



**Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Curso de Engenharia de Software**

**Sistemas Multiagentes**

**Paradigmas de Programação  
Professora: Milene Serrano**

**Autores:  
Jackes Tiago Ferreira da Fonseca - 190030291**

**Brasília, DF**



# Sumário

<b>1. Introdução e Fundamentação Teórica</b>	<b>3</b>
Framework MESA	3
<b>2. Modelagem e implementação</b>	<b>4</b>
Descrição dos Componentes do Sistema	4
Ambiente	4
Agentes	4
Parâmetros da Simulação	4
Arquitetura Multiagente	5
Diagrama de Classes ou Diagramas de Interação	5
<b>3. Comparação com trabalhos relacionados</b>	<b>5</b>
Outros Frameworks para Simulação de Tráfego	5
Soluções Multiagentes em Diferentes Contextos	6
<b>4. Vantagens e limitações</b>	<b>6</b>
Pontos Fortes	6
Desafios e Limitações	7
<b>5. Discussões sobre aplicabilidade real</b>	<b>7</b>
Validação dos Resultados	7
Extensões Potenciais	8
Impacto na Tomada de Decisão	8
<b>6. Conclusão e trabalhos futuros</b>	<b>8</b>
Trabalhos Futuros	9

# 1. Introdução e Fundamentação Teórica

Os sistemas multiagente consistem em um conjunto de agentes autônomos que interagem entre si e com o ambiente para resolver problemas complexos descentralizados. Em engenharia de software, um agente é uma entidade capaz de perceber seu ambiente, tomar decisões autônomas e executar ações para atingir seus objetivos. O sistema multiagente é definido por três características principais:

- **Autonomia:** Cada agente é autônomo, ou seja, tem controle sobre as ações e o estado do sistema. O agente é responsável por suas próprias tarefas específicas sem a necessidade de um controlador central.
- **Interação:** os agentes interagem diretamente uns com os outros ou indiretamente através do ambiente. A cooperação ou competição entre os agentes é essencial, pois as ações de um agente afetam o comportamento dos outros e do sistema como um todo.
- **Adaptação:** os agentes devem ser capazes de alterar seu comportamento para o ambiente ou para as interações no ambiente. Desta forma, os agentes podem se adaptar às condições dinâmicas do sistema

## Comparação com Modelos de Simulação Tradicional

A abordagem multiagente apresenta algumas vantagens em relação à abordagem centralizada em modelos baseados em equações diferenciais:

**Limitações dos modelos centralizados:** eles dependem de um controlador central que toma decisões e administra o sistema. Como resultado, pode ser um gargalo computacional e uma impossibilidade de lidar com sistemas dinâmicos e imprevisíveis. Vantagens de MAS: eliminam um único ponto de falha através de sistemas distribuídos e inteligência descentralizada.. Limitações dos modelos baseados em equações diferenciais: sem modelagem do comportamento de cada veículo, simplificando os desvios da estrada e restringindo a possibilidade de modelo de interação social emergente.

**Vantagens do MAS:** detém os agentes com suas próprias regras e possibilidade de trocar e/ou atualizar informação. `ArgumentOutOfRangeException`., MAS facilmente adaptado para diferentes peças e/ou novas peças.

---

## Framework MESA

O MESA é um framework Python que diz respeito a agentes modelares padrões e sistemas de simulação em ambientes discretos. Foi concebido para ser um projeto simples de usar e customizar, e foi desenvolvido para ajudar a construir rapidamente simulações acadêmicas e profissionais. As razões pelas quais escolhi para o apresentado projeto de trabalho incluem:

**Usabilidade:** A API da MESA é extremamente simples e fácil de usar e, dada a qualidade da documentação fornecida, deve permitir que os desenvolvedores construam um modelo em um tempo muito curto. É facilmente integrado em outras bibliotecas Python populares, como Pandas e NumPy, o que é muito bom para a análise de dados e visualização.

**Flexibilidade:** Permite a criação de ambientes altamente customizáveis, como redes de ruas e interseções. Dá suporte à definição de um algoritmo de comportamento específico para cada tipo de agente, por exemplo, veículo ou semáforo.

**Real-time View:** O software também inclui ferramentas de visualização ao vivo para interagir com as simulações, o que permite monitorar o comportamento dos agentes e validar o modelo de forma rápida e eficiente.

**Aplicabilidade a ambientes discretos:** Os ambientes discretos, como o tráfego da cidade, onde um ambiente pode ser representado por um grid de células, como ruas e intersecções, e agentes interagem localmente, são bem representados pelo MESA.

Todas essas razões tornam o MESA adequado para simulações de sistemas multiagentes baseadas em tráfego de cidade, o que me permitirá construir um modelo escalável e acessível. Além disso, ele é modulado, então acho que posso incluir novos agentes, como pedestres ou ciclistas e implementar políticas de tráfego mais sofisticadas.

## 2. Modelagem e implementação

### Descrição dos Componentes do Sistema

#### Ambiente

O ambiente é a representação física e funcional do tráfego urbano, estruturado como uma **grade discreta** de células, onde cada célula pode representar uma rua, um cruzamento ou um semáforo. Ele é responsável por definir as regras e as interações dos agentes. Os principais componentes do ambiente são:

- **Ruas:** Representadas como caminhos unidimensionais na grade. Elas contêm limites de velocidade e direções de fluxo que os agentes devem respeitar.
- **Semáforos:** Localizados em cruzamentos ou intersecções, controlam o fluxo de veículos. Cada semáforo possui estados (vermelho, verde, amarelo) e opera com base em ciclos predefinidos ou adaptativos.
- **Cruzamentos:** Áreas onde múltiplas ruas se encontram. São pontos críticos que exigem coordenação entre os agentes para evitar colisões e congestionamentos.

#### Agentes

Os agentes representam os veículos no sistema e são projetados para exibir **comportamentos autônomos**. Cada agente é modelado com as seguintes características:

- **Estado interno:** Inclui sua posição, velocidade e destino.
- **Regras de comportamento:**
  - **Movimentação:** Cada agente decide sua direção com base nas regras do ambiente, como seguir o fluxo da via e respeitar limites de velocidade.
  - **Respeito às Regras de Trânsito:** Agentes param em sinais vermelhos e dão preferência em cruzamentos.
  - **Evitação de Colisões:** Detectam outros agentes nas proximidades e ajustam sua velocidade ou direção para evitar colisões.
- **Tomada de Decisão:** Cada agente toma decisões com base nas informações disponíveis no ambiente, sem depender de um controle central.

---

### Parâmetros da Simulação

A simulação é parametrizada com os seguintes elementos:

- **Tamanho do mapa:** Definido no modelo como uma grade de tamanho ajustável (mínimo de 11x11 e máximo de 30x30 células).
- **Número de agentes:** Veículos e semáforos são instanciados de acordo com a configuração inicial.
- **Frequência de geração de veículos:** Carros são posicionados em pontos iniciais, permitindo a adição de novos veículos ao longo da simulação.

- **Regras de trânsito:** Incluem a sincronização dos semáforos e a permissão de conversões em cruzamentos específicos.
- 

## Arquitetura Multiagente

O modelo utiliza uma arquitetura descentralizada para gerenciar a simulação. Cada agente (veículo ou semáforo) opera de forma autônoma, realizando decisões baseadas em suas percepções locais. A descentralização é gerenciada pelo framework MESA, que coordena a execução dos agentes e a interação deles com o ambiente (representado como uma grade discreta).

- **Semáforos:** Controlam o fluxo de veículos em cruzamentos, alternando dinamicamente entre os estados para prevenir colisões.
- **Veículos:** Decidem suas ações (mover-se ou virar) com base no conteúdo das células adjacentes e nas permissões do ambiente.

Essa abordagem facilita a escalabilidade e promove um comportamento emergente, essencial para modelar sistemas complexos como o tráfego urbano.

---

## Diagrama de Classes ou Diagramas de Interação

Um **diagrama de classes** para o sistema pode ser descrito com as seguintes entidades principais:

1. **TrafficCell**
  - Atributos: `cell_type`, `allowed_turn`
  - Métodos: `step`
2. **TrafficLightAgent**
  - Atributos: `state`, `timer`
  - Métodos: `step`
3. **CarAgent**
  - Atributos: `direction`, `path`
  - Métodos: `go_straight`, `turn`, `step`
4. **TrafficModel**
  - Atributos: `grid`, `datacollector`
  - Métodos: `step`

O **diagrama de interação** pode ilustrar as interações entre agentes:

- **Semáforos** alteram seu estado com base no tempo e impactam a decisão dos **carros** ao atravessar cruzamentos.
- **Carros** consultam o tipo de célula e agem de acordo (movimentação direta ou conversão).

## 3. Comparação com trabalhos relacionados

### Outros Frameworks para Simulação de Tráfego

Ao comparar o framework **MESA** com outras ferramentas, como **SUMO**, **NetLogo** e **AnyLogic**, algumas diferenças e semelhanças relevantes foram identificadas:

- **Escalabilidade:**
  - SUMO é altamente eficiente para grandes redes rodoviárias, suportando

milhares de veículos simultaneamente. Em contraste, o MESA é mais adequado para simulações em menor escala devido à sua abordagem baseada em agentes.

- NetLogo, por sua simplicidade, é menos eficiente para simulações de tráfego em larga escala.
- **Complexidade:**
  - SUMO e AnyLogic exigem maior esforço de configuração inicial devido à riqueza de funcionalidades, como importação de mapas reais e análise de tráfego em redes complexas.
  - MESA é mais simples de configurar e personalizar para experimentos acadêmicos e protótipos.
- **Facilidade de uso:**
  - NetLogo é considerado o mais acessível para iniciantes, enquanto MESA oferece boa flexibilidade para programadores Python com conhecimento intermediário. AnyLogic, apesar de ser robusto, apresenta uma curva de aprendizado mais íngreme devido à sua interface orientada a objetos.

## Soluções Multiagentes em Diferentes Contextos

O trabalho desenvolvido possui paralelos com outras aplicações de sistemas multiagentes (MAS):

- **Logística:** Modelos semelhantes são usados para simular o fluxo de mercadorias em centros de distribuição, onde agentes representam veículos de transporte ou robôs autônomos.
- **Redes de sensores:** Em redes de sensores inteligentes, agentes colaboram para monitorar e gerenciar dados ambientais, como temperatura ou tráfego de informações.
- **Jogos:** Em jogos de simulação, como *SimCity*, agentes desempenham papéis semelhantes, representando indivíduos ou veículos com comportamentos locais e emergentes.

Em suma, a vantagem do MAS sobre outros métodos de simulação reside na capacidade de modelar comportamentos complexos de sistemas reais com interações locais e descentralizadas que, no entanto, permitem a análise de padrões globais emergentes, como congestionamentos ou fluxos otimizados. O uso do MESA demonstrou ser eficaz para a exploração de problemas no campo do tráfego urbano devido à sua flexibilidade e facilidade de personalização para cenários experimentais.

## 4. Vantagens e limitações

### Pontos Fortes

- **Capacidade de modelar comportamentos individuais e interações diretas entre agentes:**

A abordagem baseada em agentes permite representar comportamentos autônomos e descentralizados de cada veículo no sistema. Essa granularidade é especialmente útil para simular decisões locais, como acelerar, frear ou esperar em cruzamentos, o que contribui para análises detalhadas de padrões emergentes no tráfego.
- **Flexibilidade na adaptação do modelo a diferentes cenários:**

O uso do framework **MESA** proporciona uma alta personalização. É possível ajustar os parâmetros dos agentes (e.g., velocidades, prioridades), modificar a infraestrutura simulada (e.g., número de faixas, presença de semáforos), e até mesmo incluir regras personalizadas, como zonas de pedestres ou vias exclusivas para transporte público. Essa versatilidade facilita a aplicação em diferentes contextos urbanos, desde pequenos bairros até redes maiores.
- **Facilidade de implementação e personalização:**

Sendo desenvolvido em Python, o MESA oferece integração com uma ampla variedade

de bibliotecas e ferramentas de análise de dados, como NumPy, Pandas e Matplotlib, o que simplifica a análise de resultados e o ajuste fino do modelo.

---

## Desafios e Limitações

- **Restrições computacionais em cenários com grande número de agentes:**  
À medida que o número de agentes na simulação aumenta, os requisitos de processamento crescem exponencialmente. Essa limitação é especialmente evidente em simulações de tráfego intenso, onde cada agente precisa processar interações complexas com os demais. Diferentemente de frameworks como **SUMO**, que são otimizados para simulações em larga escala, o desempenho do MESA em cenários densos é mais limitado.
- **Dificuldade na calibração de parâmetros para simulações realistas:**  
Configurar parâmetros, como tempos de semáforo, velocidades médias e prioridades de direção, pode ser desafiador, especialmente para representar cenários reais. Pequenas alterações nos valores podem levar a grandes mudanças nos resultados, exigindo testes iterativos e validação cuidadosa com dados empíricos.
- **Escalabilidade comparada a sistemas baseados em aprendizado de máquina ou otimização:**  
Embora os modelos multiagentes sejam eficazes para capturar interações locais e comportamentos emergentes, sua escalabilidade é inferior à de abordagens baseadas em aprendizado de máquina (ML) ou otimização. Sistemas de ML, como redes neurais, podem aprender padrões de tráfego em larga escala de forma mais eficiente e podem ser utilizados para prever congestionamentos ou otimizar o fluxo em redes grandes. Por outro lado, algoritmos de otimização (e.g., algoritmos genéticos) são melhores para encontrar soluções globais, como temporizações ideais de semáforos, sem depender de representações explícitas de cada agente.
- **Foco limitado na integração com dados reais:**  
Frameworks como **SUMO** possuem ferramentas integradas para importar mapas reais e dados de tráfego (e.g., OpenStreetMap, dados de sensores). O MESA, por sua vez, é mais genérico e pode exigir esforço adicional para incorporar cenários baseados em dados do mundo real.

## 5. Discussões sobre aplicabilidade real

### Validação dos Resultados

A qualidade da aplicabilidade real dos resultados do modelo dependerá de quão bem ele será validado nas séries temporais dos dados reais. Para tornar a simulação confiante, deve ser calculada imediatamente com base em diferentes métricas-chave, como o fluxo de carros, o tempo de viagem de carro médio e os padrões de congestionamento com os dados reais de sensores urbanos, câmeras de trânsito, estudos de mobilidade, e assim por Destruição Validação pode ser dividida em:

1. Comparar trajetórias simuladas com rotas reais observadas.
2. Analisar a correlação entre os tempos simulados em cruzamentos e medições reais.
3. Verificar se padrões emergentes, como congestionamentos ou áreas de tráfego intenso, correspondem aos observados em mapas de calor urbanos.

Embora o modelo seja simplificado, ele pode fornecer insights úteis sobre o comportamento geral do tráfego, desde que os parâmetros sejam ajustados para refletir condições reais, como densidade de veículos, limites de velocidade e tempo dos semáforos.

---

## Extensões Potenciais

### 1. **Integração com dados reais de sensores de tráfego:**

O modelo poderia ser estendido para utilizar dados de sensores de tráfego em tempo real, como aqueles fornecidos por dispositivos IoT, radares ou sistemas de monitoramento de câmeras. Esses dados poderiam servir tanto para inicializar o modelo (e.g., definindo o número de agentes em uma determinada área) quanto para ajustar os parâmetros dinamicamente, permitindo uma simulação mais precisa e adaptativa.

### 2. **Uso de aprendizado de máquina para agentes adaptarem seus comportamentos:**

Incorporar algoritmos de aprendizado de máquina permitiria que os agentes (e.g., veículos e semáforos) se tornassem mais inteligentes e adaptativos. Por exemplo:

- Semáforos poderiam aprender a otimizar seus tempos com base no fluxo local, minimizando congestionamentos.
- Veículos poderiam adotar comportamentos mais sofisticados, como prever congestionamentos futuros e planejar rotas alternativas.
- Redes neurais ou algoritmos de reforço profundo poderiam ser usados para modelar interações mais complexas, especialmente em cenários onde os dados históricos estão disponíveis.

### 3. **Expansão para áreas urbanas complexas:**

O modelo poderia ser aplicado em mapas urbanos reais utilizando ferramentas como **OpenStreetMap** para importar redes viárias e definir cenários mais realistas, como ruas com diferentes níveis de prioridade, faixas exclusivas e rotas alternativas.

---

## Impacto na Tomada de Decisão

A abordagem proposta pode ser uma ferramenta valiosa para apoiar gestores urbanos na formulação de políticas de tráfego e planejamento de mobilidade. Alguns exemplos de impacto incluem:

- **Simulação de cenários antes da implementação de mudanças:**

Gestores podem usar o modelo para testar diferentes estratégias, como ajustar tempos de semáforos, implementar faixas exclusivas ou restringir certas direções em cruzamentos. A simulação permite prever os impactos antes da execução real, reduzindo custos e riscos associados a decisões mal informadas.

- **Identificação de gargalos e otimização de fluxos:**

Os resultados podem ajudar a identificar cruzamentos problemáticos, horários de pico mais críticos e áreas com maior propensão a congestionamentos. Com base nessas informações, é possível alocar melhor os recursos de fiscalização e implementar medidas de mitigação, como redirecionamento de tráfego.

- **Base para soluções de cidades inteligentes:**

Integrado a sistemas de transporte inteligentes, o modelo pode contribuir para a automação de decisões, como ajustes dinâmicos de semáforos, envio de alertas para motoristas e adaptação em tempo real a mudanças no tráfego. Além disso, os dados gerados podem ser usados para informar políticas públicas, como incentivos ao uso de transporte público ou caronas compartilhadas.

## 6. Conclusão e trabalhos futuros

Este trabalho demonstrou a aplicabilidade do paradigma multiagente (MAS) na simulação de tráfego urbano, evidenciando como a interação entre agentes pode fornecer uma visão detalhada e realista de fenômenos complexos, como congestionamentos e padrões de fluxo. O modelo implementado destacou-se pela capacidade de representar comportamentos individuais e suas interações no ambiente simulado, permitindo a análise de métricas relevantes, como tempo médio de viagem, tempos de espera em cruzamentos e padrões de congestionamento.



Os resultados mostraram que o uso de sistemas multiagentes oferece uma ferramenta flexível e adaptável para explorar soluções de mobilidade urbana, além de abrir caminhos para aplicações em planejamento de infraestrutura e gestão de tráfego em tempo real. A implementação utilizando o framework MESA reforçou a simplicidade e eficiência na criação de cenários customizados, contribuindo para o aprendizado e experimentação no campo da simulação.

Além disso, o trabalho evidenciou os desafios inerentes a simulações realistas, como a necessidade de calibrar parâmetros adequados e as limitações computacionais quando se lida com cenários de grande escala.

---

## Trabalhos Futuros

Embora o modelo atual tenha alcançado seus objetivos iniciais, há diversas possibilidades de extensão e aprimoramento que podem ampliar o impacto e a aplicabilidade prática do trabalho. Entre elas:

### **Expansão do modelo para incluir pedestres e ciclistas:**

A inclusão de agentes representando pedestres e ciclistas permitiria analisar a interação entre diferentes modos de transporte, contribuindo para o planejamento de soluções mais inclusivas e seguras para todos os usuários das vias urbanas. Essa expansão poderia incluir a simulação de faixas exclusivas, cruzamentos compartilhados e o impacto de medidas como passarelas e ciclovias.

- **Avaliação de políticas de tráfego diversificadas:**

Simulação de rodízio de veículos para reduzir congestionamentos em horários de pico.

Avaliação de estratégias para mudanças dinâmicas no tempo dos semáforos, considerando diferentes horários do dia ou condições de tráfego em tempo real.

Testes de impacto de novas políticas públicas, como tarifas de congestionamento urbano (e.g., pedágio em áreas centrais) ou incentivos ao transporte coletivo.

- **Integração com aprendizado de máquina:**

O uso de aprendizado de máquina pode ajudar a melhorar a adaptação dos agentes, permitindo que aprendam a reagir de forma mais eficiente às mudanças no ambiente de tráfego. Isso incluiria a capacidade de prever congestionamentos e ajustar comportamentos em tempo real.

- **Simulação de redes urbanas reais:**

A aplicação do modelo em redes viárias reais, utilizando dados provenientes de fontes como **OpenStreetMap** ou sistemas de monitoramento de tráfego, poderia validar ainda mais sua aplicabilidade e relevância prática.

- **Otimização computacional e escalabilidade:**

Investir em técnicas de paralelismo e distribuição de carga, utilizando tecnologias como MPI ou integração com a nuvem, permitiria simular cenários maiores e mais complexos sem comprometer o desempenho.

- **Aplicação em contextos emergentes de mobilidade:**

Investigar o impacto de novas tecnologias, como veículos autônomos e sistemas de carona compartilhada, no tráfego urbano. Esses cenários poderiam fornecer insights sobre como preparar as cidades para o futuro da mobilidade.