

Relatório da Simulação de Tráfego Urbano no SemaFlux

José Roberto de Sousa Brito Júnior
Icev
E-mail: jose_roberto.junior@somosicev.com

Warney Rego Ferreira Filho
Icev
E-mail: warney.filho@somosicev.com

Abstract—Este artigo apresenta os detalhes de modelagem e implementação do sistema de simulação de tráfego urbano SemaFlux. Desenvolvido em Java com JavaFX, o SemaFlux é projetado para simular e analisar o fluxo de tráfego e estratégias de controle de semáforos em ambientes urbanos. O sistema modela a rede viária como um grafo, utiliza estruturas de dados personalizadas para gerenciar elementos da simulação e implementa múltiplos algoritmos para controle de semáforos. Este relatório descreve a abordagem de modelagem do sistema, as estruturas de dados empregadas e os algoritmos utilizados para o controle de semáforos.

Index Terms—Simulação de tráfego, Teoria dos grafos, Estruturas de dados, Algoritmos, Controle de semáforos

I. INTRODUÇÃO

O congestionamento de tráfego é um desafio significativo em áreas urbanas, resultando em aumento do tempo de viagem, maior consumo de combustível e poluição ambiental. A gestão eficaz do tráfego, especialmente por meio de controles otimizados de semáforos, é essencial para mitigar esses problemas. O projeto SemaFlux, disponível em [invalidurl,donotcite], oferece uma plataforma de simulação que permite testar e avaliar várias estratégias de controle de semáforos em um ambiente virtual, auxiliando planejadores urbanos e engenheiros de tráfego na otimização do fluxo viário.

Desenvolvido em Java com a biblioteca JavaFX, o SemaFlux proporciona um ambiente de simulação em tempo real, onde os usuários podem visualizar movimentos de veículos e experimentar diferentes modos de controle para alcançar objetivos como redução de congestionamentos e minimização do consumo de energia.

II. MODELAGEM DO SISTEMA

No SemaFlux, a rede viária urbana é modelada como um grafo direcionado ou não direcionado, dependendo da propriedade de sentido único das arestas. Os cruzamentos são representados como nós (No), caracterizados por um identificador único, latitude e longitude. As ruas são representadas como arestas (Aresta), com atributos como comprimento (em metros), tempo de viagem (em segundos), status de sentido único, velocidade máxima (em km/h) e capacidade (número de veículos suportados). Semáforos (SinalTransito) são associados a nós específicos para regular o fluxo de veículos nos cruzamentos.

Os veículos (Veiculo) são gerados com origens e destinos aleatórios, e suas rotas são calculadas utilizando o algoritmo

de Dijkstra para encontrar o caminho mais curto através do grafo [2]. Cada veículo possui propriedades como posição atual, tempo de viagem, tempo de espera e consumo de combustível, permitindo uma simulação realista do comportamento do tráfego.

III. ESTRUTURAS DE DADOS

As estruturas de dados no SemaFlux são projetadas para gerenciar eficientemente os componentes da simulação. As principais estruturas incluem:

A. Grafo

A classe Grafo encapsula a rede viária, mantendo três coleções principais:

- `nodesList`: Uma lista ligada personalizada (`ListaLigada`) que armazena todos os nós do grafo.
- `edgesList`: Uma lista ligada que armazena todas as arestas.
- `trafficLightsList`: Uma lista ligada que armazena todos os semáforos.

Além disso, um `HashMap` (`nodeMap`) é utilizado para recuperação rápida de nós por seus identificadores, oferecendo complexidade de tempo média $O(1)$ para buscas [3].

B. No

A classe No representa um cruzamento ou ponto na rede viária, com os seguintes atributos:

- `id`: Identificador único do nó.
- `latitude` e `longitude`: Coordenadas geográficas.
- `isTrafficLight`: Booleano indicando se o nó possui um semáforo.
- `edges`: Lista ligada de arestas conectadas ao nó.

C. Aresta

A classe Aresta representa uma rua conectando dois nós, com atributos como:

- `id`: Identificador único da aresta.
- `source` e `target`: IDs dos nós de origem e destino.
- `length`: Comprimento da rua em metros.
- `travelTime`: Tempo de travessia em segundos.
- `oneway`: Booleano indicando se a rua é de sentido único.
- `maxspeed`: Velocidade máxima em km/h.
- `capacity`: Capacidade da rua em número de veículos.

D. Fila

A classe *Fila* é uma implementação de lista ligada simples usada para gerenciar filas de veículos aguardando em semáforos. Cada instância de *Fila* mantém:

- *size*: Número de veículos na fila.
- *front* e *back*: Ponteiros para o primeiro e último veículo.

A classe *SinalTransito* contém um array de quatro objetos *Fila*, correspondendo às direções norte, leste, sul e oeste, permitindo o gerenciamento de filas por direção. Um *HashMap* (*directionNameToIndexMap*) mapeia nomes de direções para índices do array, facilitando o acesso rápido.

TABLE I
ESTRUTURAS DE DADOS NO SEMAFLUX

Estrutura	Descrição
Grafo	Armazena nós, arestas e semáforos usando listas ligadas e um <i>HashMap</i> para busca rápida.
No	Representa cruzamentos com coordenadas e lista de arestas conectadas.
Aresta	Representa ruas com atributos como comprimento e capacidade.
Fila	Lista ligada para gerenciar veículos aguardando em semáforos por direção.

IV. ALGORITMOS

O SemaFlux implementa três algoritmos distintos para controle de semáforos, cada um implementado em uma classe que adere à interface *Semaforo*. Esses algoritmos gerenciam as transições entre fases (verde, amarelo, vermelho) e ajustam os tempos com base nas condições de tráfego ou objetivos de eficiência energética.

A. Tempo Fixo (*TempoFixo*)

O modo de tempo fixo utiliza ciclos constantes para as fases verde, amarelo e vermelho, independentemente do volume de tráfego. Os parâmetros principais incluem:

- *strategyGreenDuration*: 15 segundos (20 em horário de pico).
- *strategyYellowDuration*: 3 segundos.
- *strategyRedDuration*: 18 segundos.

O algoritmo segue uma sequência fixa de quatro fases para cruzamentos de quatro vias, alternando entre direções Norte-Sul e Leste-Oeste. É ideal para condições de tráfego estáveis, mas não se adapta a mudanças dinâmicas.

B. Adaptativo (*FilaAdaptativa*)

O modo adaptativo ajusta dinamicamente a duração da fase verde com base no tamanho das filas de veículos em cada direção. Os parâmetros incluem:

- *TempoVerdeBase*: 10 segundos (15 em horário de pico).
- *TempoAmarelo*: 3 segundos.
- *TempoMaximoVerde*: 30 segundos.
- *tempoTransicao*: 3 veículos (limiar para extensão do tempo verde).

- *tempoExtraPorVeiculo*: 1 segundo por veículo acima do limiar.

O algoritmo calcula o tempo verde começando com a base e adicionando tempo extra para cada veículo além do limiar, garantindo que o tempo permaneça entre os limites mínimo e máximo. Isso reduz congestionamentos em cruzamentos com alto tráfego.

C. Economia de Energia (*EconomiaEnergia*)

O modo de economia de energia é otimizado para períodos de baixo tráfego, minimizando o tempo de verde para reduzir o consumo de energia. Os parâmetros incluem:

- *tempoVerdeBase*: 20 segundos (22 em horário de pico).
- *tempoAmarelo*: 3 segundos.
- *tempoVerdeMinimo*: 7 segundos.
- *limiteTrafegoBaixo*: 1 veículo.

Se o número de veículos nas filas for igual ou inferior ao limiar e não for horário de pico, o tempo verde é reduzido ao mínimo, mantendo o fluxo de tráfego dentro de limites aceitáveis.

TABLE II
PARÂMETROS DOS ALGORITMOS DE CONTROLE DE SEMÁFOROS

Parâmetro	Tempo Fixo	Adaptativo	Economia gia
Tempo Verde Base	15 s (20 s pico)	10 s (15 s pico)	20 s (22 s pico)
Tempo Amarelo	3 s	3 s	3 s
Tempo Verde Máximo	-	30 s	40 s
Limiar de Tráfego	-	3 veículos	1 veículo

V. CONCLUSÃO

O sistema de simulação de tráfego SemaFlux oferece uma plataforma robusta para modelagem de dinâmicas de tráfego urbano e avaliação de estratégias de controle de semáforos. Por meio de sua representação baseada em grafos da rede viária, estruturas de dados eficientes e algoritmos flexíveis, o SemaFlux permite análises detalhadas e otimização de técnicas de gerenciamento de tráfego para diversos cenários urbanos. A capacidade de alternar entre modos de controle fixo, adaptativo e de economia de energia torna o sistema uma ferramenta valiosa para planejadores urbanos e engenheiros de tráfego.

REFERENCES

- [1] W. Rego, "SemaFlux," GitHub Repository, [invalidurl,donotcite] 2023.
- [2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms," 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
- [3] "Hash table," Wikipedia, The Free Encyclopedia. [Online]. Available: [invalidurl,donotcite] [Accessed: May 23, 2025].