **Calibração Robusta da Heurística Adaptativa (H2):** Explorar uma gama mais ampla de parâmetros e testar em cenários com níveis de saturação variados para identificar configurações que superem consistentemente a Heurística 1.
- **Refinamento da Heurística de Economia de Energia (H3):** Investigar lógicas mais sofisticadas.
- **Implementação de Algoritmos de Coordenação de Semáforos.**
- **Validação com Dados Reais.**
- **Expansão do Modelo.**