

Equipa SmartTLS

Desafio #4 Aveiro City Hack - Relatório

Título do Projeto: Inteligência Artificial na Gestão de Tráfego Rodoviário

1. Introdução

O projeto **SmartTLS** (Smart Traffic Light System) foi desenvolvido com vista a otimizar o tráfego nas interseções semaforizadas no centro da cidade de Aveiro. Estas interseções são monitorizadas pela atual infraestrutura do Aveiro Tech City Living Lab, através dos seguintes *smart lamp posts*: P25 - Escola da Glória, P30 - Dr. Mário Sacramento I e P33 - Esquina dos Bombeiros Velhos.

Utilizando técnicas de **Inteligência Artificial** e **Multi Agent Reinforcement Learning (MARL)**, o sistema simula e otimiza a gestão do tráfego, adaptando-se às condições em tempo real. Para isso, integra dados fornecidos pelos radares do **Aveiro Tech City Living Lab**, permitindo simulações realistas e ajustadas ao ambiente urbano.

Este projeto visa não apenas melhorar a fluidez do tráfego, mas também adaptar-se à infraestrutura existente, com câmaras e radares já instalados, tendo a capacidade de escalar facilmente. Com mais sensores e dados, o sistema pode obter resultados ainda melhores sem necessidade de alterações no algoritmo, aproveitando ao máximo os recursos disponíveis.

2. Objetivos

Os objetivos centrais do projeto são:

- Reduzir as emissões de CO₂ na cidade;
- Priorizar a circulação fluída dos transportes públicos, melhorando, consequentemente, a respetiva pontualidade;
- Diminuir os tempos de espera dos restantes veículos nas interseções semaforizadas;
- Integrar o sistema com as câmaras e sensores de tráfego já disponíveis, com a possibilidade de expansão para outras interseções semaforizadas.

3. Estratégia

3.1 Ferramentas e Tecnologias Utilizadas

- **Linguagem de Programação:** Python
- **Bibliotecas:**
 - Reinforcement Learning: Stable Baselines3
 - Algoritmo: PPO MlpPolicy (Proximal Policy Optimization)
 - Simulação de tráfego: SUMO (Simulation of Urban MObility)
 - Visualização de dados: Matplotlib, Pandas
 - Integração com o Aveiro Tech City Living Lab
- **Infraestrutura:** Dados e suporte técnico fornecidos pelo Aveiro Tech City Living Lab, incluindo sensores e dados de tráfego em tempo real.

3.2 Algoritmo Utilizado

A implementação do projeto **SmartTLS** consiste na utilização de um algoritmo de **Reinforcement Learning**, em que o agente é responsável por uma interseção semaforizada, que prioriza a circulação dos transportes públicos e promove a redução do tempo de espera dos restantes veículos. De forma aos vários agentes cooperarem entre si de forma a otimizar a gestão global do tráfego rodoviário, utilizou-se **Multi Agent Reinforcement Learning**.

Estrutura do Algoritmo

- **Agentes:** Cada semáforo é um agente autónomo que toma decisões localmente. As decisões são coordenadas de forma distribuída entre os diferentes agentes, com troca mínima de informações entre os semáforos vizinhos.
- **Estados:** Os estados observados pelo algoritmo representam as condições atuais do tráfego. No código, os estados incluem variáveis como:
 - **Número de veículos em cada via compreendida pelo radar;**
 - **Tempo de espera acumulado por veículo;**
 - **Velocidade média dos veículos em cada via compreendida pelo radar.**

Estes estados fornecem ao agente as informações necessárias para tomar decisões sobre a mudança de fase dos semáforos.

Detalhes:

- **queue_weight** (peso da fila): Um cálculo que retorna o peso de uma fila, atribuindo mais peso a veículos maiores, como autocarros (peso 5) ou veículos prioritários (peso 10), em relação a carros normais (peso 1). Este peso é parametrizável.
- **current_phase** (fase atual do semáforo): Indica o estado atual do semáforo (vermelho, verde, ou amarelo).

- **action_available** (disponibilidade de ação): Determina se o semáforo pode tomar uma ação naquele momento. Após uma transição de fase, o semáforo entra num período bloqueante, definido pela fórmula:

$$self.lock_time = yellow_time + min_phase_time$$

Nesta fórmula, o *yellow_time* é o tempo de transição do amarelo, e o *min_phase_time* é o tempo mínimo da fase verde (por padrão, 5 segundos).

- **Ações:** As ações correspondem às possíveis mudanças de fase dos semáforos (de vermelho para verde, de amarelo para vermelho, etc.). Os agentes tomam estas ações com base no estado e nas observações de tráfego.

3.3 Função de Recompensa

A função de recompensa é essencial no processo de treino do modelo, refletindo o que o agente prioriza nas suas ações. A recompensa é calculada com base na diferença do **tempo de espera acumulado** entre o instante anterior e o atual, recompensando os agentes quando o tempo de espera é reduzido:

- **Recompensa:** $reward_n = d_n - d_{\{n-1\}}$,
- onde d_n é o tempo de espera acumulado por todos os veículos naquela faixa no instante atual. A recompensa será positiva quando o tempo de espera diminuir e negativa quando aumentar.

A função de recompensa também:

- **Reduz o tempo de espera acumulado.**
- **Diminui as emissões de CO₂**, como consequência do ponto anterior
- **Prioriza veículos prioritários** (como transportes públicos ou de emergência) ao ponderar mais fortemente o peso na fila (**queue_weight**).

3.4 Necessidades de Infraestrutura Adicional

Atualmente, utilizamos câmaras e sensores disponibilizados pelo **Aveiro Tech City Living Lab** para monitorar o tráfego. No entanto, de forma a melhorar a eficiência e obter ainda melhores resultados, sem alterar o algoritmo implementado, algumas melhorias poderiam ser ponderadas na infraestrutura, tais como a utilização de:

- **Câmaras 360°:** Estas seriam úteis em interseções complexas, permitindo uma visão completa de todas as direções e sentidos do tráfego (e não apenas de 1 fila, na direção e sentido do semáforo, como atualmente é feito);
- **Câmaras para cada via da interseção:** Para cada via na interseção, seria ideal ter câmaras posicionadas estrategicamente que cubram todos os veículos;
- **Sensores adicionais:** Sensores de velocidade e contagem de veículos mais precisos em cada faixa ajudariam a calibrar o sistema com maior precisão.

Com estas melhorias na infraestrutura, conseguiríamos capturar dados mais precisos sobre a posição, velocidade e número de veículos na faixa de rodagem, sem alterar a solução apresentada.

3.5 Dados Utilizados

Os dados fornecidos pelas câmaras e sensores existentes são essenciais para o sistema de semáforos inteligentes. Esses dados incluem:

- **Posição e velocidade dos veículos:** Dados em tempo real sobre as posições e velocidades dos veículos, que são usados para estimar o tempo de espera e a contagem do número de veículos em cada faixa de rodagem.
 - **Posição dos transportes públicos:** Detecção de transportes públicos e de emergência, bem como as suas posições, garantindo que recebem prioridade nas interseções.

Estes dados são processados em tempo real, permitindo que os agentes ajustem as suas decisões de forma dinâmica.

Como já foi referido, para começarmos a desenvolver a ideia com casos mais simples e perceptíveis, optámos por utilizar uma ferramenta de simulação - SUMO. Mas, apesar de ser simulado, não deixámos de usar os dados que também seria possível obter através da API do Aveiro Tech City Living Lab.

Para implementar o agente nos semáforos da cidade (reais), teríamos de, a partir da API fornecida, extrair as entradas para o nosso modelo:

- número de veículos por fila (valorizando mais os autocarros);
- tempo de espera acumulativo dos veículos (em 60m para lá do semáforo);
- posições dos autocarros na cidade a cada momento.

Com isto, de modo a demonstrar que, de facto, conseguiríamos implementar este serviço com os dados reais fornecidos, criámos pequenos scripts (diretório **/real_data_processors**) que extraem da API os dados necessários para o modelo.

Além dos scripts, são também disponibilizados ficheiros *.json* com dados para testes (destinados a serem passados como argumento aos scripts), por sua vez localizados na pasta **/real_data_processors/tests**: *test_real_data.json*, com os dados reais fornecidos pela API; *test_simple_data1.json* e *test_simple_data2.json*, que possuem ambos dados mais simples e perceptíveis (para garantir a validade da solução), com campos filtrados, constando apenas aqueles que realmente são utilizados pelos scripts para obtenção dos dados; *test_bus_locator.json*, que utiliza os dados da API para localizar a cada momento os autocarros (veículo prioritário) conectados à rede.

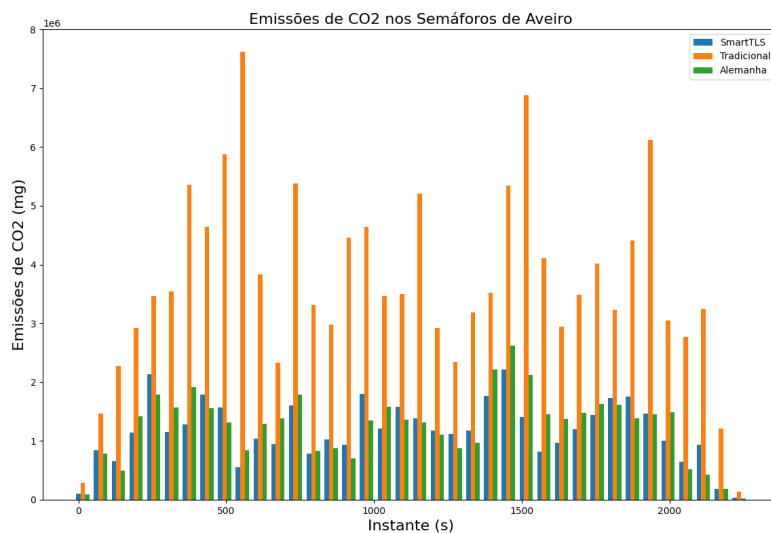
4. Resultados

Treinou-se o modelo em diferentes cenários de tráfego: baixo, médio, alto e em situações extremas como acidentes.

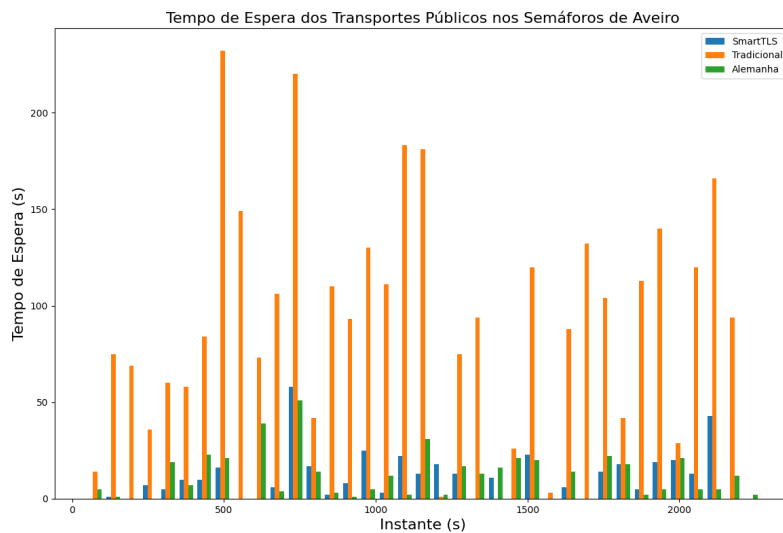
De forma a avaliarmos o desempenho do nosso modelo baseado em orquestração global com Inteligência Artificial, comparou-se com outros dois modelos: **modelo Tradicional**, utilizado atualmente em Aveiro, baseado em tempos fixos e um **modelo gap-based**, implementado na Alemanha.

Nas secções seguintes apresentamos os resultados obtidos num cenário de tráfego médio. Os resultados obtidos nos diferentes cenários, que se encontram na pasta **/docs** do repositório, poderão ser confirmados por quem quiser testar a implementação localmente.

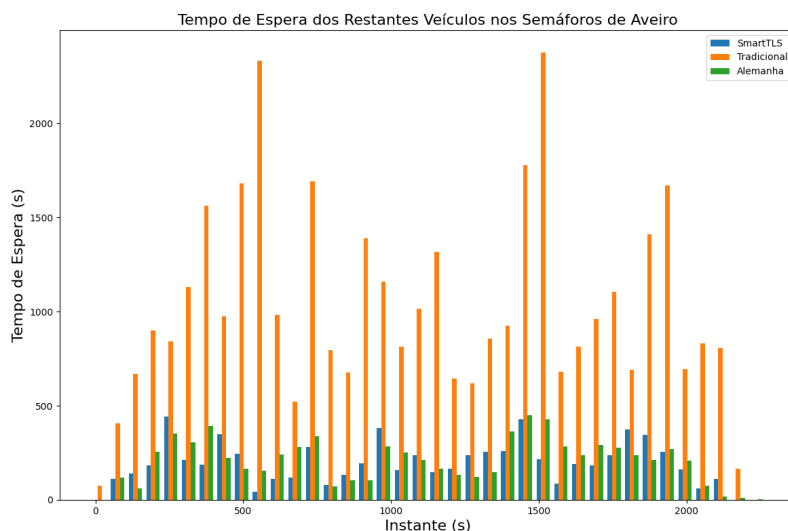
4.1 Redução da Emissão de CO₂



4.2 Redução do Tempo de Espera dos Transportes Públicos



4.3 Redução do Tempo de Espera dos Restantes Veículos



4.4 Análise dos Resultados

- O modelo **Tradicional** apresentou o pior desempenho, tanto em termos de emissões de CO₂ quanto de tempo de espera dos transportes públicos e restantes veículos.
- O modelo da **Alemanha** trouxe uma melhoria notável relativamente ao **Tradicional**, e, em termos de resultados, equipara-se ao modelo **SmartTLS**, porém é inferior ao nosso modelo.
- O modelo **SmartTLS** mostrou ser o mais otimizado, evidenciando que o modelo **Tradicional** tem um aumento de **418%** no tempo de espera, **731%** no tempo de espera de transportes públicos e **213%** de emissões de CO₂. Além disto, mostra também, que o modelo da **Alemanha** tem um aumento de **7%** no tempo de espera, **7%** no tempo de espera de transportes públicos e **6%** de emissões de CO₂, o que valida o uso da solução proposta.

5. Conclusão

Os resultados mostram uma grande redução na emissão de CO₂ e na redução do tempo de espera dos transportes públicos e restantes veículos. O uso de **Reinforcement Learning**, especialmente **Multi Agent Reinforcement Learning**, oferece uma solução dinâmica que responde em tempo real às condições do tráfego, reduzindo os tempos de espera dos veículos e melhorando a qualidade do ar na cidade.