Matheus Guimarães Couto de Melo Afonso

Uso de agentes de IA para gerenciamento de tráfego baseado em emissões de carbono

Belo Horizonte, Minas Gerais
2025

Matheus Guimarães Couto de Melo Afonso

Uso de agentes de IA para gerenciamento de tráfego baseado em emissões de carbono

Relatório final submetido como requisito para a disciplina "Monografia em Sistemas de Informação 2" do Bacharelado em Sistemas de Informação na UFMG

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Orientador: Prof. Dr. Michele Nogueira Lima

Belo Horizonte, Minas Gerais
2025

Resumo

De acordo com dados do ano de 2019, 23% das emissões de dióxido de carbono na atmosfera são causadas pelo setor de transportes. A concentração alta de CO_2 na atmosfera impacta de múltiplas maneiras na saúde das pessoas, sendo possível observar efeitos respiratórios e cardiovasculares, além de estar associada a diabetes tipo 2 e demência. Diante desse cenário, este trabalho aborda o desenvolvimento de um agente de inteligência artificial que tem como objetivo gerenciar o tráfego de veículos com base em emissões de gás carbônico (CO_2) em uma cidade inteligente. O agente foi treinado usando o Deep Q-Network (DQN), algoritmo de aprendizado por reforço que combina o método clássico Q-Learning com redes neurais profundas. Por meio da interação com o ambiente de simulação e da análise das recompensas recebidas, o agente aprendeu uma política que reduziu o nível total de CO_2 em Zonas de Baixa Emissão (ZBE) em até 50% sem impactar no trânsito de maneira significativa.

Palavras-chave: Agentes de inteligência artificial; Gerenciamento de tráfego; Aprendizado por reforço; Emissão de carbono; Zonas de Baixa Emissão; Deep Q-Network; Cidades inteligentes

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

DQN	Deep Q-Network
ZBE	Zona de Baixa Emissão
MDP	Processo de Decisão de Markov

Sumário

1 Introdução

De acordo com dados da International Energy Agency (IEA) [1], 23% das emissões de dióxido de carbono (CO_2) no ano de 2019 foram causadas pelo setor de transportes. Dessa parcela de emissões, 70% provêm de veículos terrestres, dados que se intensificam quando considerados apenas países em desenvolvimento. Outros dados [2], coletados pelo *Centraal Bureau voor de Statistiek (CBS)*, um instituto neerlandês de estatística, mostram que, após um declínio entre 2019 e 2020, causado pela pandemia de COVID-19, as emissões de CO_2 causadas pelo tráfego de veículos terrestres nos Países Baixos cresceram em média 2,5% ao ano entre 2020 e 2023.

A alta concentração de CO_2 na atmosfera não se trata apenas de um problema climático; ela tem um impacto direto na saúde pública. Conforme um relatório [3] do *Health Effects Institute*, publicado em 2025 e elaborado a partir de dados de 2023, a poluição do ar tem implicações em diversos sistemas do corpo humano. No documento, o instituto expõe efeitos respiratórios (aumento da incidência de asma, pneumonia e câncer de pulmão) e cardiovasculares (insuficiência cardíaca, derrame, aumento da pressão arterial), além de conexões com diabetes tipo 2 e demência. O relatório também destaca que a exposição a altos níveis de poluição na infância possui impactos ainda mais significativos em comparação com a população adulta, podendo levar a um subdesenvolvimento dos pulmões e a doenças pulmonares crônicas. Consequências dessa exposição na infância também levam a aumentos nos índices de mortalidade infantil e de crianças menores de 5 anos [4].

Diante do cenário apresentado, que demonstra a contribuição do setor de transportes para as emissões globais de CO_2 e seus impactos na saúde pública, é importante direcionar esforços para mitigar essas emissões. As duas frentes de ação no que tange a esse objetivo são a transição para fontes de energia limpa no trânsito, como carros elétricos, e a redução imediata das emissões. Este trabalho aborda a segunda dessas alternativas.

1.1 Objetivo Geral

O objetivo geral deste trabalho é investigar a aplicabilidade de agentes de Inteligência Artificial (IA) no gerenciamento do tráfego de veículos, baseado nas emissões de gás carbônico em uma cidade inteligente. O agente é capaz de se adaptar dinamicamente às mudanças na qualidade do ar causadas pelas emissões dos veículos que circulam na cidade, visto que as métricas de emissões são variáveis ao longo do dia.

Por meio de uma simulação, feita através do *Simulation of Urban MObility* (SUMO) [5] em um cenário fidedigno, foram identificadas quais são as zonas da cidade em que os níveis das emissões de CO_2 são mais críticos. A partir disso, foram estabelecidas 10 Zonas de Baixa Emissão (ZBE), regiões onde o tráfego de veículos deve ser controlado de forma

mais rígida com o objetivo de controlar a emissão de gás carbônico. Ao interagir com a simulação em tempo real, um agente de IA é capaz de aprender, baseado nos dados gerados pelo fluxo dos veículos, a regular a entrada dos veículos nas ZBEs sem impactar no trânsito de forma significativa. Por conseguinte, após o treinamento, o agente será capaz de contribuir de forma substancial nas reduções das emissões causadas pelo tráfego, levando a impactos positivos na saúde da população.

1.2 Objetivos Específicos

Os objetivos específicos deste trabalho estão divididos em duas partes, que correspondem, respectivamente, às disciplinas de Monografia em Sistemas de Informação 1 e 2 (MSI 1 e MSI 2).

Na primeira etapa, foi elaborado um protótipo de agente usando o Q-Learning, um algoritmo de aprendizado por reforço. O agente interage com o ambiente da cidade, que foi modelada como um grafo, e aprende o melhor caminho entre todos os nós desse grafo, levando em consideração as medidas de distância e poluição de cada aresta do grafo. O agente desenvolvido na primeira etapa do trabalho foi um protótipo rudimentar que considerou poucas características de uma cidade real para seu treinamento. No cenário simplificado que foi elaborado, o agente teve um treinamento satisfatório, convergindo para uma política "ótima".

A etapa final do projeto consistiu, inicialmente, em integrar o agente ao SUMO, para que os dados utilizados no treinamento e o modelo da cidade sejam mais realistas, o que tornou o agente mais robusto. Em seguida, o agente da primeira etapa foi completamente reformulado para usar o Deep Q-Network (DQN), uma evolução do Q-Learning aplicada no contexto de redes neurais, o que permitiu que o agente trabalhe com um espaço de estados e ações mais complexo, o que foi essencial para que a integração com o SUMO fosse bem sucedida. Finalmente, o agente foi treinado ao longo de uma série de simulações do SUMO, interagindo com o ambiente em tempo real e capturando as emissões de CO_2 dos veículos da rede. Ao fim do treinamento, observou-se que o agente aprendeu uma política capaz de reduzir as emissões dentro das ZBEs em até 50%, o que resultado considerado promissor e que, se aplicado a um cenário real mediante as devidas adaptações, poderia impactar positivamente na saúde da população da cidade.

1.3 Organização do texto

Esta monografia está estruturada da seguinte forma. O Capítulo 2 detalha o referencial teórico necessário. O Capítulo 3 descreve a arquitetura proposta e a metodologia de pesquisa. O Capítulo 4 apresenta os resultados obtidos e avalia a performance do agente

treinado. Finalmente, o Capítulo 5 conclui o trabalho, expondo as contribuições e as direções para trabalhos futuros.

2 Referencial Teórico

Este capítulo apresenta os conceitos fundamentais que sustentam a abordagem proposta. Primeiramente, discute-se o conceito de cidades inteligentes. Em seguida, é apresentado o conceito de Zonas de Baixa Emissão, uma medida já implementada que visa reduzir as emissões de carbono e outros poluentes do ar no tráfego. Finalmente, são detalhados conceitos de aprendizado por reforço, a técnica utilizada para o treinamento do agente.

2.1 Cidades Inteligentes

A definição do que é uma cidade inteligente varia na literatura acadêmica de acordo com fatores como escopo, regionalidade e comunidade envolvida [6]. O presente trabalho assume o conceito de que cidades inteligentes são cidades "instrumentadas, interconectadas e inteligentes" [7, 8]. Neste cenário, instrumentadas diz respeito à capacidade de capturar dados do mundo real por meio de sensores; interconectadas significa que os dados coletados são integrados em uma plataforma computacional que comunica a informação entre diversos serviços da cidade; e inteligentes se refere ao uso dessas informações para analisar, modelar, otimizar e visualizar serviços com a proposta de melhorar a eficiência operacional em domínios como energia, saúde, mobilidade ou educação.

Na sua implementação, as cidades inteligentes se valem fortemente da tecnologia de Internet das Coisas (IoT, do inglês *Internet of Things*). Os dispositivos IoT são objetos equipados com sensores capazes de monitorar e receber dados do ambiente [9], coletando informações como temperatura, umidade e localização [10]. Esses dispositivos são conectados a redes sem fio, permitindo a comunicação com a Internet e, notadamente, com serviços de computação em nuvem, estabelecendo um canal de comunicação para controle remoto dos dispositivos IoT.

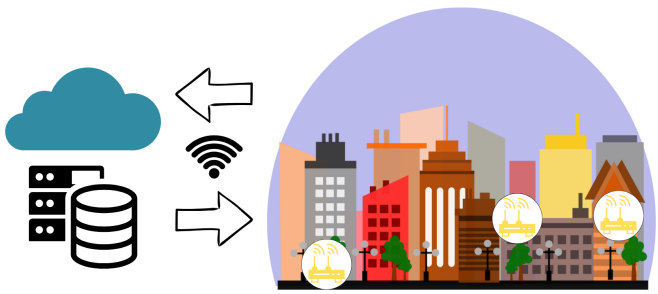

Figura 1 – Representação da comunicação entre a cidade e a nuvem

Um estudo de caso [11] conduzido em 2022 na cidade de Lusail, no Catar, analisa as Tecnologias de Informação e Comunicação (TICs) implementadas na cidade e seus impactos na mobilidade urbana. As TICs implementadas incluem um sistema de gerenciamento e orientação de estacionamento, controle de sinal de tráfego em tempo real e um sistema de informações de tráfego. O estudo observou que a integração da cidade inteligente por meio das TICs otimizou o fluxo de tráfego, fortaleceu o uso do transporte público e reduziu as emissões de carbono causadas pelo congestionamento.

2.2 Zonas de Baixa Emissão (ZBEs)

Perante os desafios para reduzir a poluição do ar urbana, países da União Europeia implementaram gradualmente as chamadas Zonas de Baixa Emissão (ZBE), áreas onde o acesso de veículos é mais restrito. As diretrizes para estabelecer uma ZBE não são universais, podendo cada governo determinar seus próprios padrões como desejar. Em casos mais críticos, a circulação de veículos não elétricos pode ser proibida de maneira geral [12]. Os primeiros registros da introdução de Zonas de Baixa Emissão datam de 1996, em Estocolmo, na Suécia [13]. Nos dias de hoje, cidades como Berlim, Londres, Madrid e Lisboa também adotam essa medida [14].

Ao analisar os efeitos da introdução das ZBEs na Alemanha, nota-se que elas reduzem as concentrações de material particulado no ar em 3%, além de resultarem em uma redução do número de pacientes com doenças cardiovasculares em cerca de 3%, efeito que se intensifica quando considerada a faixa etária acima dos 65 anos [15]. Dados de hospitais localizados dentro de ZBEs alemãs também mostram impactos positivos, registrando menos diagnósticos tanto de doenças cardiovasculares quanto de doenças respiratórias crônicas [12]. Hospitais localizados fora das ZBEs mas nas mesmas cidades não registram mudanças significativas nas composições das hospitalizações.

2.2.1 ZBEs Dinâmicas

O conceito original das ZBEs prevê a criação de áreas estáticas sujeitas à regulação do tráfego de acordo com características de emissão dos veículos. Contudo, uma vez que o tráfego na cidade não para, as medidas de qualidade do ar são voláteis. Em vista disso, é interessante explorar o conceito de ZBEs dinâmicas[16], idealizando um modelo de criação de zonas adaptáveis que se ajustam em tempo real de acordo com dados atualizados de tráfego e emissões de carbono.

2.3 Aprendizado por reforço

O aprendizado por reforço é uma técnica de *Machine Learning* (ML) na qual o agente aprende a partir da interação com o ambiente, buscando maximizar uma função de recompensa. Diferentemente de outras técnicas de ML, o aprendizado por reforço não precisa de dados históricos rotulados para treinar o agente.[17] Através do treinamento, o agente aprende uma política que, nesse contexto, nada mais é do que um mapeamento de estados para ações. A meta do aprendizado por reforço é aprender uma política ótima, de forma que as recompensas estimadas sejam tão grandes quanto possível.

2.3.1 Conceitos

- **Política (π):** mapeamento de estados para ações que guia o comportamento do agente.
- **Episódio:** sessão de treinamento do agente. São necessários múltiplos episódios para treinar o agente de forma efetiva.

2.3.2 Processos de Decisão de Markov (MDP)

O aprendizado por reforço é formalizado matematicamente como um Processo de Decisão de Markov (MDP, do inglês Markov Decision Process) [17]. Um MDP é definido pela tupla (S, A, P, R), em que:

- **Estado (S):** conjunto de estados possíveis.
- **Ações (A):** conjunto de ações possíveis.
- **Função de transição (P):** função dada por $P_a = Pr(s_{t+1}|S = s_t, A = a)$ que descreve a probabilidade de ir para o estado s_{t+1} quando a ação a é tomada no estado s_t.
- **Recompensa (R):** função de recompensa que determina o valor recebido pelo agente ao passar pela transição.

A interação entre agente e ambiente ocorre em uma sequência discreta de passos de tempo. Em cada passo, o agente observa o estado atual S_t, escolhe uma ação a de acordo com sua política π, recebe a recompensa R_{t+1} e observa o novo estado S_{t+1}. O objetivo do aprendizado por reforço é encontrar uma política π que maximize as recompensas esperadas a partir de qualquer estado inicial.

O termo "Markov" refere-se à propriedade de que a transição para o próximo estado depende apenas do estado e da ação atuais, e não do histórico completo que precedeu o passo atual:

$$P(S_{t+1} \mid S_t, A_t) = P(S_{t+1} \mid S_0, A_0, \ldots, S_t, A_t)$$

Sendo assim, o estado deve conter toda a informação relevante para prever o comportamento futuro. Essa exigência motiva, na modelagem proposta neste trabalho (Seção 3.4), a inclusão de uma janela de memória recente e do histórico de ações aplicadas nas ZBEs no vetor de estado, dado que uma leitura instantânea da emissão de CO_2, isolada, não seria suficiente para que o agente inferisse a dinâmica recente do sistema.

2.3.3 Exploração VS *Exploitation*

Um dos desafios que surge na utilização do aprendizado por reforço é o dilema entre exploração e *exploitation*. Dado que o objetivo do aprendizado por reforço é maximizar o ganho de recompensas, parece intuitivo que o agente sempre busque executar aquelas ações que resultem em maior acúmulo das recompensas, ou seja, *exploitando* as ações ótimas. Contudo, uma vez que no início do treinamento essa experiência é inexistente, o agente precisa explorar o ambiente, executando ações desconhecidas, de forma a entender quais são as ações que geram as melhores recompensas. Ao fazer isso, ele contradiz sua premissa inicial de maximização dos ganhos, uma vez que executa ações não-ótimas.

Nesse cenário, é de suma importância saber balancear os dois processos para atingir a convergência em uma política ótima. Um dos modos de fazer isso é usando um modelo de transição estocástico para escolher as ações. Para isso, é definida uma probabilidade $\varepsilon, 0 \leq \varepsilon \leq 1$ com a qual uma ação aleatória é escolhida dentre o conjunto de ações possíveis em um estado. Quando a ação escolhida é aleatória, o agente explora as possibilidades; quando a ação escolhida é ótima, é feita a *exploitation*. Conforme o treinamento progride, o valor de ε decai para favorecer a *exploitation*.

2.3.4 Hiperparâmetros

Hiperparâmetros são variáveis usadas no treinamento de modelos de ML para obter melhor desempenho na execução de suas tarefas [18]. Os hiperparâmetros devem ser ajustados através da experimentação, com o propósito de definir os valores ideais de acordo com cada cenário. Neste trabalho, os hiperparâmetros relevantes são os seguintes:

- **Taxa de aprendizado (α):** determina com qual intensidade as informações novas substituem as informações já aprendidas.
- **Fator de desconto (γ):** mede a importância das recompensas futuras em relação às recompensas imediatas. Quanto mais próximo de 1, maior a relevância das recompensas de longo prazo.
- **Epsilon (ε):** controla o equilíbrio entre exploração e *exploitation*. Quanto mais alto, maior é o estímulo à exploração. A medida que o treinamento progride e o agente aprende, o valor de ε decai para favorecer a *exploitation*.

Os três hiperparâmetros acima são definidos no intervalo entre 0 e 1.

2.3.5 Iteração de Política Generalizada

A Iteração de Política Generalizada é um paradigma de aprendizado por reforço que divide o aprendizado em duas fases que se alternam indefinidamente: avaliação e aprimoramento de política. Na primeira fase, o agente avalia o quão boa é sua política atual em termos de recompensas esperadas; em seguida, a partir da avaliação, o agente altera sua política de forma a maximizar os retornos esperados. Ao seguir esse paradigma, é garantido que o algoritmo converge para uma política ótima [17].

2.4 Deep Q-Network (DQN)

Esta seção apresenta o Deep Q-Network, algoritmo de aprendizado por reforço utilizado para treinar o agente neste trabalho. O DQN é uma extensão direta do Q-Learning, aplicado a um cenário em que o espaço de estados é excessivamente grande para ser tratado de forma tabular. Para contornar esse problema de dimensionalidade, o DQN usa aproximação de função com redes neurais.

2.4.1 Q-Learning

O Q-Learning é um algoritmo de aprendizado por reforço baseado em uma tabela $Q(s, a)$ que armazena as recompensas esperadas para todos os pares possíveis de estados s e ações a. A cada passo do episódio no tempo t, o agente atualiza o valor $Q(S_t, A_t)$ baseado na recompensa obtida R_{t+1} após tomar a ação A_t em S_t e no resultado esperado ao tomar a ação gulosa em S_{t+1}

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{2.1}$$

O Q-Learning é um método tabular. Isso significa que, dado todo o conjunto de estados S e ações A do ambiente, ele armazena uma tabela com todos os pares $Q(s, a)$ possíveis. Em função disso, conforme o espaço de estados e ações cresce, a tabela Q se torna esparsa.

2.4.2 Deep Q-Network

Conforme discutido na subseção anterior, o Q-Learning armazena uma tabela para todos os pares de estado-ação possíveis. Essa abordagem se torna o problema quando o espaço de estados ou ações possíveis é muito grande ou contínuo. O Deep Q-Network [19] resolve esse problema substituindo a tabela $Q(s, a)$ com uma aproximação de funções parametrizada por uma rede neural $Q(s, a, \theta)$, onde θ representa os pesos da rede. Dessa forma, em vez de armazenar um valor para cada par estado-ação possível, a rede recebe o

vetor de estado como entrada e produz, na sua saída, uma estimativa $Q(s, a)$ para cada ação a do espaço de ações. Através dessa generalização, o agente consegue estimar o retorno esperado de pares estado-ação nunca antes observados diretamente a partir da semelhança entre eles [17].

2.4.2.1 Instabilidade no treinamento

Treinar uma rede neural diretamente sobre os alvos do TD (*Temporal Difference*) usados pelo Q-Learning é instável por dois motivos, identificados por [20]: (i) as transições observadas em uma trajetória de treinamento são sequenciais e fortemente correlacionadas, violando o pressuposto de que as amostras são independentes e identicamente distribuídas (i.i.d.) sobre o qual o gradiente estocástico se apoia; e (ii) o alvo do treinamento depende dos mesmos parâmetros θ que estão sendo atualizados, criando um alvo que se move a cada passo de treinamento e dificultando a convergência.

Para mitigar esses dois problemas, o DQN introduz dois mecanismos:

- **Replay buffer:** as transições $(s_t, a_t, r_{t+1}, s_{t+1})$ observadas pelo agente são armazenadas em um buffer de tamanho fixo. A cada passo de treinamento, ao invés de atualizar os pesos da rede considerando apenas a transição mais recente, sorteia-se um *minibatch* aleatório das transições do buffer, quebrando a correlação temporal entre amostras consecutivas e permitindo reaproveitar a mesma transição em múltiplas atualizações.

- **Rede alvo (*target network*):** o DQN mantém uma segunda cópia da rede com parâmetros θ^-, que é usada exclusivamente para calcular o alvo do treinamento:

$$y = r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) \tag{2.2}$$

  Enquanto a rede principal, de pesos θ, se atualiza a cada passo, os valores de θ^- são atualizados de forma mais lenta. Assim, o alvo permanece estável durante várias atualizações consecutivas, evitando o problema do alvo móvel exposto por [20].

A atualização dos pesos da rede alvo pode ser feita de duas formas diferentes. A tradicional, chamada também de *hard update*, consiste em periodicamente fazer uma cópia direta de θ. A segunda forma, que é utilizada neste trabalho, é feita via uma média móvel exponencial dos pesos a cada passo, técnica conhecida como *soft update* ou *Polyak averaging* [21]:

$$\theta^- = \tau\theta + (1 - \tau)\theta^-, \quad \tau \ll 1 \tag{2.3}$$

O uso dessa atualização suave evita os saltos abruptos que ocorrem na rede alvo após uma cópia completa.

2.4.2.2 Função de perda

O treinamento da rede $Q(s, a, \theta)$ é feito minimizando o erro entre a estimativa atual e o alvo de treinamento. Embora a formulação original do DQN utilize o erro quadrático médio (*Mean Squared Error*, MSE), é comum na literatura substituí-lo pela perda de Huber [22], que se comporta como o erro quadrático para erros pequenos e como o erro absoluto para erros grandes:

$$L_\delta(y, Q(s,a;\theta)) = \begin{cases} \frac{1}{2}\left(y - Q(s,a;\theta)\right)^2 & \text{se } |y - Q(s,a;\theta)| \leq \delta \\ \delta\left(|y - Q(s,a;\theta)| - \frac{1}{2}\delta\right) & \text{caso contrário} \end{cases} \tag{2.4}$$

Essa propriedade torna o treinamento mais robusto a erros de TD anômalos, evitando que esses outliers dominem o gradiente e desestabilizem o aprendizado, sem abandonar a suavidade do MSE para erros pequenos, onde a precisão do ajuste é mais relevante.

2.4.3 Hiperparâmetros do DQN

A substituição da tabela $Q(s, a)$ por uma rede neural introduz hiperparâmetros decorrentes da arquitetura da rede e dos mecanismos de replay de experiência e rede-alvo:

- **Arquitetura da rede:** número de camadas ocultas e de neurônios por camada, que determina a capacidade da rede de representar funções Q complexas.
- **Capacidade do *replay buffer*:** número máximo de transições armazenadas para amostragem. Conforme novas amostras são coletadas, as mais antigas são descartadas.
- ***Batch size*:** quantidade de transições amostradas do buffer de replay a cada passo de treinamento para calcular o gradiente.
- **Taxa de atualização da rede-alvo (τ):** parâmetro associado ao *soft update* que controla a velocidade com que a rede-alvo incorpora os pesos da rede principal.

2.5 Trabalhos relacionados

Não foram encontradas muitas fontes na literatura que tratam da temática da redução das emissões por meio do gerenciamento dinâmico do tráfego de veículos. Esta seção apresenta um trabalho de revisão sobre a efetividade das Zonas de Baixa Emissão Dinâmicas, tratando um escopo semelhante ao analisado neste trabalho.

2.5.1 Dynamic Low-Emission Zones for Urban Mobility: A Systematic Review

Em [16], os autores fazem uma análise sistemática da implementação de Zonas de Baixa Emissão Dinâmicas em múltiplos contextos. Por meio do estudo, observaram-se melhorias consideráveis nas emissões veiculares em comparação com cenários que abordam o uso das ZBEs estáticas. Além disso, o uso da abordagem dinâmica não causou impactos significativos no fluxo habitual de tráfego.

Na conclusão do trabalho, os autores ressaltam os desafios mais críticos na implementação das ZBEs dinâmicas. Um dos obstáculos que eles tratam com maior atenção é a complexidade de tratar grandes volumes de dados mantendo a integridade dos mesmos, bem como preocupações com a necessidade de atualizar o modelo em curtos intervalos de tempo. Além disso, todo esse processo deve ser feito garantindo a privacidade dos dados gerados pelos veículos, meta que é atingida em alguns casos com o uso de criptografia e *blockchain* na comunicação entre os carros e o agente central que treina o modelo da ZBE.

3 Metodologia

Como mencionado anteriormente, foi criado e treinado um agente com aprendizado por reforço que gerencia o tráfego em uma cidade inteligente, com o objetivo de controlar e reduzir as emissões de gás carbônico. Este capítulo descreve como o agente foi criado, incluindo detalhes sobre o cenário e a simulação no SUMO, a modelagem dos estados e ações, e a construção do agente usando o algoritmo de Deep Q-Network. O trabalho foi conduzido usando a linguagem Python.

3.1 *Simulation of Urban MObility* (SUMO)

O *Simulation of Urban MObility* (SUMO) [5] é um simulador de tráfego microscópico de código aberto, desenvolvido e mantido pelo *German Aerospace Center*. Os veículos do SUMO são modelados individualmente, de maneira que cada um dos veículos possui posição, velocidade e aceleração próprias. Essa granularidade, em oposição a modelos macroscópicos (que tratam o tráfego como um fluxo agregado), permite que as características dinâmicas dos veículos sejam extraídas a cada passo da simulação. Dessa forma, se torna possível analisar dados de emissões, consumo de combustível e roteamento dos automóveis em tempo real.

Além de se sobressair pela independência na operabilidade dos veículos, as dinâmicas de controle longitudinal e lateral sobre os veículos também merecem destaque. O comportamento de aceleração é controlado pelo modelo *car-following* de Krauss [23], enquanto a troca de faixas segue um modelo próprio do simulador. O uso desses mecanismos faz com que o SUMO funcione de forma fiel a um fluxo de tráfego real.

Por fim, o SUMO também oferece uma interface gráfica que permite acompanhar a simulação em tempo real. Através dela, é possível observar o comportamento dos carros, trocas de faixa, cruzamentos e funcionamento dos semáforos, como mostra a Figura 2.

3.1.1 *Traffic Control Interface* (TraCI)

O *Traffic Control Interface* (TraCI) é uma API que permite que um programa externo interaja em tempo real com a simulação SUMO em andamento. Através do TraCI, é possível ler o estado da simulação (posição dos carros, velocidade, emissões) e intervir sobre ela, alterando rotas, tempo de viagem percebido e o estado dos semáforos. No contexto do uso de um agente para controlar o tráfego, o TraCI é quem viabiliza o aprendizado, pois é através dele que o agente consegue, a cada passo de decisão, observar o estado da rede e aplicar as ações recomendadas.

Figura 2 – Interface gráfica do SUMO - Imagem aumentada

3.2 Cenário e ambiente de simulação

Os experimentos foram conduzidos sobre um cenário microscópico de tráfego construído no simulador SUMO, correspondente a um recorte do centro de Belo Horizonte. A rede viária foi importada do *OpenStreetMap* [24] através da ferramenta `osmWebWizard` disponibilizada pelo SUMO. A escolha pela região central se deu porque ali se concentra o tráfego mais intenso e, consequentemente, os maiores níveis de emissão. A imagem 3 mostra a área escolhida para o projeto.

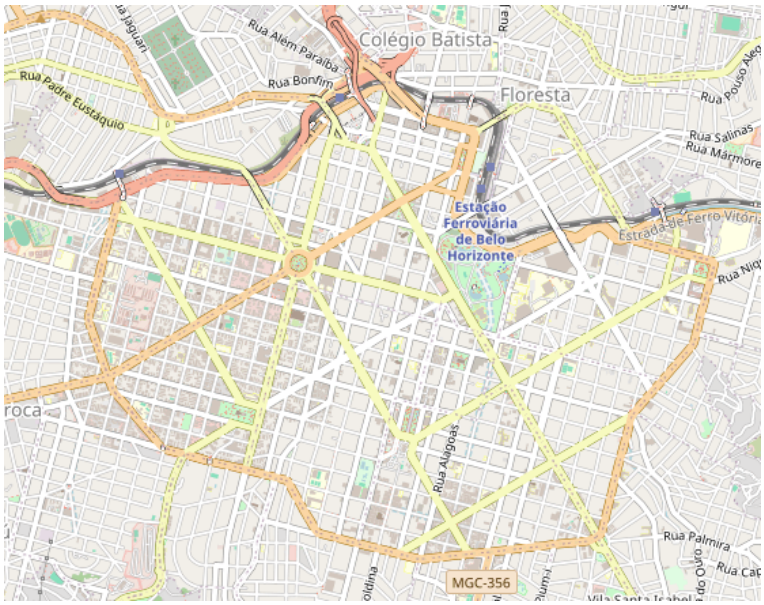

Figura 3 – Região central de Belo Horizonte na qual o treinamento foi feito

A região escolhida consiste de uma malha de 5410 arestas direcionadas que representam as ruas da cidade. Tratar o problema apresenta num nível de granularidade tão pequeno causa dois problemas, tanto em relação ao custo computacional de monitorar as emissões

específicas de cada uma das ruas, quanto para gerar os estados a serem avaliados pelo agente. Frente a esse cenário, as arestas foram agregadas em 100 zonas por meio do algoritmo de clusterização K-Means aplicado sobre o centroide geográfico de cada aresta. A partir dessa divisão, foram definidas 10 Zonas de Baixa Emissão, cuja seleção foi orientada por uma simulação sem intervenção e é detalhada na seção a seguir. A imagem 4 mostra o resultado do algoritmo de clusterização.

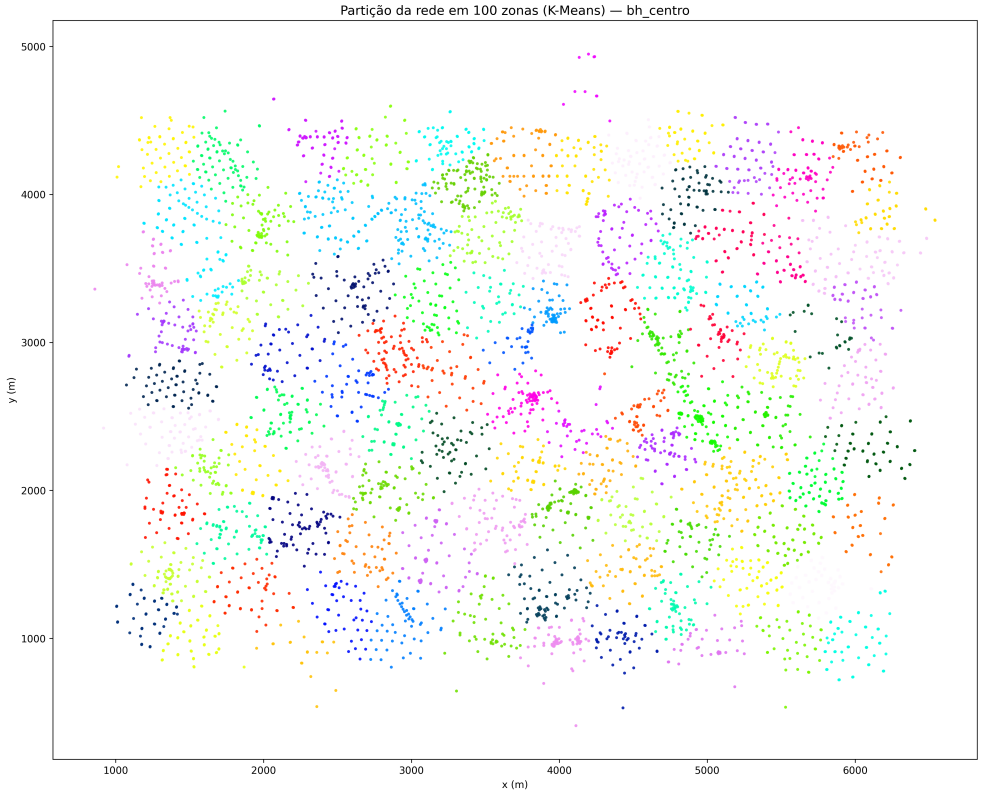

Figura 4 – Particionamento das 100 zonas via K-Means

3.3 Seleção das ZBEs

Com o cenário bem definido e a clusterização das zonas já executada, a próxima etapa foi selecionar as 10 Zonas de Baixa Emissão a serem controladas pelo agente. Em um primeiro momento, as ZBEs foram selecionadas aleatoriamente, mas isso rapidamente se mostrou um problema, uma vez que não faz sentido restringir o acesso em regiões onde os níveis de poluição não são críticos. Com esse problema em mente, foi realizada uma simulação de 1 hora sem intervenção do agente, a fim de visualizar qual é o padrão de poluição que ocorre no centro de Belo Horizonte. O resultado da execução desse *baseline* gerou o mapa de calor exibido na imagem 5.

O *baseline* evidencia a concentração das emissões de CO_2 em poucas zonas, que se localizam no interior da região central, em amarelo e laranja e, em menor escala, em vias

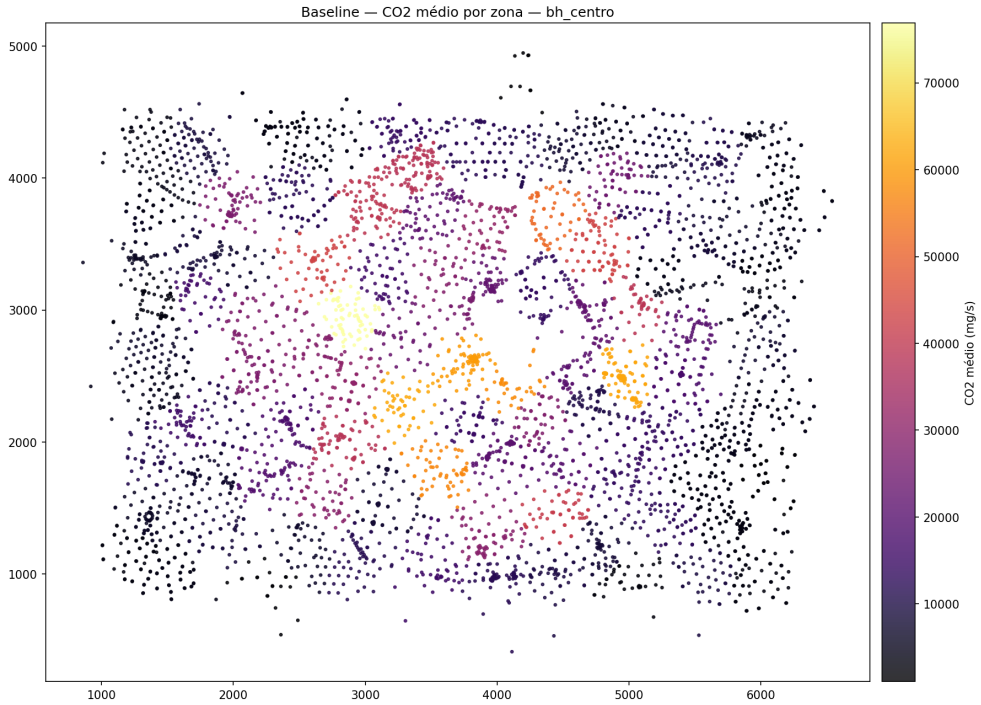

Figura 5 – Baseline de emissões de CO_2 por zona

de acesso ao interior da Avenida do Contorno, áreas destacadas em vermelho, enquanto as áreas mais periféricas apresentam um nível de emissão consideravelmente inferior. O ranqueamento das zonas permitiu, então, escolher as 10 ZBEs a serem utilizadas no treinamento (Imagem 6). Foram selecionadas as 5 zonas mais poluídas e 5 de poluição mediana, para avaliar a generalização da política aprendida. Para cada uma delas, o *baseline* registrou o percentil 95 (`max_co2_zone`), usado para normalizar o estado de CO_2 de cada zona, e o percentil 50 (`limiar_zbe`), a fim de recompensar a zona com base em se ela está operando acima ou abaixo de suas emissões médias.

3.4 Formulação do MDP

O problema é modelado como um MDP episódico. Cada episódio corresponde a uma simulação completa do cenário, totalizando 1 hora de simulação. O agente toma uma decisão a cada 30 segundos de simulação, totalizando 120 decisões por episódio.

3.4.1 Espaço de estados

O estado combina um conjunto de *features* das emissões de CO_2 com memória temporal, nível de restrição atual das 10 ZBEs e histórico das últimas 2 ações tomadas:

- **Por zona** ($\times 100$)**:** `[CO_2 atual, CO_2 médio na janela, tendência]`, normalizados por `max_co2_zone`. O valor de tendência pertence ao intervalo `[-1, 1]` e

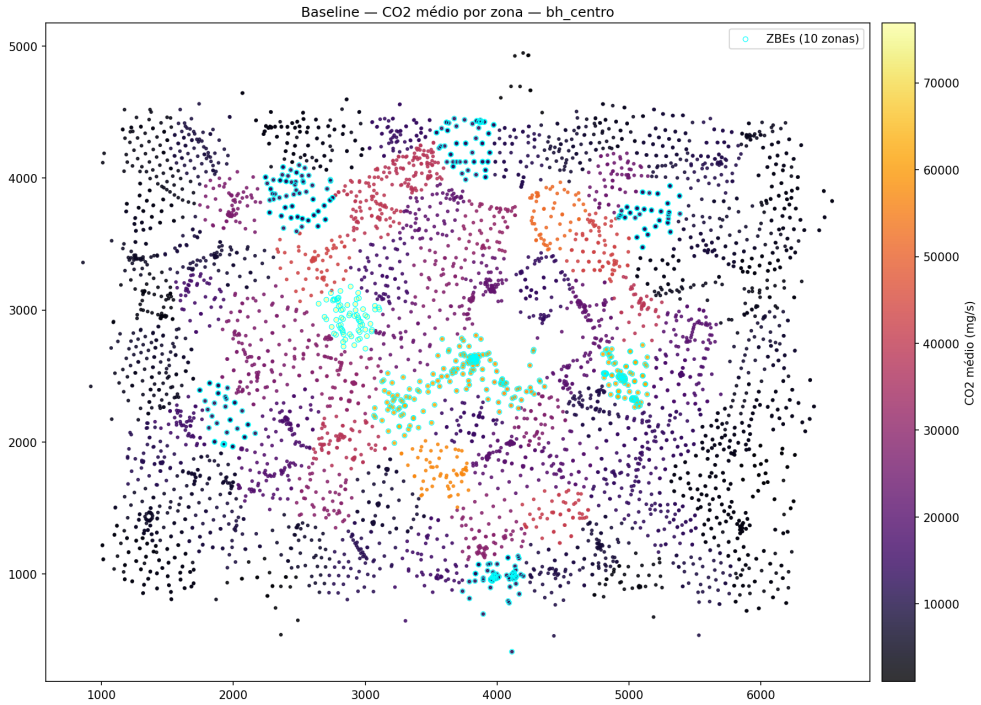

Figura 6 – Baseline de emissões de CO_2 - ZBEs em azul

representa a diferença entre a média da segunda metade e da primeira metade da janela de memória.

- **Por ZBE ($\times 10$):** nível de restrição atual aplicado na zona ($L \in \{0.00, 0.25, 0.50, 0.75, 1.00\}$) e a restrição aplicada a ela nas últimas duas janelas de decisão.

A dimensão total do estado, portanto, é de 330 ($3 \times 100 + 3 \times 10$). O uso da janela de memória das últimas decisões é crítico porque permite que o agente considere o comportamento recente da zona nos últimos 60 segundos, em vez de considerar a emissão apenas no segundo que precede a decisão, visto que a emissão por segundo é ruidosa e poderia prejudicar o aprendizado do agente.

3.4.2 Espaço de ações

A cada passo de decisão, o agente toma uma ação discreta única. As ações disponíveis correspondem a escolher um nível de restrição ($L \in \{0.00, 0.25, 0.50, 0.75, 1.00\}$) a ser aplicado a uma das 10 ZBEs, totalizando 50 ações de ajuste de restrição, somadas à 1 ação de no-op em que o agente escolhe não fazer nada. O nível determinado em cada ZBE persiste até ser futuramente alterado.

3.4.3 Função de recompensa

A função de recompensa consiste de um balanceamento entre as emissões de gás carbônico excedentes (acima do `limiar_zbe`) nas ZBEs e a perda de tempo `timeLoss` média entre todos os veículos ativos na janela de decisão. O uso desses dois fatores é necessário para atingir o objetivo de melhorar as emissões sem impactar no fluxo veicular. A fórmula completa da recompensa é dada por:

$$R = -\frac{\text{w\_emis} \cdot \sum_{zbe} \max(0, \text{co2\_zbe\_janela} - \text{limiar\_zbe})}{\text{emis\_scale}} - \frac{\text{w\_mob} \cdot \text{timeLoss}}{\text{mob\_scale}} \quad (3.1)$$

onde `w_emis = 3.0` e `w_mob = 1.0` são os pesos associados às emissões e à perda de tempo, respectivamente, e `emis_scale` e `mob_scale` são fatores de escala, que foram medidos empiricamente na fase de seleção das ZBEs e são usados para ajustar o tamanho da recompensa.

O termo de mobilidade usa o `timeLoss` médio dos veículos ativos. O `timeLoss` é uma métrica nativa do SUMO que mede todo o tempo adicional gasto no trajeto em comparação com uma viagem sem interferências. Dessa forma, ele captura todo o tempo parado em semáforos, preso em engarrafamentos e também o tempo adicional decorrente de uma rota mais longa.

3.5 Treinamento do agente

O agente foi treinado usando o algoritmo Deep Q-Network. A cada passo de decisão, o agente observa o estado da simulação, escolhe uma ação através de uma política ε-gulosa, aplica a restrição correspondente na ZBE selecionada e armazena a transição $(s_t, a_t, r_{t+1}, s_{t+1})$ no *replay buffer*. O treinamento ocorre ao longo de 120 episódios, cada um correspondendo a uma simulação independente completa do cenário no SUMO.

3.5.1 Enforcement das restrições

O SUMO não oferece, dentro de seu ambiente, o conceito nativo de Zonas de Baixa Emissão. Portanto, foi necessário traduzir o nível de restrição definido pelo agente em uma intervenção concreta dentro da simulação, por meio de um processo de *enforcement*. A modelagem desse processo foi uma etapa crítica do trabalho e foi concluída com sucesso após o descarte de duas abordagens iniciais que produziram efeitos nulos:

1. `adaptTravelTime` **global por aresta:** `adaptTravelTime` é uma função nativa do TraCI que ajusta o tempo de viagem percebido de uma aresta para todos os veículos. Na prática, essa adaptação foi ignorada pelo dispositivo de reroteamento do SUMO, e o efeito sobre o fluxo dos veículos foi nulo mesmo ao aumentar artificialmente a restrição da zona para 100%.

2. **Redirecionamento de veículos dentro da zona escolhida:** ao aumentar a restrição de uma zona, os veículos que já estão circulando dentro dela ou cuja rota foi recalculada antes da restrição não são afetados.

O mecanismo adotado desvia uma fração L dos veículos cuja rota cruza cada zona com restrição $L > 0$. Isso é feito através de duas chamadas TraCI: `setAdaptedTravelTime`, que infla o tempo de viagem percebido das arestas da ZBE para cada veículo individualmente; e `rerouteTravelTime`, forçando o veículo a recalcular sua rota considerando o tempo de viagem inflado pelo passo anterior.

A cada passo de decisão, todos os veículos ativos na simulação são reavaliados, inclusive os que já estão a caminho da ZBE restrita. A decisão de desviar um veículo ou não é feita uma única vez para cada par (veículo, zona) por meio de um sorteio com probabilidade L, e não é re-sorteada em decisões subsequentes, o que produz um desvio estável de uma fração L do fluxo. Veículos cuja origem ou destino estão dentro da própria zona não são afetados pelo *enforcement*, preservando o acesso local.

3.5.2 Captura das emissões e desempenho

A leitura do estado da simulação a cada passo é feita por meio de *subscriptions* do TraCI (`vehicle.subscribe`). Uma vez que um veículo é inscrito, as variáveis de emissões e `timeLoss` são obtidas através de uma única chamada `getAllSubscriptionResults`, o que reduz o tempo de leitura por passo de simulação de 108ms para 31ms. Essa otimização foi essencial para viabilizar o treinamento, reduzindo o tempo total de uma sessão de 14 para 5h, possibilitando a experimentação com hiperparâmetros e pesos de recompensa.

3.5.3 Hiperparâmetros

A Tabela 1 apresenta os valores utilizados para os hiperparâmetros de treinamento do agente.

Tabela 1 – Hiperparâmetros utilizados no treinamento

Hiperparâmetro	**Valor**
Dimensão da camada oculta	128
Taxa de aprendizado (otimizador)	5×10^{-4}
Fator de desconto γ	0,99
ε inicial / mínimo	1,0 / 0,05
Taxa de decaimento do epsilon	0,9998
Capacidade do *replay buffer*	10.000
Tamanho do *batch*	128
Taxa de atualização da rede-alvo (τ)	0,005

4 Resultados

Este capítulo apresenta e avalia o desempenho do agente treinado. Primeiramente, descreve-se o protocolo de avaliação adotado para comparar a política aprendida com políticas de referência. Em seguida, analisa-se a curva de aprendizado obtida ao longo do treinamento. Por fim, os resultados quantitativos são apresentados e discutidos sob a ótica do compromisso entre a redução das emissões e o impacto na mobilidade.

4.1 Protocolo de avaliação

Após o treinamento, a política aprendida pelo agente foi avaliada de forma gulosa ($\varepsilon = 0$), de modo que o agente sempre escolhe a ação com o maior retorno esperado a cada passo de decisão. Para contextualizar o desempenho do agente, ele foi comparado a três políticas de referência fixas:

- ***No-op*:** política que representa a simulação sem intervenção do agente e serve como piso de comparação.
- **Aleatória:** a cada passo de decisão, é escolhida uma ação aleatória do espaço de ações possíveis. Mede quanto da redução decorre de restringir as zonas sem aplicar nenhuma estratégia aprendida.
- **Sempre 100%:** impõe o nível máximo de restrição $L = 1.00$ a todas as ZBEs desde o início de cada simulação. Representa o teto das reduções possíveis de serem alcançadas pelo mecanismo de *enforcement*.

As quatro políticas foram avaliadas sobre o mesmo conjunto de 5 *seeds* de simulação, que controlam a demanda de tráfego. O uso das mesmas *seeds* garante que as diferenças observadas são resultado da política e não de uma variação aleatória do tráfego. Para cada política foram reportadas a média e desvio padrão do CO_2 nas ZBEs, CO_2 global e `timeLoss` médio dos veículos.

4.2 Curva de aprendizado

A figura 7 mostra a evolução das recompensas obtidas pelo agente ao longo de seus 120 episódios de treinamento. Ela mostra um crescimento consistente, partindo de valores ao redor de -80 e se estabilizando próxima de -40 na metade do treino. Esse comportamento indica que o agente progressivamente aprendeu uma política melhor.

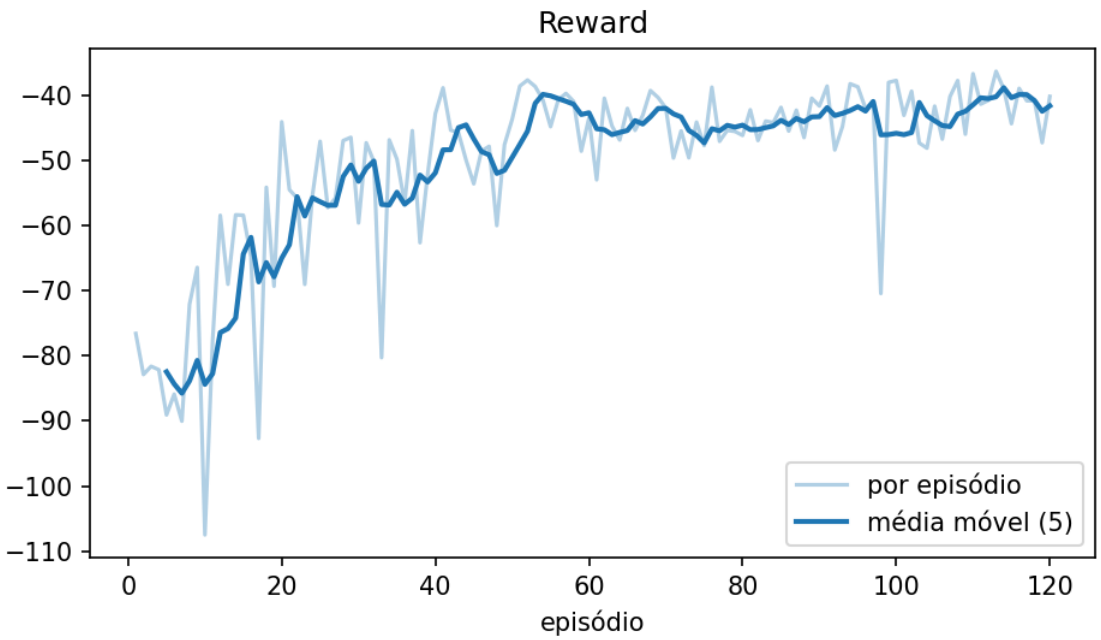

Figura 7 – Recompensa obtida pelo agente ao longo de 120 episódios

De forma análoga, a imagem 8 reforça essa ideia. No início do treinamento, as emissões de CO_2 encontram-se próximas ao *baseline* no-op e caem de maneira progressiva à medida que o treinamento avança, equilibrando-se no mesmo momento em que as recompensas. Isso mostra que a recompensa foi modelada de maneira correta pois, conforme ela cresce, a métrica-alvo se aproxima do objetivo do treinamento.

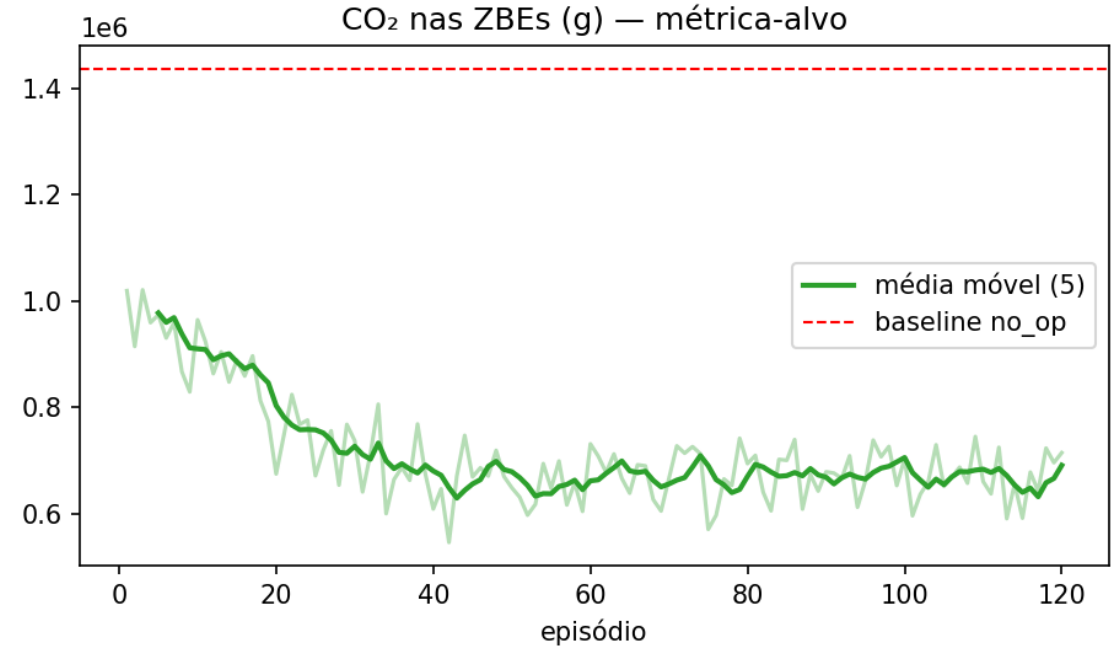

Figura 8 – Evolução das emissões ao longo do treinamento

4.3 Resultados comparativos

A Tabela 2 consolida os resultados da avaliação das quatro políticas sobre as 5 *seeds*, e a Figura 9 mostra os mesmos dados de forma gráfica. As variações percentuais Δ são calculadas em relação à política *No-op*.

Tabela 2 – Comparação das políticas avaliadas (média ± desvio padrão, 5 *seeds*)

Política	CO_2 **ZBEs** ($\times 10^6$ **g**)	CO_2 **global** ($\times 10^6$ **g**)	***timeLoss*** **(s)**
No-op	$1{,}442 \pm 0{,}021$	$6{,}809 \pm 0{,}026$	$113{,}9 \pm 4{,}4$
Aleatória	$0{,}924 \pm 0{,}052$ $(-35{,}9\%)$	$7{,}337 \pm 0{,}040$ $(+7{,}8\%)$	$116{,}0 \pm 5{,}5$ $(+1{,}9\%)$
Aprendida	$\mathbf{0{,}726 \pm 0{,}054}(\mathbf{-49{,}6\%})$	$7{,}535 \pm 0{,}059(+10{,}7\%)$	$119{,}1 \pm 3{,}8$ $(+4{,}5\%)$
Sempre 100%	$0{,}252 \pm 0{,}001$ $(-82{,}5\%)$	$8{,}069 \pm 0{,}019(+18{,}5\%)$	$128{,}7 \pm 1{,}6(+12{,}9\%)$

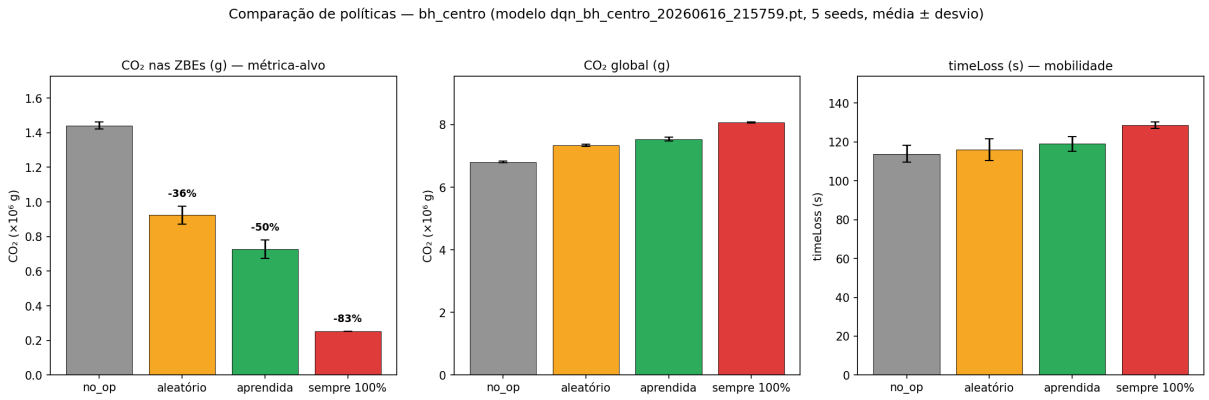

Figura 9 – Comparação das políticas nas três métricas avaliadas

A política aprendida reduziu as emissões nas ZBEs em 49,6% em relação ao cenário No-op. Ao comparar com a política aleatória, podemos ver que, apesar de reduzir o nível de CO_2 em 35,9% simplesmente por aplicar restrições nas zonas, ela é superada pelo agente com certa folga, demonstrando que ele aprendeu onde e quando concentrar as restrições para maximizar a redução nas regiões críticas. Por fim, a política de restrição máxima atinge a maior redução possível (82,5%), mas a um custo de mobilidade proporcionalmente muito mais elevado.

4.4 Análise do *trade-off* entre emissão e mobilidade

Os resultados evidenciam um *trade-off* entre a redução das emissões nas ZBEs e o impacto na mobilidade, medido pelo `timeLoss`. A política de restrição máxima, embora atinja a maior redução de emissões, impõe o maior custo de mobilidade (+12,9%). A política aprendida, por sua vez, posiciona-se em um ponto intermediário: alcança cerca de 60% da redução de emissões obtida pela restrição máxima ($-49{,}6\%$ contra $-82{,}5\%$), porém a aproximadamente um terço do seu custo de mobilidade ($+4{,}5\%$ contra $+12{,}9\%$).

Também cabe destacar uma limitação revelada pelos resultados. Em todas as políticas que aplicam restrições, o CO_2 global da rede aumenta em relação ao cenário sem intervenção. Esse aumento decorre da própria natureza do mecanismo de *enforcement*, porque, ao desviar veículos para fora das ZBEs, suas rotas se tornam mais longas, o que eleva a distância total percorrida e, consequentemente, a emissão total. Portanto, o objetivo alcançado por este trabalho não foi a redução da emissão total da cidade, mas sim a redução da exposição da população à poluição nas zonas consideradas críticas.

5 Conclusão

Os altos níveis de CO_2 e de outros poluentes do ar na atmosfera têm um efeito negativo na saúde da população, estando associados principalmente ao desenvolvimento de doenças respiratórias e cardiovasculares, além de atingirem de forma ainda mais severa grupos vulneráveis, como crianças e idosos. Diante desse cenário, este trabalho investigou a aplicação de um agente de Inteligência Artificial para o gerenciamento do tráfego urbano baseado em emissões de gás carbônico, com o objetivo de reduzir a exposição da população à poluição em zonas críticas de uma cidade inteligente.

Para isso, foi construído um cenário microscópico de tráfego no simulador SUMO a partir de um recorte do centro de Belo Horizonte. A malha viária, composta por 5410 arestas, foi agregada em 100 zonas por meio do algoritmo K-Means. Em seguida, uma simulação de referência sem intervenção permitiu identificar as regiões de maior emissão, que orientaram a seleção das 10 Zonas de Baixa Emissão controladas pelo agente. O problema foi formalizado como um Processo de Decisão de Markov, no qual o agente, a cada passo de decisão, observa o estado das emissões nas zonas e decide o nível de restrição a ser aplicado a cada uma delas. O treinamento foi feito com o algoritmo Deep Q-Network, e o nível de restrição definido por ele foi traduzido em uma intervenção concreta na simulação por meio de um mecanismo de *enforcement* que desvia, em tempo real, uma fração dos veículos cuja rota cruza as zonas restritas.

Os resultados demonstraram que o agente aprendeu uma política não trivial: a redução de 49,6% nas emissões de CO_2 dentro das ZBEs superou com folga a política aleatória (35,9%), evidenciando que o agente aprendeu quando e onde concentrar as restrições. Além disso, a política aprendida posicionou-se em um ponto de *trade-off* favorável: alcançou cerca de 60% da redução obtida pela política de restrição máxima, porém a aproximadamente um terço do seu custo de mobilidade. Dessa forma, o trabalho cumpriu seu objetivo de reduzir as emissões nas zonas críticas sem impactar de forma significativa o fluxo de tráfego.

Apesar dos resultados promissores, o trabalho apresenta limitações a serem destacadas. A principal delas é o aumento do CO_2 global da rede observado em todas as políticas que aplicam restrições: ao desviar veículos para fora das ZBEs, suas rotas se tornam mais longas, elevando a distância total percorrida e, portanto, a emissão agregada. Assim, o ganho obtido é uma redução da exposição local nas zonas sensíveis, e não da emissão total da cidade. Além disso, a avaliação foi conduzida sobre um único cenário com um conjunto fixo de ZBEs definido antes do treinamento. Por fim, o agente leva em consideração apenas as emissões de gás carbônico, ignorando outros poluentes do ar que são igualmente relevantes, como o monóxido de carbono (CO), óxidos de nitrogênio (NO_x) e material particulado (PM).

Como direções para trabalhos futuros, destacam-se: a investigação do efeito de transbordamento (*spillover*) das emissões para as zonas vizinhas às ZBEs, quantificando para onde a poluição é deslocada por conta do reroteamento; a transição de um conjunto fixo para Zonas de Baixa Emissão verdadeiramente dinâmicas, em que o próprio agente seleciona, do conjunto total de zonas da cidade, quais delas restringir conforme as condições de tráfego evoluem; e a avaliação do agente sob múltiplos cenários e padrões de demanda, a fim de medir a generalização da política aprendida. Em conjunto, esses avanços tornariam a aplicação prática da abordagem proposta mais realista, possibilitando um impacto real na saúde da população.

Bibliografia

[1] JARAMILLO, P. et al. Transport. In: SHUKLA, P. et al. (Ed.). *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, UK and New York, NY, USA: Cambridge University Press, 2022. Chapter 10.

[2] Centraal Bureau voor de Statistiek (CBS). *Emissions to air on Dutch territory; road traffic*. 2025. Dados estatísticos. Disponível em: `https://www.cbs.nl/en-gb/figures/detail/85347ENG`. Accesso em: 2025-12-05.

[3] INSTITUTE, H. E. *State of Global Air Report 2025*. [S.l.], 2025. Disponível em: <https://www.stateofglobalair.org/resources/report/state-global-air-report-2025>.

[4] OYEDELE, O. Carbon dioxide emission and health outcomes: is there really a nexus for the nigerian case? *Environmental Science and Pollution Research*, v. 29, n. 37, p. 56309–56322, August 2022. ISSN 1614-7499. Disponível em: <https://doi.org/10.1007/s11356-022-19365-x>.

[5] LOPEZ, P. A. et al. Microscopic traffic simulation using sumo. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2018. p. 2575–2582. Disponível em: <https://elib.dlr.de/127994/>.

[6] LAI, C. S. et al. A review of technical standards for smart cities. *Clean Technologies*, v. 2, n. 3, p. 290–310, 2020. ISSN 2571-8797. Disponível em: <https://www.mdpi.com/2571-8797/2/3/19>.

[7] QONITA, M.; GIYARSIH, S. R. Smart city assessment using the boyd cohen smart city wheel in salatiga, indonesia. *GeoJournal*, v. 88, n. 1, p. 479–492, feb 2023. ISSN 1572-9893. Disponível em: <https://doi.org/10.1007/s10708-022-10614-7>.

[8] HARRISON, C. et al. Foundations for smarter cities. *IBM Journal of Research and Development*, v. 54, n. 4, p. 1–16, 2010.

[9] GIONA, R. et al. Governança da cibersegurança e privacidade dos dados sob a perspectiva das cidades inteligentes. In: *Minicurso Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) 2025*. [S.l.]: Sociedade Brasileira de Computação, 2025.

[10] MARTINS, L. D.; CAMARGO, E. T. de; FONSECA, K. V. O. Smart sensors for smart environment. In: _____. *Smart Cities for Inclusive Innovation*. [s.n.]. cap. Chapter 5, p. 67–89. Disponível em: <https://digital-library.theiet.org/doi/abs/10.1049/PBBE005E_ch5>.

[11] TAHMASSEBY, S. The implementation of smart mobility for smart cities: A case study in qatar. *Civil Engineering Journal*, v. 8, n. 10, p. 2154–2171, Oct. 2022. Disponível em: <https://www.civilejournal.org/index.php/cej/article/view/3569>.

[12] PESTEL, N.; WOZNY, F. Health effects of low emission zones: Evidence from german hospitals. *Journal of Environmental Economics and Management*, v. 109, p. 102512, 2021. ISSN 0095-0696. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0095069621000802>.

[13] OBSERVATORY, A. M. *The evolution of Low Emission Zones (LEZ) in Europe: Impacts and perspectives*. `https://www.arval.com/amo/the-evolution-of-low-emission-zones-lez-in-europe-impacts-and-perspectives-amo`. Accesso em: 2025-12-03.

[14] GASTALDI, M.; CECCATO, R.; ROSSI, R. Analysis of factors driving the acceptability of a low emission zone. *Open Transportation Journal*, v. 18, 2024. Cited by: 0; All Open Access, Gold Open Access. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-105007429734doi=10.2174

[15] MARGARYAN, S. Low emission zones and population health. *Journal of Health Economics*, v. 76, p. 102402, 2021. ISSN 0167-6296. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167629620310481>.

[16] MANGLANO-REDONDO, P.; PARICIO-GARCIA, A.; LOPEZ-CARMONA, M. A. Dynamic low-emission zones for urban mobility: A systematic review. *Applied Sciences*, v. 15, n. 6, 2025. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/15/6/2915>.

[17] SUTTON, R. S.; BARTO, A. G. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. Disponível em: <http://incompleteideas.net/book/the-book-2nd.html>.

[18] BELCIC, I.; STRYKER, C. *O que é ajuste de hiperparâmetros?* 2024. Acesso em: 2025-12-05. Disponível em: <https://www.ibm.com/br-pt/think/topics/hyperparameter-tuning>.

[19] MNIH, V. et al. *Playing Atari with Deep Reinforcement Learning*. 2013. Disponível em: <https://arxiv.org/abs/1312.5602>.

[20] MNIH, V. et al. Human-level control through deep reinforcement learning. *Nature*, Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., v. 518, n. 7540, p. 529–533, fev. 2015. ISSN 00280836. Disponível em: <http://dx.doi.org/10.1038/nature14236>.

[21] IQBAL, A. et al. Mitigating catastrophic overfitting in fast adversarial training via dynamic polyak weight averaging and gradient norm dual-penalty. *Neurocomputing*, v. 678, p. 133149, 2026. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231226005461>.

[22] HUBER, P. J. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 35, n. 1, p. 73–101, mar. 1964. Disponível em: <https://doi.org/10.1214

[23] KRAUSS, S. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. Tese (Doutorado) — Universität zu Köln, Cologne, Germany, 1998. DLR-Forschungsbericht 98-08.

[24] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org* . 2017. `https://www.openstreetmap.org`.