Explorando o Impacto do Conjunto de Ações em Semáforos Inteligentes Baseados em Aprendizado por Reforço

João Pedro Fernandes Silva
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Brasil
joao.fernandes@dcc.ufmg.br

Guilherme Maia de Menezes
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Brasil
jgmm@dcc.ufmg.br

***Resumo*—O controle semafórico eficiente é um dos problemas mais explorados em computação urbana, e abordagens baseadas em aprendizado por reforço profundo (DRL) têm apresentado resultados promissores. No entanto, diferentes trabalhos da literatura adotam escolhas de projeto muito distintas entre si, e escolhas como a representação de estado, função de recompensa, algoritmo e o conjunto de ações impactam muito no comportamento final do agente, de modo que não se sabe qual dessas escolhas de fato determina o desempenho. Neste trabalho, foi investigado especificamente como a escolha do conjunto de ações impacta no comportamento do agente, comparando de forma controlada dois paradigmas centrais da literatura: a seleção livre de fase e o controle cíclico com duração. O efeito da ação foi analisado isoladamente, com os experimentos conduzidos em dois cenários realísticos das cidades alemãs de Cologne e Ingolstadt, por meio de simulações no SUMO, dirigidas utilizando o RESCO. A comparação o desempenho dos agentes a um semáforo estático em diferentes condições de operação da via mostrou diferenças de até 80% no desempenho do agente, variando apenas a ação permitida pelo agente.**

***Index Terms*—Controle semafórico, Aprendizado por reforço, Aprendizado profundo, Sistemas de Transporte Inteligente.**

I. Introdução

O processo de urbanização, atrelado ao crescimento populacional, traz consigo a deterioração da mobilidade em cidades ao redor do mundo. Neste contexto, o controle semafórico (TLC, do inglês *Traffic Light Control*) eficiente se torna uma tarefa de grande interesse, uma vez que têm impacto significativo sobre o fluxo de veículos aliviando o tráfego próximo à interseções [16].

Diferentes abordagens são propostas para atacar o problema de controle de semáforos, como a utilização de técnicas tradicionais de otimização [4], [11], soluções baseadas em algoritmos genéticos [19] e o emprego de lógica fuzzy [1], [8]. No entanto, muitos métodos sofrem com problemas de escalabilidade devido ao alto custo computacional. Assim, abordagens baseadas no arcabouço de aprendizado por reforço(RL, do inglês *Reinforcement Learning*) surgem como uma alternativa viável para o desenvolvimento de sistemas robustos. A alta adaptabilidade deste tipo de modelo, podendo responder rapidamente às demandas de tráfego, além da possibilidade de aprendizado contínuo de forma *online* [15]. Especialmente, com o sucesso recente de técnicas de aprendizado profundo nas mais diversas aplicações, diferentes abordagens de controle semafórico utilizando DRL (do inglês *Deep Reinforcement Learning*) foram propostas [6], [7], [12], [17], [23].

O aprendizado profundo no problema de TLC se destaca por sua capacidade de extrair representações complexas de grandes espaços de estado, como padrões de tráfego em tempo real. Aravés de redes neurais profundas (DNNs, do inglês *Deep Neural Networks*), agentes de DRL conseguem processar entradas de diferentes fontes, como sensores veiculares, câmeras de monitoramento e sinais V2X (do inglês *Vehicle-to-Everything*). Essa flexibilidade em relação à origem dos dados é de extrema importância para sistemas de gerenciamento de tráfego, onde diferentes fontes de informação são utilizadas [20]. Outro aspecto relevante é a capacidade desses modelos de se generalizarem para diferentes configurações de interseções e escalas urbanas, sendo uma solução promissora para redes de trânsito em larga escala [15].

Dentre as diversas soluções propostas, dois exemplos ilustram bem as diferentes formas de se definir o conjunto de ações de um agente semafórico. Vidali et al. [23] adotam a seleção livre de fase: a cada decisão, o agente escolhe livremente qual das fases semafóricas ativar, em qualquer ordem, com duração de verde fixa. Já Cai e Wei [3] adotam o controle cíclico com duração: as fases seguem uma ordem cíclica fixa e o agente decide apenas a duração do verde de cada fase. Ambos os trabalhos empregam aprendizado por reforço profundo, mas partem de premissas distintas sobre qual aspecto do semáforo o agente deve controlar.

A diferença entre esses dois conjuntos de ações não é apenas técnica. Cai e Wei [3] argumentam explicitamente que a seleção livre de fase, embora possa melhorar a eficiência do tráfego, pode alterar a sequência original das fases e, com isso, comprometer a segurança dos motoristas, e por esse motivo, optam por restringir o agente à escolha da duração do verde. Apesar dessa justificativa, ainda há uma lacuna na literatura quanto à compreensão detalhada de como a escolha

do conjunto de ações afeta, de fato, o comportamento e o desempenho do agente. Este trabalho busca explorar exatamente essa questão: isolar e comparar o impacto da escolha do conjunto de ações sobre o desempenho do agente. Para tanto, foi mantida a rede neural, a representação de estado, a função de recompensa e os cenários de avaliação, variando unicamente o paradigma de ação (seleção livre de fase *versus* cíclico com duração), em uma comparação controlada conduzida sobre cenários realísticos por meio do arcabouço de avaliação RESCO [2].

Assim, este estudo está organizado da seguinte maneira: na Seção II são apresentados os trabalhos relacionados na área; na Seção III é descrito o agente de DRL implementado, incluindo as técnicas utilizadas para aprimorar a rede neural e os dois conjuntos de ações comparados; na Seção IV são descritos o arcabouço RESCO, os cenários utilizados, a caracterização dos cenários e os experimentos realizados, juntamente com os resultados alcançados; por fim, na Seção VI estão contidas as conclusões e diretrizes para trabalhos futuros.

II. Trabalhos relacionados

Genders e Razavi (2016) [7] propuseram um sistema inovador de controle semafórico baseado em DRL denominado Deep Q-Network Traffic Signal Control Agent (DQTSCA). O modelo utiliza redes neurais convolucionais (CNNs, do inglês *Convolutional Neural Networks*) para otimizar a política de controle semafórico em interseções isoladas. O estado do tráfego é modelado por meio de uma nova representação, chamada de Discrete Traffic State Encoding (DTSE), que combina informações sobre a presença de veículos, suas velocidades normalizadas e a fase semafórica atual. Essa representação busca maximizar a densidade de informações úteis, permitindo ao modelo capturar características relevantes do ambiente. O treinamento foi realizado no simulador SUMO com *experience replay*, evidenciando melhorias substanciais frente a uma rede neural rasa: redução de 82% no atraso cumulativo médio, 66% no comprimento médio das filas e 20% no tempo médio de viagem.

Mousavi et al. (2017) [17] apresentaram uma abordagem para o controle de semáforos utilizando DRL com dois agentes distintos: um baseado em gradiente de política (*policy-gradient*) e outro baseado na função de valor. Ambos os métodos processam imagens capturadas diretamente de simulações no SUMO por meio de uma CNN, eliminando a necessidade de pré-processamento manual. A função de recompensa foi definida como a diferença no atraso acumulado entre passos consecutivos. Os experimentos mostraram que ambas as abordagens superam significativamente os modelos de controle baseados em redes rasas e sistemas de controle semafórico fixo, reduzindo o atraso acumulado médio em até 68% e o comprimento das filas em até 73%.

Liu et al. (2023) [12] investigaram o uso de DRL para o controle de semáforos em sistemas inteligentes de transporte (ITS), aplicando o DeepQLearning (DQN) a interseções isoladas e o *Deep Deterministic Policy Gradient* (DDPG) a redes viárias em grade. Para o cenário de interseção isolada, os autores demonstraram que o DQN foi capaz de produzir uma política que coincide com a solução ótima calculada por métodos tradicionais de Processos de Decisão de Markov (MDP), enquanto o DDPG destacou-se em cenários multi-agente ao gerar o comportamento inteligente da *greenwave* (onda verde). Os resultados evidenciam o potencial do DRL para resolver problemas complexos de controle semafórico em tempo real, especialmente em cenários com múltiplos agentes.

Vidali et al. (2019) [23] propuseram um agente de controle semafórico baseado em DeepQLearning para interseções isoladas. O estado é representado por um vetor binário que discretiza cada via de aproximação em células, indicando apenas a presença ou ausência de veículos, sem informação de velocidade ou de fase. O espaço de ações é de seleção livre de fase: a cada decisão, o agente escolhe livremente qual das quatro fases semafóricas ativar, com duração de verde fixa. A recompensa é a variação do tempo de espera acumulado dos veículos entre decisões. Os experimentos, conduzidos em cenário sintético no SUMO, mostraram ganhos expressivos frente a um semáforo estático.

Cai e Wei (2024) [3] propuseram o PN_D3QN, um agente que combina múltiplas técnicas modernas de aprendizado profundo. O estado é composto por três blocos: uma matriz de posição (presença de veículos por célula), uma matriz de velocidade e a fase atual do semáforo, processados por uma rede convolucional. O espaço de ações é cíclico com duração: as fases seguem uma ordem fixa e o agente escolhe apenas a duração do verde de cada fase, dentro de um conjunto discreto de valores. A recompensa é a variação do comprimento das filas. O algoritmo incorpora as técnicas *Dueling Network*, *Double Q-Learning*, *Prioritized Experience Replay* e *Noisy Networks*, alcançando convergência mais rápida e melhor desempenho que variantes mais simples. Justamente esses dois paradigmas de ação (seleção livre de fase e cíclico com duração) constituem o eixo de comparação deste trabalho.

III. Agente de DRL

Um problema de RL pode ser formalizado como um processo de decisão de Markov (*Markov decision process*, MDP), definido por um conjunto de estados, ações e recompensas.

- **Espaço de estados** $\mathcal{S}$: Representa a percepção do ambiente pelo agente, ou seja, o estado atual da interseção. Cada estado s_t no tempo t pode ser representado por um vetor $s_t \in \mathcal{S}$ que encapsula informações relevantes sobre a interseção, como a densidade de veículos, o comprimento da fila de veículos, a ocupação das faixas e a fase atual do semáforo.
- **Espaço de ações** $\mathcal{A}$: Representa o conjunto de ações possíveis que o agente pode tomar para controlar o semáforo. Cada ação $a_t \in \mathcal{A}$ pode corresponder a uma combinação de luzes do semáforo ou a uma mudança no tempo de duração de cada fase do semáforo.
- **Função de transição** $P(s_{t+1}|s_t, a_t)$: A função de transição descreve a probabilidade de o agente transitar para o estado s_{t+1} no tempo $t+1$, dado o estado atual s_t e a ação a_t tomada pelo agente no tempo t.

- **Função de recompensa** r_{t+1}: A função de recompensa r_{t+1} fornece ao agente uma medida de quão boa foi a ação a_t em relação ao desempenho desejado, após a transição para o estado s_{t+1}. A recompensa pode ser baseada em indicadores de eficiência, como o tempo de espera dos veículos, o comprimento das filas ou o fluxo de tráfego.
- **Política** π: A política $\pi(a_t|s_t)$ é uma função que mapeia o estado s_t para uma distribuição de probabilidade sobre as ações possíveis a_t. A política π^* é a política ótima que maximiza a recompensa acumulada ao longo do tempo.

O objetivo final do agente é aprender uma política π^* que maximize a soma das recompensas esperadas ao longo do tempo. Isso é formalizado pela função de valor $V^\pi(s)$, que representa a recompensa esperada a partir de um estado s seguindo a política π:

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s\right],$$

onde γ é o fator de desconto que determina a importância de recompensas futuras em relação às recompensas imediatas. A política ótima π^* é aquela que maximiza a função de valor, $\pi^*(s) = \arg\max_\pi V^\pi(s)$, e pode ser calculada iterativamente por meio da equação de Bellman.

A. Definição do agente implementado

Para responder à pergunta central deste trabalho, foi adotado um agente seguindo o projeto de Cai e Wei [3]. Esse agente base combina uma representação de estado baseada na posição, na velocidade e na fase dos veículos; uma recompensa dada pela variação do comprimento das filas; e uma rede neural que integra camadas convolucionais e totalmente conectadas, treinada com o algoritmo D3QN (que reúne as técnicas *Dueling Network*, *Double Q-Learning*, *Prioritized Experience Replay* e *Noisy Networks*). O único elemento que varia entre os modelos é o conjunto de ações, que constitui o objeto de comparação deste estudo. Cada um desses componentes é detalhado nas subseções a seguir.

1) Representação do estado: O estado é representado por meio de codificação discreta do estado do tráfego (DTSE, do inglês *Discrete Traffic State Encoding*), na qual cada via de aproximação é dividida em células de tamanho fixo (aproximadamente 7 m cada, cobrindo cerca de 147 m a partir da interseção). O estado é composto por três componentes:

- **Matriz de posição:** matriz binária em que cada célula recebe o valor 1 se houver um veículo naquela posição da via, e 0 caso contrário.
- **Matriz de velocidade:** matriz de mesma dimensão que a de posição, em que cada célula contém a velocidade do veículo ali presente.
- **Fase atual:** vetor que codifica, de forma *one-hot*, qual é a fase semafórica ativa no momento.

Essa representação é mais rica que o vetor puramente binário de presença utilizado por Vidali et al. [23], pois adiciona as informações de velocidade e de fase, permitindo que a rede convolucional extraia padrões espaciais mais informativos do tráfego.

2) Função de recompensa: A recompensa adotada mede a variação do comprimento total das filas entre duas decisões consecutivas, conforme proposto por Cai e Wei [3]:

$$r_t = -\left(\sum_{i=1}^{N} q_{t+1}^i - \sum_{i=1}^{N} q_t^i\right),$$

onde q_t^i é o comprimento da fila da via i no instante de decisão t e N é o número de vias. Assim, $r_t > 0$ quando as filas diminuem após a ação (melhora do tráfego) e $r_t < 0$ caso contrário. O comprimento de fila é escolhido como indicador por ser possível de obter tempo real a partir de detectores nas vias, ao contrário do tempo de espera, que exige o rastreamento individual de cada veículo.

3) Conjunto de ações: O conjunto de ações é o elemento variado neste estudo. Foram comparados os os dois paradigmas propostos pelos trabalhos de referência, ilustrados na Figura 1.

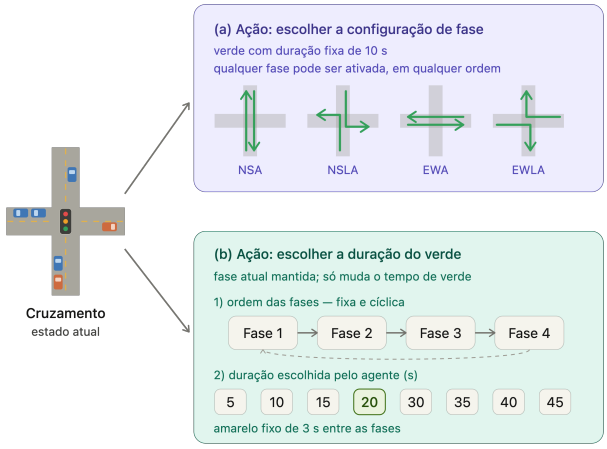


Figura 1: Paradigmas de ação comparados neste trabalho (a) Seleção livre de fase (Vidali) e (b) Cíclico com duração (Cai)

a) Seleção livre de fase: O conjunto de ações corresponde às configurações semafóricas disponíveis para o cruzamento:

$$\mathcal{A} = \{NSA, NSLA, EWA, EWLA\}$$

A cada decisão, o agente escolhe livremente qual dessas fases ativar, em qualquer ordem, com duração de verde fixa, conforme ilustrado na Figura 1(a).

b) Cíclico com duração dinâmica: As fases seguem uma ordem cíclica fixa e o agente escolhe apenas a duração do verde da fase corrente, dentro do conjunto discreto, conforme a Figura 1(b):

$$\mathcal{A} = \{5, 10, 15, 20, 30, 35, 40, 45\} \text{ s}$$

Após o verde de cada fase, há um período de amarelo de 3 s para garantir a segurança do tráfego. Nesse paradigma,

a ordem de atendimento das aproximações é garantida por construção; a decisão do agente recai apenas sobre quanto tempo cada fase permanece verde.

Em ambos os casos, o estado, a função de recompensa, a rede neural e os cenários são mantidos idênticos, de modo que a comparação isole o efeito do conjunto de ações.

4) Algoritmo de DRL: O agente utiliza o DeepQLearning, uma variante do algoritmo QLearning [26] que emprega redes neurais para estimar os valores de recompensa esperada (valores Q) de cada ação. Dado um estado s_t, a rede calcula um valor $Q(s_t, a_t)$ para cada ação possível, e o agente seleciona a ação de maior valor. A atualização dos valores Q segue a equação:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big(r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\Big),$$

onde α é a taxa de aprendizado e γ o fator de desconto.

Assim, foi adotado o conjunto de aprimoramentos proposto por Cai e Wei [3], denominado D3QN, que combina uma rede convolucional para extração de características a técnicas modernas de estabilização e exploração do aprendizado profundo. Essas técnicas são detalhadas a seguir de forma introdutória.

B. Técnicas de aprimoramento da rede neural

1) Rede convolucional (CNN): Como o estado é representado por matrizes (de posição e de velocidade), semelhantes a imagens, utiliza-se uma rede neural convolucional para extrair automaticamente padrões espaciais do tráfego, como aglomerados de veículos ou filas se formando em determinada via. A CNN dispensa a engenharia manual de características e produz um vetor compacto que resume o estado da interseção, o qual é então concatenado à informação de fase e processado por camadas totalmente conectadas.

2) Arquitetura Dueling: A arquitetura *Dueling* [24] divide a estimativa do valor $Q(s, a)$ em duas partes: (i) o valor do estado $V(s)$, que mede o quão bom é estar em determinado estado independentemente da ação escolhida; e (ii) a vantagem $A(s, a)$, que mede o quanto cada ação é melhor ou pior que a média naquele estado. As duas são recombinadas para produzir $Q(s, a)$. Essa separação permite que o agente aprenda quais estados são valiosos mesmo quando a escolha da ação tem pouco impacto, acelerando o aprendizado.

3) Double Q-Learning: O DeepQLearning tradicional tende a superestimar os valores Q, pois usa a mesma rede tanto para escolher quanto para avaliar a melhor ação. O *Double Q-Learning* [22] corrige esse viés utilizando duas redes: uma rede principal seleciona a ação considerada melhor, e uma segunda rede (a rede-alvo) avalia o valor dessa ação. Essa separação reduz a superestimação e torna o treinamento mais estável.

4) Repetição priorizada de experiências (Prioritized Experience Replay): Durante o treinamento, o agente armazena suas experiências passadas em uma memória e as reutiliza para aprender. No DQN tradicional, essas experiências são sorteadas de maneira uniforme. A repetição priorizada de experiências [18] atribui a cada experiência um peso proporcional ao seu "erro" de predição, sorteando com mais frequência as experiências mais surpreendentes ou informativas, tipicamente os raros momentos de congestionamento. Isso acelera a convergência ao concentrar o aprendizado nos eventos mais relevantes.

5) Redes ruidosas (Noisy Networks): Para explorar novas ações, o DQN tradicional costuma agir de forma aleatória com certa probabilidade (estratégia ϵ-*greedy*). As redes ruidosas [5] substituem essa aleatoriedade externa por um ruído aprendível adicionado diretamente aos pesos da rede. Assim, a exploração passa a ser adaptativa: o próprio agente aprende quanto e onde explorar, resultando em uma busca mais eficiente pela política ótima.

6) Atualização suave da rede-alvo: Por fim, a rede-alvo utilizada na avaliação das ações é atualizada de forma suave, isto é, seus parâmetros se aproximam gradualmente dos da rede principal a cada passo, controlados por um fator τ. Essa atualização gradual, em vez de cópias abruptas, contribui para a estabilidade do treinamento.

IV. Experimentos

A. O arcabouço RESCO

Os experimentos foram conduzidos utilizando o RESCO (*Reinforcement Learning Benchmarks for Traffic Signal Control*) [2], um arcabouço de avaliação para controle semafórico com aprendizado por reforço construído sobre o simulador SUMO [14]. O RESCO padroniza cenários, representações de estado, funções de recompensa, algoritmos e métricas, o que o torna especialmente adequado para comparações controladas. Além disso, a plataforma disponibiliza cenários de interseções reais, com demanda de tráfego derivada de dados realísticos, e agentes de referência (como um semáforo estático e políticas gulosas), utilizados como *baselines* de comparação.

B. Cenários utilizados

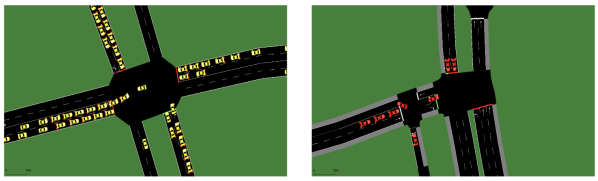
Foram utilizados dois cenários de interseção única, correspondentes a cruzamentos reais de duas cidades europeias, disponíveis no RESCO:

- ***cologne1*:** um cruzamento de quatro vias da cidade de Colônia (Alemanha), com demanda extraída do cenário TAPAS Cologne [21]. Apresenta um pico de tráfego no período da manhã.
- ***ingolstadt1*:** um cruzamento em "T" da cidade de Ingolstadt (Alemanha), com demanda extraída do cenário InTAS [13]. Apresenta um pico de tráfego no período da tarde.

Ambos os cenários utilizam a demanda real de um dia completo de tráfego (24 h) através da respectiva interseção, o que permite avaliar o agente sob condições de operação variadas e fiéis ao comportamento real da via.

C. Caracterização dos cenários

Ao longo das 24 h, o cruzamento de Colônia recebe cerca de 46,8 mil veículos, distribuídos de forma relativamente equilibrada entre as quatro aproximações, com leve predomínio



Cenário	Norte	Sul	Leste	Oeste	Total
Cologne	11.811	11.677	13.746	9.499	46.733
Ingolstadt	4.721	12.527	—	7.438	24.686

Figura 2: Topologia e demanda dos dois cruzamentos, no simulador SUMO: à esquerda, *cologne1* (Colônia), com quatro vias; à direita, *ingolstadt1* (Ingolstadt), em "T". A tabela indica o número de veículos por aproximação ao longo das 24 h (Ingolstadt não possui braço Leste).

da aproximação Leste (29%). Já o cruzamento de Ingolstadt, em "T", recebe cerca de 24,7 mil veículos, fortemente concentrados na aproximação Sul (51%). A Figura 2 apresenta a topologia dos dois cruzamentos e o número de veículos por aproximação em cada cenário.

Ao longo de um dia completo, cada interseção atravessa diferentes condições de operação: períodos de baixa demanda, formação e dissipação de filas, picos de congestionamento e transições entre esses regimes. Para caracterizar automaticamente essas condições, executou-se o controlador estático sobre as 24 h de cada cenário, registrando a ocupação de cada aproximação ao longo do tempo. Sobre essa série temporal aplicou-se o algoritmo PELT (*Pruned Exact Linear Time*) [9], uma técnica de detecção de pontos de mudança. O PELT modela a série como uma sequência de segmentos, cada um com um comportamento estatístico próprio (por exemplo, uma média de ocupação), e busca o conjunto de instantes de corte que minimiza o custo total de ajuste desses segmentos somado a uma penalidade por corte, que evita a fragmentação excessiva (um passo de poda descarta candidatos que nunca poderão ser ótimos, o que confere ao método um custo computacional linear no tamanho da série). O resultado é a partição do dia em segmentos homogêneos.

Em seguida, os segmentos são reunidos em um pequeno conjunto de condições representativas por meio do método de Ward [25], um algoritmo de agrupamento hierárquico aglomerativo. Partindo de cada segmento como um grupo isolado, o método une repetidamente o par de grupos cuja fusão provoca o menor aumento possível na variância interna total (a soma das distâncias quadráticas de cada segmento ao centro do seu grupo), até restar o número desejado de grupos. A semelhança entre os segmentos é medida por sua assinatura, composta pela distribuição direcional do fluxo entre as aproximações e pelo nível de saturação. Como o controlador estático não depende do agente, os regimes obtidos são uma propriedade do cenário, e não da política avaliada, o que garante uma base de comparação idêntica para todos os conjuntos de ações.

A Figura 3 apresenta o resultado dessa mineração para os dois cenários. Em cada bloco, o gráfico superior mostra a ocupação total ao longo do dia, com as faixas coloridas indicando os regimes de operação descobertos. O gráfico inferior detalha a ocupação por aproximação (Norte, Sul, Leste e Oeste), evidenciando como a direção dominante do fluxo se altera entre os regimes: por exemplo, o predomínio da aproximação Leste durante os picos de tráfego em *cologne1*. Na legenda, cada regime é identificado por sua direção dominante e por seu nível de saturação (*sat*), definido como a ocupação média do regime normalizada para o intervalo $[0, 1]$ entre todos os regimes do dia, de modo que 0 corresponde ao regime mais vazio e 1 ao regime mais congestionado. As regiões hachuradas assinalam as janelas de uma hora selecionadas para a avaliação: cada condição descoberta é avaliada individualmente nessas janelas, o que permite comparar o desempenho de cada paradigma de ação sob cada regime de operação.

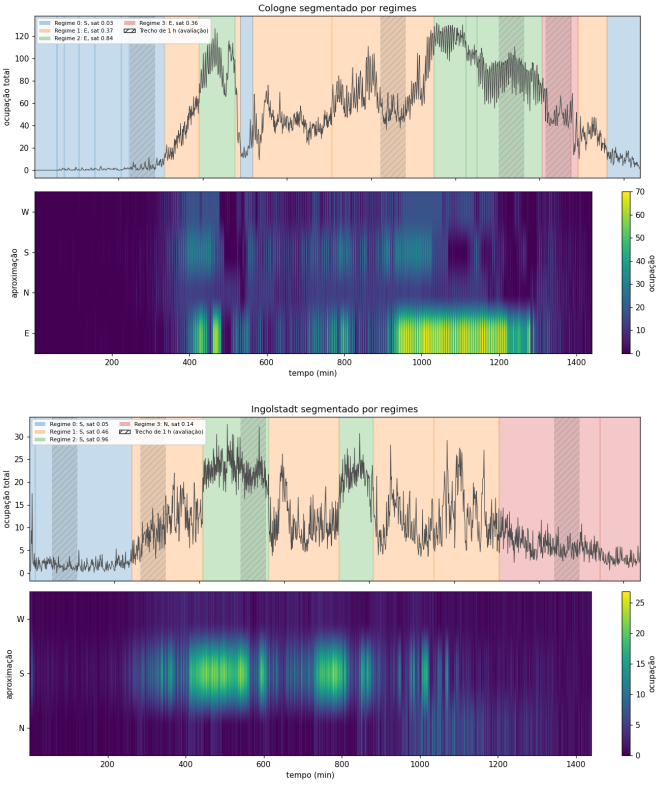


Figura 3: Regimes de operação minerados, para *cologne1* (acima) e *ingolstadt1* (abaixo). Em cada bloco: a ocupação total ao longo das 24 h, segmentada em regimes comportamentais (cada cor é um grupo de condição descoberto pelo PELT e pelo agrupamento), e a ocupação por aproximação (mapa de calor), que revela a estrutura direcional de cada regime.

O número de grupos (k) e o peso relativo entre o nível de saturação e o perfil direcional foram escolhidos por meio de uma busca em grade (*grid search*) sobre esses dois parâmetros, com a ocupação agregada em janelas de 60 s. A configuração adotada, com $k = 4$ grupos e peso maior para a saturação, foi a que produziu a separação mais coerente entre os regimes, uma vez que valores maiores de k apenas fragmentavam o

período de baixa demanda da madrugada sem revelar novos padrões. Os regimes resultantes têm interpretação direta em termos de horário e padrão de fluxo. Em Colônia, obtêm-se um regime de baixa demanda, correspondente à madrugada e ao fim da noite, com ocupação quase nula; um regime diurno de demanda moderada e bem distribuída entre as aproximações, que cobre a maior parte do dia; um regime de pico saturado, nos horários de rush da manhã (por volta das 06:30 às 08:00) e da tarde (das 16:00 às 20:00), com forte predomínio da aproximação Leste; e um breve regime de escoamento pós-pico, no início da noite, em que o fluxo se concentra quase inteiramente no Leste. O cruzamento de Ingolstadt segue um padrão semelhante, porém dominado pela aproximação Sul, com madrugada vazia, um longo período diurno moderado, picos de rush fortemente saturados pela manhã e no início da tarde e, à noite, uma inversão do fluxo dominante para a aproximação Norte.

V. Resultados

A. Treinamento

Todos os modelos compartilham o mesmo agente base descrito na Seção III. Cada modelo é treinado por 200 episódios sobre demanda sintética, gerada a partir das taxas reais de chegada por aproximação de cada cenário. A cada episódio, a demanda é reamostrada de forma estocástica (a inserção de veículos segue um processo de Poisson sobre a taxa-alvo), reproduzindo o processo de geração aleatória de Cai e Wei [3] e evitando que o agente memorize uma única realização de tráfego. Após o treinamento, os modelos são avaliados de forma congelada sobre a demanda real dos cenários.

Para investigar também o efeito da diversidade da demanda vista em treino, foram consideradas duas configurações de treinamento:

- **Treino restrito:** o agente treina apenas sobre o config típico (c1), cujas taxas por aproximação espelham a demanda real média do cruzamento.
- **Treino diverso:** a cada episódio, o agente treina sobre um dos cinco configs (de c1 a c5), alternados em rodízio, cobrindo desde demanda balanceada baixa e alta até cenários assimétricos e não estacionários.

A Tabela I descreve os cinco conjuntos de demanda sintética (c1 típico, c2 baixo balanceado, c3 alto balanceado, c4 assimétrico e c5 não estacionário). Combinando os dois cenários, as duas larguras de treino e os dois conjuntos de ações, foram treinados oito modelos no total.

A Figura 4 acompanha a fila média por episódio ao longo do treinamento, para os quatro modelos de cada cenário. A fila diminui ao longo do treino nos dois cenários (redução de 18% a 24% no cruzamento de Colônia), o que confirma que os agentes de fato aprendem.

Um aspecto que chama a atenção é a grande diferença de variância entre as curvas, que tem duas origens. A principal é o próprio paradigma de ação: as curvas do controle livre apresentam desvio-padrão da fila entre episódios de cerca de 22 a 26 veículos, contra apenas 3 a 11 do controle cíclico.

Tabela I: Configs de demanda sintética usados no treinamento: taxa de chegada (veh/h) por aproximação. Os configs de c1 a c4 são estacionários (janela única de $[0, 3600]$ s); o config c5 é não estacionário, percorrendo quatro janelas de 900 s (valores separados por "/").

Cenário	Aprox.	c1	c2	c3	c4	c5 (4×900 s)
Cologne	L	570	300	650	400	650/400/1000/300
	N	320	300	650	150	650/150/200/300
	S	690	300	650	1050	650/1050/300/300
	O	440	300	650	300	650/300/500/300
Ingolstadt	N	460	300	620	250	620/250/300/300
	S	620	300	620	500	620/500/950/300
	O	635	300	620	950	620/950/450/300

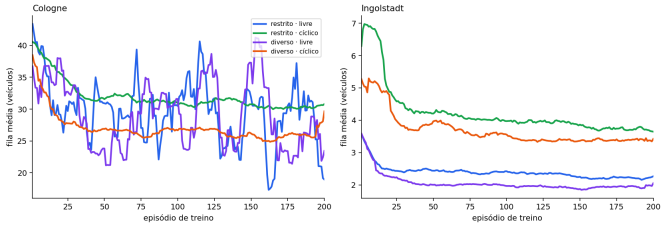


Figura 4: Fila média por episódio durante o treinamento (linhas suavizadas; ao fundo, os valores brutos por episódio), para os quatro modelos de cada cenário. A tendência de queda indica aprendizado; a variância entre episódios decorre da reamostragem estocástica da demanda.

Como essa diferença se mantém mesmo no treino restrito, em que a demanda é praticamente constante entre episódios (cerca de 2000 veículos, com variação de apenas algumas dezenas), ela reflete a instabilidade da ação livre, que pode concentrar o verde em uma única aproximação em alguns episódios e distribuí-lo de forma equilibrada em outros. A segunda origem é a diversidade da demanda: no treino diverso, o número de veículos por episódio varia amplamente (de cerca de 1100 a 2700), o que aumenta a variância da fila em relação ao treino restrito, tanto na ação livre quanto na cíclica. Em ambos os casos, essa variância é uma propriedade da ação e da demanda, e não um sinal de instabilidade do aprendizado, já que a fila média continua a diminuir ao longo do treino.

B. Avaliação em cenário realístico

Após o treinamento, cada modelo foi avaliado com a política congelada, ou seja, sem exploração e sem atualização de pesos, sobre a demanda real dos cenários. A avaliação foi feita separadamente em cada uma das condições de operação identificadas na Seção IV-C, por meio de janelas de uma hora de tráfego, comparando os modelos ao semáforo estático em duas métricas: o tempo médio de espera dos veículos e o tamanho médio das filas. As condições, identificadas por g0 a g3, correspondem aos quatro regimes minerados para cada cenário, na mesma numeração e cor com que aparecem na Figura 3: g0 é o regime de baixa demanda (madrugada), g1 o regime diurno de demanda moderada, g2 o pico saturado

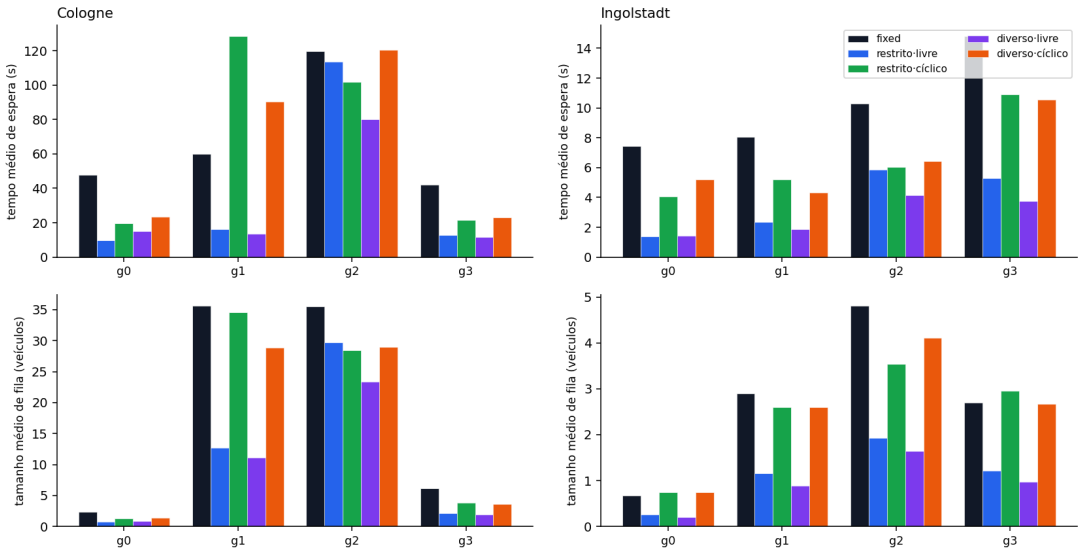


Figura 5: Desempenho por condição, para cada cenário: tempo médio de espera (acima) e tamanho médio de fila (abaixo). O semáforo estático (*fixed*) aparece em preto e os quatro modelos aprendidos, em cores.

e g3 o regime noturno (escoamento pós-pico em Colônia e inversão para o Norte em Ingolstadt). A Tabela II resume o tempo médio de espera por condição, e a Figura 5 apresenta as duas métricas.

De modo geral, os modelos aprendidos superam o semáforo estático. Em Ingolstadt, o cenário mais leve, os quatro modelos reduzem o tempo de espera em todas as condições, com ganhos entre 26% e 82%. Em Colônia os ganhos também são expressivos na maior parte das condições, mas o comportamento passa a depender do conjunto de ações, sobretudo nos regimes de maior demanda.

O agente de ação livre obtém as maiores reduções de espera na maioria das condições, como os ganhos de 79% no regime de baixa demanda e de 73% no regime moderado de Colônia, graças à sua flexibilidade para atender diretamente a aproximação mais carregada. Essa mesma flexibilidade, no entanto, é também sua fragilidade: em simulações de testes, algumas vezes pôde-se observar que sob alta demanda, escolhas livres do agente levam a alguns veículos bloquearem mutuamente o cruzamento, causando um *gridlock*. Esse é um modo de falha já documentado no controle semafórico, em que o transbordamento das filas paralisa a interseção [10].

O agente de ação cíclica é mais estável: como a ordem das fases é fixa, todas as aproximações são atendidas a cada ciclo e o *gridlock* não ocorre. Ele supera o estático nos cenários mais leves, como todo o Ingolstadt e os regimes de baixa demanda de Colônia. Entretanto, teve o pior desempenho de todos os modelos no regime de demanda moderada de Colônia (g1), com tempo de espera de 128 s contra 60 s do estático. Esse comportamendo se dá pois, para manter as filas curtas, o agente aprendeu a escolher verdes muito curtos, alternando de fase com muita frequência. Como cada troca de fase consome alguns segundos de amarelo e de tempo perdido, essa alternância excessiva desperdiça boa parte da capacidade do cruzamento e aumenta o tempo de espera, mesmo mantendo as filas baixas. Isso explica a aparente contradição na Figura 5: nesse regime, o agente cíclico mantém filas comparáveis às do estático (parte inferior), porém com espera muito maior (parte superior).

Por fim, a comparação entre as duas configurações de treino indica que treinar com demanda variável tende a ser vantajoso, sobretudo nas condições mais difíceis. O modelo treinado com

Tabela II: Tempo médio de espera (s) por condição, com o ganho percentual em relação ao semáforo estático (*fixed*) entre parênteses (positivo indica melhora). Os quatro modelos aprendidos combinam as duas larguras de treino (restrito/diverso) com os dois conjuntos de ações (livre/cíclico).

Cenário	Condição	*fixed*	restrito·livre	restrito·cíclico	diverso·livre	diverso·cíclico
Cologne	g0 empty	48	10 (+79%)	20 (+59%)	15 (+68%)	23 (+51%)
	g1 mid-spread	60	16 (+73%)	128 (−114%)	14 (+77%)	91 (−51%)
	g2 sat-peak	120	114 (+5%)	102 (+15%)	80 (+33%)	120 (+0%)
	g3 E-drain	42	13 (+70%)	22 (+48%)	12 (+72%)	23 (+45%)
Ingolstadt	g0 night-S	7	1 (+82%)	4 (+45%)	1 (+81%)	5 (+30%)
	g1 day-S	8	2 (+71%)	5 (+35%)	2 (+77%)	4 (+46%)
	g2 peak-S	10	6 (+43%)	6 (+41%)	4 (+60%)	6 (+37%)
	g3 evening-N	15	5 (+64%)	11 (+26%)	4 (+75%)	11 (+29%)

demanda diversa iguala ou supera o treinado com demanda fixa na maioria dos casos, com a diferença mais nítida nos regimes carregados de Colônia: no pico, o agente livre treinado com demanda diversa reduz a espera em 33%, contra apenas 5% do treinado com demanda fixa; e no regime moderado o agente cíclico diverso é bem menos penalizado que o restrito. Isso sugere que a exposição a diferentes níveis e distribuições de demanda durante o treino ajuda o agente a generalizar para condições que não observou de forma fixa.

VI. Conclusão e trabalhos futuros

O controle semafórico eficiente é um problema difícil, com diferentes abordagens sendo descritas na literatura. A compatibilidade do problema com o arcabouço de aprendizado por reforço, aliada ao crescente interesse recente no desenvolvimento de técnicas baseadas em aprendizado profundo, têm atraído a atenção de diferentes pesquisadores ao redor do mundo e, naturalmente, novas tecnologias serão absorvidas a fim de melhorar os sistemas atuais.

Neste trabalho, foi investigado o impacto da escolha do conjunto de ações sobre o desempenho de um agente de controle semafórico baseado em aprendizado por reforço, comparando de forma controlada a seleção livre de fase e o controle cíclico com duração em dois cruzamentos realísticos. Os dois agentes aprendidos superaram o semáforo estático na maioria das condições, e a simples variação do conjunto de ações, mantendo todo o resto fixo, levou a diferenças de desempenho de até 80%, o que evidencia que essa escolha de projeto é determinante. Cada paradigma revelou qualidades próprias: a seleção livre de fase alcança as maiores reduções no tempo de espera, graças à sua flexibilidade, mas se mostrou mais instável; o controle cíclico é mais estável e seguro, porém mais sensível à função de recompensa. Como contribuição adicional, observou-se que treinar o agente com demanda variável, em vez de fixa, melhorou seus resultados, sobretudo nas condições de tráfego mais exigentes.

Em trabalhos futuros, sugere-se estender este mesmo estudo ao contexto de múltiplos agentes, avaliando como a escolha do conjunto de ações afeta o desempenho quando vários cruzamentos precisam ser coordenados em uma rede viária. Sugere-se ainda investigar mais a fundo o fenômeno de *gridlock* observado sob a ação livre, buscando compreender as condições que o desencadeiam e mecanismos capazes de preveni-lo, de modo a combinar a eficiência da flexibilidade com a estabilidade do controle estruturado.

Referências

[1] Muzamil Eltejani Mohammed Ali, Akif Durdu, Seyit Celtek, and Alper Yilmaz. An adaptive method for traffic signal control based on fuzzy logic with webster and modified webster formula using sumo traffic simulator. *IEEE Access*, PP:1–1, 07 2021.

[2] James Ault and Guni Sharon. Reinforcement learning benchmarks for traffic signal control. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

[3] Changjian Cai and Min Wei. Adaptive urban traffic signal control based on enhanced deep reinforcement learning. *Scientific Reports*, 14(1):14116, 2024.

[4] Federal Highway Administration (FHWA). Traffic signal timing manual, June 2008.

[5] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *International Conference on Learning Representations (ICLR)*, 2018.

[6] Juntao Gao, Yulong Shen, Jia Liu, Minoru Ito, and Norio Shiratori. Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network, 2017.

[7] Wade Genders and Saiedeh Razavi. Using a deep reinforcement learning agent for traffic signal control, 2016.

[8] Roxanne Hawi, George Okeyo, and Michael Kimwele. Smart traffic light control using fuzzy logic and wireless sensor network. In *2017 Computing Conference*, pages 450–460, 2017.

[9] Rebecca Killick, Paul Fearnhead, and Idris A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[10] Mingyuan Li, Boyang Huang, Tianqi Jiang, Chenpu Li, Chunyu Liu, Yang Li, Ruimin Li, and Qiang Wu. Overflowlight: Real-time gridlock prevention and traffic signal optimization for urban intersections, 2026.

[11] Wei-Hua Lin and Chenghong Wang. An enhanced 0-1 mixed-integer lp formulation for traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):238–245, 2004.

[12] Xiao-Yang Liu, Ming Zhu, Sem Borst, and Anwar Walid. Deep reinforcement learning for traffic light control in intelligent transportation systems, 2023.

[13] Silas C. Lobo, Stefan Neumeier, Evelio M. G. Fernandez, and Christian Facchi. InTAS – the ingolstadt traffic scenario for SUMO. *arXiv preprint arXiv:2011.11995*, 2020.

[14] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[15] Patrick Mannion, Jim Duggan, and Enda Howley. *An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control*, pages 47–66. Springer International Publishing, Cham, 2016.

[16] Pitu Mirchandani and Larry Head. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6):415–432, 2001.

[17] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. Traffic light control using deep policy-gradient and value-function based reinforcement learning, 2017.

[18] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2016.

[19] Leena Singh, Sudhanshu Tripathi, and Himakshi Arora. Time optimization for traffic signal control using genetic algorithm. *LETTERS International Journal of Recent Trends in Engineering*, 2, 12 2009.

[20] Allan Souza, Celso Brennand, Roberto Yokoyama, Erick Donato, Edmundo Madeira, and Leandro Villas. Traffic management systems: A classification, review, challenges, and future perspectives. *International Journal of Distributed Sensor Networks*, 13:155014771668361, 04 2017.

[21] Sandesh Uppoor, Oscar Trullols-Cruces, Marco Fiore, and Jose M. Barcelo-Ordinas. Generation and analysis of a large-scale urban vehicular mobility dataset. *IEEE Transactions on Mobile Computing*, 13(5):1061–1075, 2014.

[22] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2094–2100, 2016.

[23] Andrea Vidali, Luca Crociani, Giuseppe Vizzari, Stefania Bandini, et al. A deep reinforcement learning approach to adaptive traffic lights management. In *Woa*, pages 42–50, 2019.

[24] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1995–2003, 2016.

[25] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[26] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.