UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LIZA LUNARDI LEMOS

Co-aprendizado entre motoristas e controladores semafóricos em simulação microscópica de trânsito

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Ana L. C. Bazzan

Porto Alegre
2018

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Lemos, Liza Lunardi

Co-aprendizado entre motoristas e controladores semafóricos em simulação microscópica de trânsito / Liza Lunardi Lemos. – Porto Alegre: PPGC da UFRGS, 2018.

56 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2018. Orientador: Ana L. C. Bazzan.

1. Aprendizado por reforço multiagente. 2. Sistemas multiagente. 3. Escolha de rotas. 4. Controlador semafórico. I. Bazzan, Ana L. C.. II. Título.

“Foi o tempo que dedicastes à tua rosa
que a fez tão importante.”
— ANTOINE DE SAINT-EXUPÉRY

RESUMO

Um melhor uso da infraestrutura da rede de transporte é um ponto fundamental para atenuar os efeitos dos congestionamentos no trânsito. Este trabalho utiliza aprendizado por reforço multiagente (MARL) para melhorar o uso da infraestrutura e, consequentemente, mitigar tais congestionamentos. A partir disso, diversos desafios surgem. Primeiro, a maioria da literatura assume que os motoristas aprendem (semáforos não possuem nenhum tipo de aprendizado) ou os semáforos aprendem (motoristas não alteram seus comportamentos). Em segundo lugar, independentemente do tipo de classe de agentes e do tipo de aprendizado, as ações são altamente acopladas, tornando a tarefa de aprendizado mais difícil. Terceiro, quando duas classes de agentes co-aprendem, as tarefas de aprendizado de cada agente são de natureza diferente (do ponto de vista do aprendizado por reforço multiagente). Finalmente, é utilizada uma modelagem microscópica, que modela os agentes com um alto nível de detalhes, o que não é trivial, pois cada agente tem seu próprio ritmo de aprendizado. Portanto, este trabalho não propõe somente a abordagem de co-aprendizado em agentes que atuam em ambiente compartilhado, mas também argumenta que essa tarefa precisa ser formulada de forma assíncrona. Além disso, os agentes motoristas podem atualizar os valores das ações disponíveis ao receber informações de outros motoristas. Os resultados mostram que a abordagem proposta, baseada no co-aprendizado, supera outras políticas em termos de tempo médio de viagem. Além disso, quando o co-aprendizado é utilizado, as filas de veículos parados nos semáforos são menores.

Palavras-chave: Aprendizado por reforço multiagente. sistemas multiagente. escolha de rotas. controlador semafórico.

Co-learning between drivers and traffic lights in microscopic traffic simulation

ABSTRACT

A better use of transport network infrastructure is a key point in mitigating the effects of traffic congestion. This work uses multiagent reinforcement learning (MARL) to improve the use of infrastructure and, consequently, to reduce such congestion. From this, several challenges arise. First, most literature assumes that drivers learn (traffic lights do not have any type of learning) or the traffic lights learn (drivers do not change their behaviors). Second, regardless of the type of agent class and the type of learning, the actions are highly coupled, making the learning task more difficult. Third, when two classes of agents co-learn, the learning tasks of each agent are of a different nature (from the point of view of multiagent reinforcement learning). Finally, a microscopic modeling is used, which models the agents with a high level of detail, which is not trivial, since each agent has its own learning pace. Therefore, this work does not only propose the co-learnig approach in agents that act in a shared environment, but also argues that this taks needs to be formulated asynchronously. In addtion, driver agents can update the value of the available actions by receiving information from other drivers. The results show that the proposed approach, based on co-learning, outperforms other policies regarding average travel time. Also, when co-learning is use, queues of stopped vehicles at traffic lights are lower.

Keywords: multiagent reinforcement learning, multiagent system, route choice, traffic signal control.

LISTA DE ABREVIATURAS E SIGLAS

KSP	*Shortest Loopless Path*
MARL	Apredizado por reforço multiagente
MDP	Processo de decisão de Markov
QL	Q-learning
RL	Aprendizado por reforço
SUMO	*Simulation of Urban Mobility*

LISTA DE SÍMBOLOS

- Parâmetros gerais

$G = (V, L)$ Grafo / rede de tráfego

N demanda (aprox. número de motoristas)

TL número de semáforos

$pares\ OD$ conjunto de pares OD

$timesteps$ número de passos da simulação

τ tempo antes dos semáforos começarem a aprender

P_i conjunto de parâmetros

- Parâmetros dos agentes motorista

K número de rotas

α_d taxa de aprendizagem

ε_d taxa de exploração

d_d taxa de decaimento em ε_d

A_d conjunto de ações

- Parâmetros dos agentes semafórico

ρ número de fases

δ tempo decorrido da fase corrente

λ_i tamanho da fila na fase i

AQL tamanho médio da fila

$minVerde$ tempo mínimo de verde

$maxVerde$ tempo máximo de verde

$minVeloc$ limiar para considerar um veículo parado

α_t taxa de aprendizagem

γ fator de desconto

ε_t taxa de exploração

d_t taxa de decaimento em ε_t

S_t conjunto de estados

A_t conjunto de ações

Δ intervalo para seleção de ação

LISTA DE FIGURAS

Figura 2.1 Exemplo de conjunto de vias que recebem o sinal verde ou vermelho em cada fase 20
Figura 2.2 Exemplo de sequência das fases ao longo do tempo 20

Figura 4.1 Esquema de aprendizado dos agentes motorista (azul) e agentes semafóricos (vermelho) 31

Figura 5.1 Rede viária em grade 6 $\times$ 6: todas as vias possuem dois sentidos, com uma faixa em cada sentido. Linhas em destaque indicam uma maior capacidade das faixas 41
Figura 5.2 Fluxo de veículos na rede viária ao longo do tempo. Motoritas viajam sempre pela menor rota e os semáforos executam sua política fixa. Linha pontilhada vertical indica 1 hora de simulação 41
Figura 5.3 Variação de ε para taxa de decaimento $dr = 0,92; 0,95$ 43
Figura 5.4 Rede viária em grade 6 $\times$ 6: cruzamentos destacados C3, D3 e E3 possuem maior fluxo de veículos 46
Figura 5.5 Número de veículos em fila para duas classificações dos semáforos. 47
Figura 5.6 Comparação entre os quatro casos investigados em termos de tempo médio de viagem 48

Figura 6.1 Comparação entre os quatro casos investigados em termos de tempo médio de viagem. Motoristas possuem episódios síncronos 53

LISTA DE TABELAS

Tabela 4.1 Comparação entre os MDPs dos agentes motorista e semafórico 35

Tabela 5.1 Resultados obtidos para o caso (ii) Fixo+MotorQL, quando somente os motoristas aprendem. Tempo médio de viagem (e desvio padrão) para diferentes valores da taxa de aprendizado α_d, taxa de decaimento dr_d e K 45

Tabela 5.2 Resultados obtidos para o caso (iiI) SemafQL+MenorRota quando somente os semáforos aprendem. Tempo médio de viagem (e desvio padrão) para $\alpha_t = 0.1$, $\gamma = 0.8$ e diferentes valores da taxa de decaimento dr_t 45

Tabela 5.3 Resultados obtidos nos quatro casos investigados. Valores em tempo médio de viagem. 47

SUMÁRIO

1 INTRODUÇÃO 12
1.1 Contribuições 13
1.2 Organização dos capítulos 14
2 FUNDAMENTAÇÃO TEÓRICA 16
2.1 Agentes autônomos e sistemas multiagente 16
2.2 Aprendizado por reforço 16
2.3 Aprendizado por reforço multiagente 18
2.4 Sistemas de transporte 19
2.5 Simulação de tráfego 21
2.6 Resumo 23
3 TRABALHOS RELACIONADOS 24
3.1 Escolha de rotas 24
3.2 Controladores semafóricos 25
3.3 Ambas classes de agentes 28
3.4 Resumo 29
4 ABORDAGEM PROPOSTA 30
4.1 MDP dos agentes motoristas 32
4.2 Algoritmo dos agentes motoristas 34
4.3 MDP dos agentes semafóricos 35
4.4 Algoritmo dos agentes semafóricos 36
4.5 Resumo 38
5 EXPERIMENTOS 39
5.1 Cenário estudado 39
5.2 Métricas 42
5.3 Configuração 42
5.4 Resultados experimentais 44
5.4.1 Resultados - motoristas 44
5.4.2 Resultados - semáforos 45
5.4.3 Resultados - co-aprendizado 46
6 CONSIDERAÇÕES FINAIS 49
6.1 Contribuições 50
6.2 Perspectivas de continuação 50
ANEXO A - MODELAGEM DE SIMULAÇÃO SíNCRONA 52
REFERÊNCIAS 54

1 INTRODUÇÃO

O número de veículos nas ruas cresce juntamente com o aumento da população urbana, pelo menos em economias emergentes. Por exemplo, de 2004 a 2014, o número de veículos para 1000 habitantes no leste Europeu aumentou de 222,5 para 346,8 e no Brasil de 119,7 para 206,0 (DAVIS; WILLIAMS; BOUNDY, 2016). Sabe-se que os investimentos na infraestrutura da rede de transporte não acompanham o crescimento da frota veicular; portanto, os congestionamentos são um fenômeno sempre presente que representa grande desafio para a mobilidade urbana. Uma maneira popular de mitigar esse problema é usar melhor a infraestrutura existente.

Para esse fim, as abordagens populares vão desde o controle e a otimização clássicos; por exemplo, uma central atribui rotas para os motoristas; aos sistemas multiagente e ao aprendizado por reforço multiagente (MARL - *Multiagent Reinforcement Learning*). Este trabalho segue o último tópico. Em MARL, múltiplos agentes interagem em um ambiente compartilhado. O objetivo de cada agente é aprender como se comportar, a fim de maximizar sua política. Um desafio nesse contexto é que, quando diversos agentes estão aprendendo em um ambiente no qual há um alto acoplamento entre as ações dos agentes, as tarefas individuais de aprendizagem são muito difíceis. Além disso, os agentes têm de lidar com o ruído decorrente do próprio ambiente e, também, com o ruído causado pelas ações de outros agentes.

A literatura sobre MARL raramente aborda a tarefa de aprendizagem em que mais de uma classe de agentes aprende. Na verdade, quase todos os trabalhos lidam com o aprendizado de semáforos (ver Seção 3.2) ou com o aprendizado em nível de motoristas (por exemplo, escolha de rotas, ver Seção 3.1). No presente trabalho, modelam-se duas classes de agentes que aprendem: agentes motorista (representando demanda) e agentes semafórico (representando a oferta). A demanda está associada aos motoristas, os quais realizam viagens na rede viária. A oferta é a infraestrutura física, compreendendo também os semáforos. Tem-se então, o co-aprendizado, em que as duas classes de agentes aprendem simultaneamente. Essa abordagem é mais realista, porque o co-aprendizado lida com demanda e oferta, adaptando simultaneamente, conforme detalhado na Seção 2.4. Além disso, do ponto de vista de MARL, colocam-se alguns desafios. Primeiro, o ambiente onde interagem motoristas e semáforos é competitivo, pois os seus objetivos são conflitantes e os motoristas possuem comportamento egoísta. Segundo, ambas as tarefas de aprendizado são de natureza diferente, como será explicado, com mais detalhes, adiante.

Na área de sistemas multiagente, poucos trabalhos se aproximam da abordagem proposta neste trabalho. No trabalho de (WIERING, 2000), motoristas e semáforos estão aprendendo e compartilhando as mesmas funções valor, enquanto o trabalho descrito em (BAZZAN et al., 2007) discute os efeitos de adaptar motoristas e semáforos (porém os agentes motoristas não empregam MARL). No entanto, nenhuma das abordagens utilizam todo o potencial do modelo de simulação microscópico como ambiente para o MARL; ambas utilizam uma modelagem discreta (não-contínua) das vias. A última utiliza um modelo baseado em fila, enquanto a primeira utiliza um simulador de eventos discretos. Dessa forma, nenhuma leva em consideração a aceleração/desaceleração ou velocidades diferentes pelos veículos. Nesse sentido, eles podem, na melhor das hipóteses, ser considerados uma espécie de modelagem mesoscópica do movimento do trânsito.

Neste trabalho, por outro lado, utiliza-se um simulador microscópico, cujo modelo permite a modelagem de agentes em um nível fino; por exemplo, é possível modelar não apenas velocidades e acelerações diferentes, mas também um comportamento do tipo car-following (ver Seção 2.4). Esse modelo contínuo (não discreto) tem um impacto nas escolhas dos motoristas no nível do link (por exemplo, mudança de faixa) que afeta seus tempos de viagem, algo que não é considerado em uma modelagem mesoscópica. Mais detalhes sobre os modelos de simulação são dados na Seção 2.5.

Finalmente, uma grande diferença em relação a este trabalho é que se consideram duas classes de agentes na tarefa de aprendizado, enfrentando o desafio de que essas tarefas não podem ser sincronizadas. Por exemplo, enquanto a tarefa de aprendizado do semáforo é de horizonte infinito, quando se trata da tarefa de aprendizado do motorista, a formulação mais comum é episódica (por exemplo, um episódio é concluído quando todos os motoristas chegam ao destino). No entanto, mesmo assim, não é totalmente realista, pois nem todos os motoristas chegam ao seu destino ao mesmo tempo. Portanto, neste trabalho propõe-se um modelo totalmente assíncrono, onde os semáforos aprendem continuamente, e cada motorista descobre ao seu próprio ritmo, de modo que o episódio de aprendizado do motorista i possa ser diferente do episódio do motorista j.

1.1 Contribuições

As principais contribuições deste trabalho são:

- Consideram-se duas classes de agentes de aprendizado – motoristas e semáforos

– que aprendem em um sistema de tráfego urbano. As implicações são múltiplas. Primeiro, cada classe tem objetivos diferentes (os agentes motoristas visam minimizar os tempos de viagem individuais, enquanto o objetivo dos agentes semafóricos é minimizar as filas localmente). Em segundo lugar, as ações dos agentes são altamente acopladas: no caso dos motoristas, os tempos de viagem individuais são altamente afetados pelas políticas de controle implementadas pelos controladores dos semáforos, bem como por outras opções de rotas dos motoristas, uma vez que uma viagem se estende por várias seções de uma rede. Cada controlador semafórico não é apenas afetado pelas rotas dos motoristas, mas também pelas políticas executadas nas intersecções vizinhas.

- Os agentes motoristas aprendem em episódios assíncronos, enquanto os agentes semafóricos aprendem continuamente (horizonte infinito).
- Também se considera a comunicação entre motoristas para permitir que um motorista que está terminando sua viagem possa enviar mensagens para outro motorista que iniciará sua viagem com origem e destino semelhantes, para disseminar informações sobre a rota que acabou de ser usada e ajudar outros motoristas a fazer suas escolhas de rota.
- Utiliza-se um simulador de tráfego microscópico, o qual modela as entidades do sistema de transporte com mais precisão.

1.2 Organização dos capítulos

Os próximos capítulos desta dissertação estão organizados da seguinte forma:

- Capítulo 2: apresenta a fundamentação teórica necessária para dar entendimento a esta dissertação. São discutidos alguns conceitos básicos em relação aos sistemas multiagente (2.1), aprendizado por reforço (2.2) e sistemas de transporte(2.4).
- Capítulo 3: apresenta uma discussão acerca dos principais trabalhos relacionados a esta dissertação. É dividida em três partes: escolha de rotas, controle semafórico e ambas as classes de agentes adaptando.
- Capítulo 4: discute a abordagem baseada em agentes para escolha de rotas, controle semafórico e co-aprendizado.
- Capítulo 5: apresenta o cenário estudado e os experimentos realizados para avaliar o desempenho da abordagem proposta.

- Capítulo 6: apresenta as conclusões, contribuições e perspectivas de continuidade deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Esta dissertação apresenta uma abordagem baseada em agentes para simulação de agentes motoristas e semafóricos. Este capítulo apresenta os conceitos básicos de cada um desses tópicos, para embasar o leitor com alguns dos temas abordados na dissertação. Para estudos mais aprofundados são apresentadas referências.

Este capítulo começa com uma breve revisão sobre agentes autônomos e sistemas multiagente (Seção 2.1). Além disso, discute, também, sobre aprendizado por reforço e aprendizado por reforço multiagente (Seção 2.2 e 2.3), sistemas de transporte (Seção 2.4) e simulação de tráfego (Seção 2.5).

2.1 Agentes autônomos e sistemas multiagente

Não há uma definição universalmente aceita para a definição de agente. Há um consenso geral de *autonomia* como noção central. Segundo Wooldridge (2009), um agente é um sistema de computador que é situado em um ambiente, e que é capaz de realizar ações autônomas no ambiente para atingir os objetivos para o qual foi projetado.

Um sistema multiagente consiste em múltiplos agentes que interagem uns com outros e, tipicamente, trocam mensagens (WOOLDRIDGE, 2009). Cada agente atua sobre o ambiente e possui, portanto, uma "esfera"de influência sobre o mesmo. As esferas de influências de diferentes agentes podem se sobrepor e isso pode gerar relações entre os agentes.

Para uma visão mais detalhada em agentes autônomos e sistemas multiagente, recomenda-se a leitura de (WOOLDRIDGE, 2009).

2.2 Aprendizado por reforço

No aprendizado por reforço (RL - *Reinforcement Learning*), o agente aprende um comportamento através da sua interação com o ambiente. O agente deve aprender o que fazer para maximizar um valor numérico, chamado de recompensa. Ele deve descobrir quais ações produzem a melhor recompensa, experimentando-as. A partir das interações com o ambiente, o agente define uma forma de se comportar em determinado momento; isto é, a política do agente. A política é o mapeamento de estados percebidos no am-

biente para ações a serem tomadas quando está nesses estados. A ideia é que o agente interaja com o ambiente para aprender a política que maximiza a soma das recompensas esperadas. Além de o agente lidar com a recompensa imediata, deve aprender a lidar com o problema de decisão sequencial, pois suas ações podem afetar as recompensas subsequentes originadas de sua situação futura.

O aprendizado por reforço pode ser modelado como um processo de decisão de Markov (MDP - *Markov decision process*) composto por uma tupla (S, A, T, R), onde S é o conjunto de estados; A é o conjunto de ações; T é a função de transição que modela a probabilidade do sistema de mover do estado $s \in S$ para o estado $s' \in S$, ao executar a ação $a \in A$; e R é a função de recompensa que produz um número real associado à execução de uma ação $a \in A$ quando se encontra no estado $s \in S$.

Um algoritmo amplamente utilizado de aprendizado por reforço é o Q-*learning* (QL). No QL um agente aprende a utilidade esperada de se realizar uma dada ação a em um dado estado s e seguir sua política a partir disso. Chama-se essa utilidade esperada de valor-Q, representado por $Q(s, a)$. O valor-Q pode ser aprendido diretamente a partir da recompensa recebida pelo agente. Uma tabela-Q armazena os valores-Q para cada par estado-ação. A atualização de $Q(s, a)$ é feita através da Eq. 2.1, onde $\alpha \in [0, 1]$ é a taxa de aprendizagem, $\gamma \in [0, 1]$ é o fator de desconto e a' é umas das ações que podem ser realizadas quando o agente está em s'. A taxa de aprendizado determina o quanto a nova experiência influencia o valor-Q já aprendido até o momento. O fator de desconto desvaloriza o reforço em função da diferença temporal.

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma max(Q(s', a')) - Q(s, a)) \tag{2.1}$$

Tendo calculado os valores-Q para os pares de estado-ação, um agente precisa de uma estratégia para selecionar qual ação executar dentro das possíveis ações de cada estado da tabela-Q. Uma maneira padrão de fazer isso é através do balanceamento da exploração e do aproveitamento das ações. Uma estratégia conhecida é a ε-*greedy*. Nessa estratégia, a ação com melhor valor é escolhida com uma probabilidade de 1 - ε e uma ação aleatória é selecionada com uma probabilidade ε. Na formulação convencional, o valor de ε é constante durante todo tempo. Neste caso em particular, ε começa com um valor alto, por exemplo 1, e decresce ao longo do tempo sendo multiplicado por uma taxa de decaimento; ou seja, no início há uma maior exploração das ações e, ao longo do tempo, há um maior aproveitamento das ações.

Para uma leitura mais detalhada sobre aprendizado por reforço recomenda-se a

leitura de (SUTTON; BARTO, 1998) e (TUYLS; WEISS, 2012). Uma descrição mais aprofundada sobre Q-*learning* é encontrada em (WATKINS; DAYAN, 1992).

2.3 Aprendizado por reforço multiagente

No aprendizado por reforço multiagente muito desafios surgem, pois agora existem vários agentes aprendendo no mesmo ambiente. Além de os agentes terem de se adaptar ao ambiente, devem se adaptar ao comportamento dos outros agentes.

O aprendizado por reforço multiagente (MARL) pode ser modelado por um processo de decisão de Markov multiagente (também chamado de jogo estocástico), que é uma generalização do MDP monoagente (apresentado anteriormente), mas constituído de um conjunto de agentes. O processo de decisão de Markov multiagente (MMDP - *multi-agente Markov decision process*) é composto por uma tupla $(S, A_1, ..., A_n, T, R_1, ..., R_n)$, onde n é o número de agentes, S é o conjunto de estados, A_i é o conjunto de ações disponíveis, T é a função de transição $T : S \times A \times S$; e R_i é a função de recompensa dos agentes. A função de transição T mapeia as ações combinadas que cada agente realizou no estado atual para uma distribuição de probabilidade sobre S. A função de recompensa R depende das ações realizadas pelos outros agentes também.

No entanto, a modelagem do processo de decisão de Markov multiagente pode gerar problemas de larga escala, pois o espaço de estados-ações cresce de acordo com o número de agentes. A partir disso, cada agente aprende de forma descentralizada e desconsiderando a adaptação de outros agentes. Dessa forma, o agente entende o aprendizado dos outros agentes como uma mudança no ambiente.

Em jogos estocásticos o MDP é modelado incluindo os conjuntos de estados S e ações A. Já em jogos repetidos o MDP desconsidera o conjunto de estados S (ou seja, considera um estado único). Como será discutido adiante (ver Seções 4.1 e 4.3), as duas classes de agentes que são consideradas na tarefa de aprendizado deste trabalho estão associadas a jogos repetidos (motoristas) e a jogos estocásticos (semáforos).

Uma maior discussão sobre o assunto pode ser encontrada em (BUŞONIU; BABUSKA; SCHUTTER, 2008). Nesse trabalho são apresentadas as principais técnicas de MARL, principais desafios e aplicações da área.

2.4 Sistemas de transporte

Um sistema de transporte pode ser visto como sendo composto por duas partes: oferta e demanda. A oferta representa a infraestrutura física (rede de transporte/viária, incluindo medidas de controle, como controladores semafóricos). Esta rede viária pode ser representada como um grafo direcionado $G = (V, L)$, onde o conjunto de vértices V representa as intersecções/cruzamentos, e o conjunto de links L representa os segmentos das vias. Cada link $l \in L$ tem uma capacidade, ou seja, o número de unidades de fluxo de tráfego que um link suporta por unidade de tempo.

A demanda é representada pelos usuários do sistema de transporte. Neste texto, esses usuários são chamados de agentes motoristas. Esses agentes fazem uma viagem de um vértice de origem $i \in V$ para um vértice de destino $j \in V$ na rede viária. Cada par origem-destino (de agora em diante par OD) está associado a uma série de rotas que conectam a respectiva origem ao seu destino. O conjunto de rotas é representado por $R = \{r_1^1, ..., r_1^k, ..., r_{OD}^k\}$, onde K é o número de rotas por par OD; cada rota $r \in R$ consiste em um conjunto de links; e N é o fluxo de veículos (que, com algum abuso de notação, assume-se como sendo o número de motoristas que viajam em todos os links do conjunto L). Note que, em uma simulação microscópica, esse fluxo dificilmente é constante (como assumido pela maioria das abordagens macroscópicas). Em vez disso, ele muda ao longo do tempo, como será discutido na Seção 2.5.

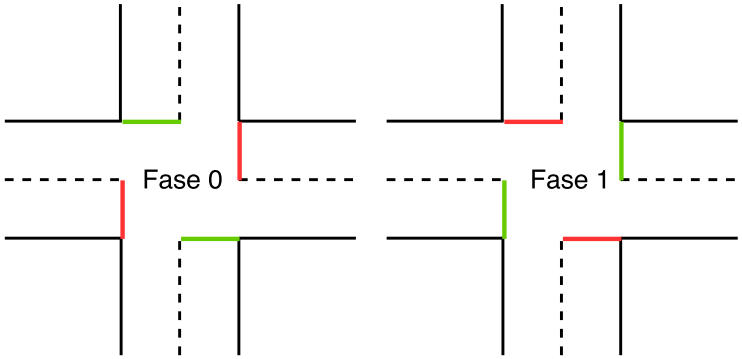
O problema de escolha de rotas trata sobre como os motoristas escolhem uma rota entre seus vértices de origem e de destino. Se todos os motoristas selecionarem aproximadamente as mesmas rotas (por exemplo, cada um seleciona sua rota mais curta) a maioria dessas rotas ficará saturada. Por outro lado, os motoristas têm outras opções de rotas para viajar que envolvem links que podem conter menos veículos. Em consequência, ao mudar ligeiramente as rotas, para a maioria dos motoristas o tempo de viagem será melhor. Lembra-se também que o problema de escolha de rotas está relacionado ao problema de alocação de tráfego (TAP). Um dos métodos mais simples e conhecidos de alocação de tráfego é chamado de *all-or-nothing* (ORTÚZAR; WILLUMSEN, 2001). Nesse método, uma autoridade central atribui uma rota para cada viagem ou veículo. Assim, não há espaço para uma decisão autônoma em nível individual dos veículos. O TAP é normalmente realizado usando um modelo macroscópico (veja Seção 2.5), ou seja, não se considera nenhum movimento físico real, e os tempos de viagem não são efetivamente medidos pelo simulador (como no modelo de simulação microscópica), mas são dados

por uma função que atribui um custo (normalmente tempo de viagem) dado o número de veículos, usando um link. Uma função de custo comumente usada é a função BPR, que é dada pela Eq. 2.2, onde f_e é o fluxo total do link e, t_e^f é o tempo de viagem de fluxo livre do link e, c_e é a capcidade do link e, e α e β são parâmetros, cujos valores padrão são 0,15 e 4, respectivamente.

$$t_e(f_e) = t_e^f(1 + \alpha(f_e/c_e)^\beta) \tag{2.2}$$

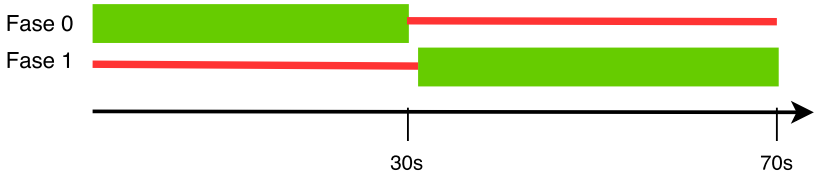
Além disso, um modelo macroscópico teria dificuldade em considerar os controladores semafóricos, uma vez que não foi projetado para esse propósito. Embora uma modelagem macroscópica e o TAP sejam razoáveis para fins de planejamento, eles fornecem apenas uma grosseira aproximação do que realmente acontece se as decisões individuais forem consideradas (por exemplo, quando um agente motorista enfrenta o ambiente físico real).

Figura 2.1: Exemplo de conjunto de vias que recebem o sinal verde ou vermelho em cada fase



Fonte: A Autora

Figura 2.2: Exemplo de sequência das fases ao longo do tempo



Fonte: A Autora

No que diz respeito à oferta ou infraestrutura, este trabalho está mais preocupado com os controladores semafóricos (daqui por diante, por simplicidade, usa-se semáforos para significar um controlador para todo o conjunto de luzes em um cruzamento/intersecção), que são um tipo de sinalização que ajuda a controlar o tráfego. De acordo com (ROESS; PRASSAS; MCSHANE, 2004), o componente fundamental de um

semáforo é o ciclo, o qual representa uma rotação completa através de todas as indicações verdes. Os ciclos estão associados a um esquema de temporização que permite o sinal verde a cada fase em determinados intervalos (que não precisam ser fixos). Uma fase consiste em um grupo de faixas que recebem sinal verde simultaneamente, conforme ilustrados na Fig. 2.1. Uma política comumente utilizada pelos semáforos é a política fixa, na qual existe um ciclo com tempo fixo e a fases também recebem um tempo fixo, mas não necessariamente o mesmo tempo para todas as fases, por exemplo, como ilustrado na Fig. 2.2, um ciclo com duas fases: a fase 0 pode ter um tempo fixo de 30s e a fase 1 um tempo fixo de 40s; sendo que o ciclo possui um tempo fixo total de 70s (está sendo ignorado os tempos de amarelo e todos vermelho a fim de simplificação), enquanto uma fase recebe o sinal de verde, a outra fase recebe o sinal vermelho.

O número de fases, bem como quais vias são agrupadas, é uma questão de projeto que envolve movimentos permitidos na intersecção e está fora do escopo deste trabalho. Assume-se que esses grupos são fornecidos. A única preocupação é como definir os tempos das fases ou a divisão do tempo de verde entre essas fases.

2.5 Simulação de tráfego

Sistemas de transportes são sistema complexos; seu comportamento é definido pelas interações entre várias entidades de comportamentos diferentes. Testes ou experimentos em campo são custosos e demandam tempo. Por isso, utilizam-se simuladores de tráfego. A simulação também pode ser utilizada para avaliar a viabilidade de técnicas inovadoras.

A literatura apresenta modelos de simulação de tráfego que são classificados de acordo com o nível de detalhe da representação do sistema. Entre os modelos existentes, destacam-se os seguintes:

- Macroscópico: há uma abstração das ações dos indivíduos e suas interações. Descreve as entidades dos sistemas com um alto nível de detalhe. O sistema é representado de maneira agregada a partir de histogramas e dados de volumes, densidades e velocidades. Por exemplo, uma mudança de pista nem é representada, pois os veículos não existem como entidades separadas, são representados apenas em agregados de densidade. Esse modelo é computacionalmente eficiente, mas sensível aos parâmetros iniciais que podem impactar sobre o comportamento do sistema. Além

disso, são úteis para predições sobre valores grosseiros de densidade e fluxo.

- Microscópico: as entidades e suas interações são representadas com um alto nível de detalhe e de maneira individual. Por exemplo, é possível modelar a mudança de pista e o comportamento do tipo car-following. Existem vários modelos de car-following (veja, por exemplo, (GIPPS, 1981) para detalhes), nos quais há uma interação entre o veículo líder e o veículo seguidor A interação segue um mecanismo de estímulo-resposta para calcular a aceleração, distância e tempo de reação em relação ao líder. Esse modelo geralmente é complexo, de desenvolvimento custoso e exige mais parâmetros de configuração.
- Mesoscópico: é o nível intermediário entre o macroscópico e o microscópico. Possui um nível de detalhe razoável das entidades, mas descreve as relações entre elas em um nível de abstração maior. Por exemplo, uma manobra de mudança de pista pode ser representada como um evento instantâneo (para todos os veículos) e ser baseada na densidade da pista e não na interação dos veículos propriamente ditos.

Conforme mencionado, este trabalho emprega um modelo de simulação microscópica, no qual é possível simular todas as entidades do sistema de transporte (tanto da oferta, como da demanda) as interações entre elas com alto nível de detalhe. Para os experimentos é utilizado o simulador de tráfego microscópico SUMO (*Simulation of Urban Mobility*) (BEHRISCH et al., 2011).

O SUMO possui um modelo de car-following incorporado para modelar o comportamento das unidades de veículos no nível físico.

No SUMO, o modelo de car-following implementa o modelo de Krauβ (KRAUSS, 1998). A aceleração do veículo seguidor é baseada na desaceleração do veículo líder, na velocidade relativa entre os veículos e no tempo de reação. No modelo de Krauβ há dois tipos de movimento: movimento livre e interação com outro veículo. No movimento livre, a velocidade do veículo está limitada a uma velocidade máxima, que pode ser a velocidade escolhida pelo motorista. Já no movimento de interação com outro veículo, a velocidade é especificada de acordo como as interações entre os veículos, garantindo que eles não colidam.

Obviamente, esse comportamento influencia o tempo de viagem. Assim, trabalhos como Bazzan et al. (2007), Wiering (2000), que consideram a adaptação em duas classes diferentes de agentes, mas não possuem modelo interno de car-following, não são totalmente realistas.

Uma explicação mais detalhada sobre sistemas de transporte e simulação de trá-

fego pode ser encontrada em (BAZZAN; KLÜGL, 2013).

2.6 Resumo

Esse capítulo apresentou os conceitos que são utilizados nessa dissertação. O sistema de transporte é dividido em: demanda e oferta. Pelo lado da demanda, os motoristas realizam viagens entre a origem e o destino. Pelo lado da oferta, os controladores semafóricos controlam o fluxo local nas intersecções. O modelo de simulação microscópico simula as entidades do sistema de transporte em um alto nível de detalhe e implementa o modelo de car-following, que é um diferencial deste trabalho.

No aprendizado por reforço os agentes aprendem uma política através de sucessivas interações com o ambiente. Um sinal de recompensa é recebido a cada interação e o objetivo é maximizar esse sinal. No aprendizado por reforço multiagente, múltiplos agentes interagem e aprendem em um ambiente compartilhado. Cada agente aprende de forma descentralizada e considera o aprendizado dos outros agentes como uma mudança do ambiente. O algoritmo de aprendizado por reforço usado é o Q-learning, no qual um agente aprende a utilidade esperada de se realizar uma dada ação em um dado estado e seguir sua política a partir disso. A interação entre os agentes da abordagem proposta é dada com mais detalhes no Capítulo 4.

3 TRABALHOS RELACIONADOS

Este capítulo discute trabalhos que também lidam com escolha de rotas e controle semafórico como forma de melhorar o fluxo do tráfego. Foca-se em métodos baseados em aprendizado, mas também se incluem alguns outros trabalhos similares. Como existem duas classes de agentes, esta seção é dividida em três partes. Primeiro, são discutidas abordagens focadas em motoristas, cuja tarefa é selecionar uma rota. Segundo, são revisados trabalhos focados em agentes semafóricos. Terceiro, são discutidos trabalhos que lidam com as duas classes de agentes aprendendo.

3.1 Escolha de rotas

No trabalho de (DIA; PANWAI, 2014), redes neurais são usadas para prever a escolha de rota dos motoristas, assim como a conformidade por tais predições, sobre uma influência de mensagens contendo informações do tráfego. Entretanto, os autores focam mais no impacto das mensagens nos agentes do que o impacto da distribuição do tráfego e tempo de viagem.

Outro trabalho que utiliza redes neurais para modelar as estratégias utilizadas pelos agentes motorista é de (BARTHÉLEMY; CARLETTI, 2017). Os parâmetros da rede neural são determinados em uma fase preliminar de treinamento. A saída da rede neural é a ação a ser realizada pelo agente: permanecer ou modificar o caminho da viagem, calculando o novo caminho mais curto, evitando os links congestionados.

O trabalho de (DIAS et al., 2014) utiliza o Inverted Ant Colony Optimization (IACO), que é baseado no algoritmo da colônia de formigas. A diferença é que, ao invés de usar o feromônio para atrair as formigas, ele inverte esse efeito, repelindo-as. O sistema é definido de forma descentralizada, de modo que as formigas (veículos) depositam seus feromônios nas vias que estão utilizando. Dessa forma, as formigas são repelidas de vias congestionadas, contribuindo para uma melhor distribuição do tráfego na rede viária. Entretanto, o feromônio precisa ser guardado por uma central; assim, essa abordagem não é adequada para uma modelagem descentralizada.

A abordagem proposta por (CLAES; HOLVOET; WEYNS, 2011) também se baseia em colônia de formigas, combinada com predição de tráfego na rede viária. É uma abordagem descentralizada, que antecipa a presença de congestionamento e roteia os veículos de acordo com essa informação antecipada. As formigas/veículos deixam infor-

mações relevantes sobre o tráfego nos feromônios para as outras formigas. Além disso, quando os veículos escolhem uma rota, informam aos agentes da infraestrutura para poder incluir essa informação na previsão de ocupação da rede viária. No entanto, esses agentes da infraestrutura, de certa forma, possuem informações centralizadas.

Uma outra abordagem para o problema de escolha de rotas utilizando teoria de jogos é (GALIB; MOSER, 2011). Essa abordagem utiliza apenas experiências passadas para a escolha de rotas e a escolha em si é feita em cada intersecção da rede viária. Entretanto, assume-se que a informação histórica está disponível para todos os veículos da rede. Além disso, a abordagem não é baseada em uma modelagem microscópica.

O trabalho de (SHARON et al., 2017) utiliza pedágios adaptativos para otimizar as rotas dos veículos conforme os pedágios mudam. Diferentemente da proposta do presente trabalho, eles estão preocupados com a alocação das escolhas para o ótimo do sistema, o qual pode ser alcançado impondo custos aos motoristas (nesse caso, pedágios). Na mesma linha, (BURIOL et al., 2010) lida com TAP de uma perspectiva centralizada para encontrar uma alocação que alinhe usuários e utilitários do sistema, impondo pedágios em alguns links.

Duas abordagens baseadas em MARL são (RAMOS; GRUNITZKI, 2015) e (BAZZAN; GRUNITZKI, 2016). Enquanto a primeira é baseada em jogos repetidos, a última considera um jogo estocástico. Em (RAMOS; GRUNITZKI, 2015), cada agente é modelado como um *learning automata* (NARENDRA; THATHACHAR, 1989). Os agentes recebem um conjunto de rotas pré-computadas que podem ser recalculadas ao longo da simulação com uma dada probabilidade. Em (BAZZAN; GRUNITZKI, 2016), os estados são definidos pelo link no qual o agentes está e não há um conjunto de rotas pré-definido. Em vez disso, a rota é construída enquanto os agentes aprendem. Nenhuma dessas abordagens é baseada no modelo de simulação microscópico. Portanto, tempos de viagem são apenas uma abstração dada por uma função de custo, descrita na Seção 2.4. Além disso, por se tratar de uma modelagem macroscópica, esses trabalhos não podem ser estendidos para incluir agentes semafóricos.

3.2 Controladores semafóricos

Existem muitos trabalhos que lidam com diferentes formas de adaptação ou heurística para melhorar o desempenho dos semáforos. Aqui, revisam-se somente alguns trabalhos que são focados em sistema multiagente e são estritamente relacionados com a

abordagem proposta.

Na maioria dos trabalhos, baseados em aprendizado por reforço para os semáforos, o aprendizado é utilizado pelos semáforos a fim de aprender uma política que mapeia os estados (normalmente as filas nas interseções) para ações (normalmente mantendo/alterando a atual divisão de tempos verdes entre as luzes de cada fase). Há várias formas de abordar essa formulação.

Em (PRASHANTH; BHATNAGAR, 2011) a abordagem é centralizada (uma única entidade detém o MDP para todos os semáforos), uma central recebe informação sobre o tamanho das filas e o tempo corrente de várias vias para realizar a decisão sobre os tempos de cada semáforo. São propostos dois algoritmos para o controle dos semáforos: Q-learning com aproximação de função e gradiente de política baseada em ator-crítico. Os autores compararam os métodos entre si e com a política fixa em redes pequenas (no máximo 5 interseções), observando que o gradiente de política baseada em ator-crítico obteve um melhor desempenho.

Por outro lado, as abordagens (MANNION; DUGGAN; HOWLEY, 2016), (PRABUCHANDRAN; KUMAR; BHATNAGAR, 2015), (DUSPARIC; MONTEIL; CAHILL, 2016), (EL-TANTAWY; ABDULHAI; ABDELGAWAD, 2013), (LÄMMER; HELBING, 2008) e (LE et al., 2015) são descentralizadas. Cada intersecção aprende por si (normalmente utilizando Q-learning). Em (MANNION; DUGGAN; HOWLEY, 2016), os autores usam uma rede viária em grade 3×3 e o simulador SUMO para realizar os experimentos. Além disso, três funções de recompensa diferentes são utilizadas; a primeira baseada na diferença do tamanho das filas anterior e atual; a segunda, baseada no tempo de espera no cruzamento e a terceira é uma combinação das duas primeiras. Os resultados mostram que uma função de recompensa baseada no tamanho médio das filas é melhor para condições de tráfego imprevisíveis.

O trabalho de (PRABUCHANDRAN; KUMAR; BHATNAGAR, 2015) procura encontrar a ordem ótima da sequência das fases do semáforo utilizando o Q-learning. Esse trabalho destaca-se quando lida com a coordenação das intersecções através de um sinal de feedback recebido dos vizinhos. Os experimentos foram realizados em duas redes viárias, com nove interseções e com 20 interseções; os autores mostraram que eles obtiveram resultados melhores que a política fixa.

O trabalho de (DUSPARIC; MONTEIL; CAHILL, 2016) utiliza uma abordagem descentralizada, baseada em aprendizado por reforço. Para isso, o algoritmo Distributed W-Learning (DWL) é implementado no simulador microscópico VISSIM. Além disso, o

trabalho consegue otimizar múltiplos objetivos do gerenciamento do tráfego, bem como aprender em colaboração com outros semáforos, otimizar a seleção e duração das fases. Embora utilize um cenário real, a rede viária utilizada possui apenas 6 intersecções com semáforos.

O trabalho de (EL-TANTAWY; ABDULHAI; ABDELGAWAD, 2013), como os outros trabalhos relacionados nessa subseção, utiliza aprendizado por reforço multiagente. Além de os semáforos aprenderem de forma independente e descentralizada, os semáforos podem aprender de forma coordenada com as intersecções vizinhas. A rede viária utilizada nos testes possui 59 intersecções e, mesmo sendo uma rede real da cidade de Toronto - Canadá, não é muito maior que a utilizada no presente trabalho.

O trabalho de (ARAGHI; KHOSRAVI; CREIGHTON, 2015) compara três métodos de inteligência artificial para controlar a política dos semáforos: Q-learning, redes neurais e sistemas fuzzy. Tais abordagens são comparados com a política fixa. Além disso, uma revisão bibliográfica sobre esses métodos é apresentada. Os testes são realizados no microssimulador Paramics em uma rede viária com apenas uma intersecção. Resultados mostram um melhor desempenho quando é utilizada uma das políticas inteligentes citadas.

No trabalho de (LÄMMER; HELBING, 2008) é proposta uma heurística de otimização descentralizada, onde o controle dos semáforos é baseado em prioridades e na antecipação do fluxo de veículos. Cada fase é servida uma vez, dentro de um intervalo de tempo T. Os autores provam que o algoritmo estabiliza a rede dentro de algumas restrições de intervalo de serviço desejado. No entanto, é necessário ter informações sobre o tempo de chegada dos veículos, antecipando as filas. Uma fila antecipada contém o número de veículos que estão no presente ou que no futuro se juntarão à fila antes da fila existente no cruzamento.

O trabalho de (LE et al., 2015) não utiliza nenhuma forma de aprendizado. No entanto, os autores (LE et al., 2015) apresentam uma abordagem descentralizada que utiliza uma adaptação do algoritmo back-pressure (TASSIULAS; EPHREMIDES, 1992), o qual não requer nenhum conhecimento *a priori*. Os autores propõem uma variação do algoritmo back-pressure com fase cíclica, oposto a outros trabalhos com a mesma política, alegando que assim é mais fácil para o motorista prever qual será a próxima fase. A validação foi feita no microssimulador SUMO com dois cenários diferentes (2 intersecções e 73 intersecções). Resultados mostraram que o algoritmo proposto pelos autores supera outras abordagens, por exemplo, a política fixa.

3.3 Ambas classes de agentes

Esta seção discute alguns trabalhos que lidam com a escolha de rotas e o controle de semáforos como forma de melhorar o fluxo de tráfego. Apenas alguns trabalhos lidam com motoristas e semáforos, adaptando, de alguma forma, no mesmo ambiente.

Como já mencionado, em (WIERING, 2000), as mesmas funções valor utilizadas pelos semáforos são empregadas para otimizar as rotas dos motoristas. Os semáforos precisam saber informações específicas dos motoristas (por exemplo, o destino) para calcular o tempo esperado de espera dos motoristas. No entanto, os motoristas e os semáforos possuem objetivos diferentes. Os efeitos de adaptar motoristas e semáforos são discutidos em (BAZZAN et al., 2007). Os pares de OD são aleatoriamente atribuídos aos motoristas, o que torna a distribuição de tráfego quase uniforme, o que não é muito realista. Os semáforos possuem três estratégias diferentes: política fixa, algoritmo guloso, Q-learning. Os motoristas possuem também possuem três estratégias: seleção aleatória, algoritmo guloso e probabilística. Assim, Q-learning não é usado pelos motoristas. Como já mencionado, o modelo subjacente em ambos (WIERING, 2000) e (BAZZAN et al., 2007) não é totalmente microscópico.

O trabalho de (TAALE; van Kampen; HOOGENDOORN, 2015) não utiliza aprendizado por reforço, mas uma estratégia baseada em back-pressure (TASSIULAS; EPHREMIDES, 1992) para integrar semáforos e orientação de rota (influenciar ou substituir a escolha da rota). Os autores utilizam o back-pressure nos semáforos para determinar a ordem das fases em um ciclo ou para alocar uma fase para cada step. Na orientação de rota, os autores utilizam diferentes variações do algoritmo de back-pressure para fazer os motoristas viajarem na rota que leva para o melhor desempenho. Os autores concluíram que é possível integrar semáforos e orientação de rota baseados no controle de back-pressure. Entretanto, a abordagem proposta foi testada utilizando apenas o modelo de simulação macroscópico. Em relação a isso, os autores, inclusive, mencionam que o simulador não inclui controladores semafóricos.

O trabalho de (SMITH, 2015) analisa a convergência de uma política P_0 para os semáforos (que é estratégia de controle local introduzida em (SMITH, 1980)), os fluxos das rotas e atrasos nos congestionamentos em uma configuração de alocação determinística abstrata. O autor demonstrou que, se o problema inicial for viável e a política P_0 for utilizada para calcular os tempos verdes, a convergência para um conjunto convexo de equilíbrio é certa com uma fila vertical. No entanto, uma central é responsável por atribuir

e controlar o modelo; além disso, não é utilizado um modelo de simulação microscópico.

3.4 Resumo

Existem duas classes de agentes abordadas nesse trabalho. Primeiro, o problema de escolha de rotas engloba a classe dos motoristas e como esses motoristas escolhem rotas para viajar. Existem várias formas de selecionar a melhor rota para melhorar o tempo de viagem dos veículos. Dentre essas formas, algumas possuem rotas pré-computadas e outras rotas são construídas ao longo da viagem. Segundo, os controladores semafóricos podem ser implementados de forma centralizada ou descentralizada. Muitos algoritmos de inteligência artificial já foram estudados para esse propósito. Dentro dos algoritmos de aprendizado por reforço, muitos utilizam o tamanho das filas como percepção de estado.

Poucos trabalhos existentes na literatura tratam duas classes de agentes, adaptando ou aprendendo em um cenário de trânsito. Como já foi mencionado, possuir duas classes de agentes aprendendo é mais realista. Além disso, os trabalhos existentes não utilizam um modelo de simulação microscópico. A partir disso, se faz necessária uma abordagem que simule oferta e demanda, de forma mais realista e descentralizada.

O Capítulo 4 detalha o funcionamento de cada agente e como funciona a interação entre eles, dando detalhes de como a modelagem microscópica influencia nesses métodos.

4 ABORDAGEM PROPOSTA

Nesta seção, discute-se a formulação que inclui ambas as classes de agentes do aprendizado. Uma vez que, não apenas os objetivos de cada classe são diferentes (o objetivo dos motoristas é minimizar o tempo médio de viagem; o objetivo dos semáforos é encontrar um esquema que melhor se ajuste ao fluxo de tráfego e minimize as filas localmente), mas também a natureza das tarefas de aprendizado é diferente; discutir-se-ão os detalhes de cada uma separadamente.

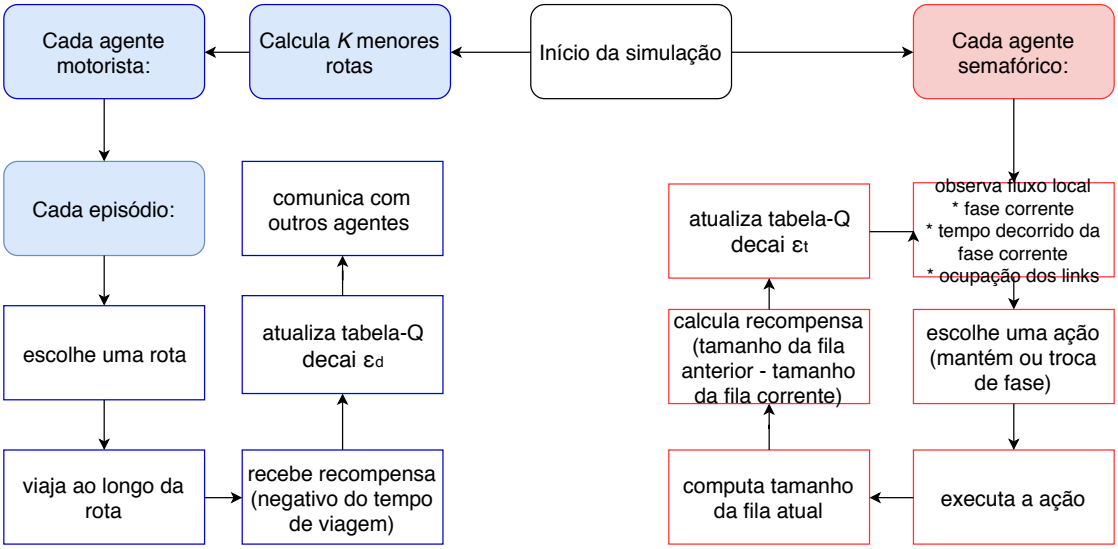
A lista de símbolos (ver página 8) apresenta uma visão geral de todos os parâmetros envolvidos. Aqueles nas primeiras linhas definem a rede viária e/ou a forma como a demanda é distribuída, bem como a forma como a simulação é executada (timesteps, etc.). O resto da lista, como os parâmetros usados para descrever o MDP dos agentes motorista e semafórico, são detalhados mais adiante na Seções 4.1 e 4.3, respectivamente.

Primeiro, será explicado, em alto nível, o que cada classe de agentes faz, com base na Fig. 4.1; depois serão dados detalhes mais aprofundados de cada aprendizado. Conforme se vê no lado direito em azul da Fig. 4.1, cada agente motorista aprende em episódios, lembrando que os episódios não são sincronizados. Em cada episódio, o agente escolha uma ação (uma rota entre a origem e o destino). Quando o motorista chega no seu destino, ele recebe uma recompensa (negativo do seu tempo de viagem). A recompensa é usada para atualizar a sua tabela-Q (através da Eq. 2.1). Após, o motorista que terminou a sua viagem se comunica com outros motoristas, que irão começar suas viagens no próximo timestep.

Simultaneamente, do lado esquerdo em vermelho (na Fig. 4.1), os semáforos aprendem o melhor esquema de tempo de verde para melhorar o fluxo local. Para os semáforos, não há episódios envolvidos; ou seja, a tarefa de aprendizado é de horizonte infinito. Cada semáforo observa o estado das vias controladas por ele e decide se mantém ou troca a fase de verde. Após executar a ação, o semáforo recebe uma recompensa (indicando o quanto o tamanho das filas mudou desde a sua última decisão). Com essa informação, cada semáforo atualiza sua tabela-Q (através da Eq. 2.1) e realiza uma nova decisão.

Agora será explicada, com mais detalhes e pseudocódigos, a simulação e os algoritmos de aprendizado. O Alg. 1 detalha como a simulação é executada; os procedimentos detalhados em relação aos motoristas e semáforos aparecem nos Alg. 2 e Alg. 3, respectivamente; estes são explicados nas próximas seções.

Figura 4.1: Esquema de aprendizado dos agentes motorista (azul) e agentes semafóricos (vermelho)



Fonte: A Autora

O Alg. 1 recebe como entrada o número de timesteps[1] de simulação (tempo de simulação); a rede viária G (isto é, os conjuntos V e L); o número de pares OD, assim como a definição de cada par (uma tupla definindo qual vértice é a origem, qual é o destino, e o número de motoristas em cada par OD), K (número de rotas mais curtas, que define o conjunto de ações dos motoristas); duração máxima do tempo de verde para uma fase $maxVerde$ (para prevenir starvation); o número de fases ρ e o número de motoristas N. Em relação ao último, como explicado anteriormente, N significa que há um fluxo quase constante de N motoristas entrando na rede viária. Na realidade, na microssimulação (em oposição aos modelos macroscópicos), não se pode garantir um fluxo constante dessa natureza. Esta é, de fato, uma das dificuldades encontradas ao usar um microssimulador, porque restrições físicas podem impedir que alguns veículos entrem no link associado às suas origens (por exemplo, quando este link está cheio). Além disso, o fato de que os semáforos regulam o fluxo contribuem para variações no número de veículos, tanto no total quanto nos links. Isso é discutido na Seção 5.1, onde é mostrado um exemplo ilustrativo.

Alg. 1 funciona da seguinte forma. Primeiro, todas as listas usadas na simulação são inicializadas. Nas linhas 8 e 9, para cada par OD, K rotas são computadas usando o algoritmo KSP (veja Seção 4.1). Na linha 11, a tabela-Q de cada agente é inicializada. Na linha 12, o conjunto de motoristas que começam suas viagens no próximo timestep é inicializado.

[1] No SUMO o timestep é utilizado para facilitar a visualização e seu valor padrão equivale a 1s, que pode ser modificado se necessário.

Como no início da simulação, a rede viária está vazia; não há fluxo na maioria das intersecções. Então, por τ timesteps, os semáforos executam uma política de tempo fixo que divide o tempo de verde igualmente entre ambas as fases. Isso é verificado na linha 16, antes de τ timesteps; apenas o algoritmo dos motoristas é executado. A partir de resultados empíricos, notou-se que τ geralmente está na ordem dos minutos; portanto, na mesma ordem que a primeira viagem de um agente motorista. Depois disso, ambos os algoritmos de aprendizado de agentes são executados simultaneamente. Depois, algumas listas são atualizadas: a simulação basicamente continua inserindo veículos na rede (linha 23), verificando aqueles que iniciam suas viagens (linha 24) e controlando o tempo que avançam nos semáforos (linha 25).

4.1 MDP dos agentes motoristas

Como mencionado, a escolha de rotas pode ser modelada como um jogo repetido, e portanto, como um MDP de estado único. O estado nesse caso é o par OD associado ao agente em questão.

A respeito das ações, no vértice de origem cada motorista pode escolher uma das K rotas, $r^k_{OD} \in R$, que o levam de sua origem ao seu destino. Essas rotas são pré-computadas utilizando o algoritmo K *Shortest Loopless Path* (KSP) (YEN, 1971), o qual encontra os K menores caminhos entre a origem e destino em um grafo, desconsiderando ciclos. Este algoritmo usa apenas o tempo de viagem de fluxo livre de cada link como peso; ou seja, ignora os efeitos de congestionamento.

Quando um motorista atinge seu destino, ele recebe um sinal de recompensa, que é o negativo do seu tempo de viagem, e atualiza a tabela-Q usando a Eq. 2.1. Note, no entanto, que no caso de jogos repetidos, a Eq. 2.1 pode ser simplificada eliminando o termo $\gamma max(Q(s', a')$. A formulação é similar com a apresentada por Claus and Boutilier (1998) (e outros), onde há apenas um estado.

A Tab. 4.1, apresenta um resumo do MDP dos motoristas e dos semáforos, que será detalhado mais adiante.

Algorithm 1 Algoritmo principal

1: **procedure** ALG_PRINCIPAL($timesteps$, G, $paresOD$, K, $maxVerde$, ρ, N)
2: motoristas_começando_agora $\leftarrow \emptyset$
3: motoristas_movendo_agora $\leftarrow \emptyset$
4: rotas $\leftarrow \emptyset$
5: stepAtual $\leftarrow 0$
6: fila_step_anterior $\leftarrow \emptyset$
7: fila_step_atual $\leftarrow \emptyset$
▷ K rotas são calculadas para cada par OD usando o algoritmo KSP (YEN, 1971):
8: **for all** od $\in$ pares OD **do**
9: rotas[od] = KSP(K)
10: **end for**
11: cada agente inicializa sua tabela-Q
12: atualiza lista *motoristas_começando_agora* com os motoristas que vão começar suas viagens no próximo step
13: motoristas_movendo_agora.add(motoristas_começando_agora)
14: **while** stepAtual < timesteps **do**
15: stepAtual ++
▷ os procedimentos de aprendizado dos motoristas e dos semáforos são chamados:
16: **if** stepAtual < τ **then**
17: agentes semafórico executam política fixa **and**
18: agente_motorista_alg(motoristas_começando_agora, motoristas_movendo_agora, rotas)
19: **else**
20: agente_semaforico_alg(maxVerde, ρ, fila_step_anterior, fila_step_atual)
21: **and** agente_motorista_alg(motoristas_começando_agora, motoristas_movendo_agora, rotas)
22: **end if**
23: atualiza lista motoristas_começando_agora com os motoristas que vão começar suas viagens no próximo step
24: motoristas_movendo_agora.add(motoristas_começando_agora)
25: atualiza para cada semáforo a lista fila_step_anterior com a lista fila_step_atual
26: **end while**
27: **end procedure**

Algorithm 2 Algoritmo dos agentes motorista

```
1: procedure AGENTE_MOTORISTA_ALG(motoristas_começando_agora, motoristas_movendo_agora, rotas)
2:     for all d ∈ motoristas_começando_agora do
3:         d escolhe uma rota utilizando ε-greedy
4:     end for
5:     for all d ∈ motoristas_movendo_agora do
6:         motoristas movem e incrementam tempo de viagem
7:         if d atinge nodo destino then
8:             computa tempo de viagem da rota
9:             salva recompensa como negativo do tempo de viagem
10:            atualiza tabela-Q usando ação e recompensa
11:            comunica com outro motorista que começará uma viagem no próximo step
12:            decai ε utilizando a taxa de decaimento d_d
13:            motoristas_movendo_agora.remove(d)
14:        end if
15:    end for
16: end procedure
```

4.2 Algoritmo dos agentes motoristas

Uma visão geral do procedimento de aprendizado associado ao algoritmo dos agentes motoristas é apresentada em Alg. 2.

Na linha 3, uma rota é selecionada para cada agente que iniciará sua viagem naquele timestep. Como indicado na linha 3, a estratégia de exploração ε-*greedy* é usada: cada agente escolhe a ação $a = \underset{a}{max} Q(a)$ com probabilidade $1 - \varepsilon_d$, ou seleciona uma ação aleatória com probabilidade ε_d.

Na linha 6, cada motorista move fisicamente no link corrente. Nota-se que aqui encontra-se o poder da microssimulação: os comportamentos individuais, como o car-following e a mudança da pista, além da velocidade desejada e aceleração/desaceleração, determinam a posição do veículo na próxima etapa de simulação[2]. Com base em seu movimento físico, cada motorista aumenta seu tempo de viagem individual. Quando um motorista i atinge seu vértice de destino, ele calcula o tempo de viagem da rota (linha 8), recebe a recompensa (linha 9) e atualiza a sua tabela-Q usando a Eq. 2.1 (linha 10). Após i atualizar sua tabela-Q, ele se comunica com os motoristas no mesmo par OD e que já utilizaram a mesma rota usada por i, e deixará o vértice de origem no próximo step. Essa comunicação é uma mensagem informando o valor-Q da rota feita por i. O motorista j

[2]Embora SUMO seja baseado em uma modelagem contínua, o *step* é usado para fins de visualização.

recebe essa mensagem e utiliza o valor-Q de i para atualizar a sua tabela-Q. Dessa forma, j tem acesso a informações atualizadas sobre o tempo de viagem pelo menos uma de suas K rotas, ajudando j a tomar decisões sobre a próxima rota. Na linha 12, cada agente decai seu valor ε_d usando a taxa de decaimento d_d. Na linha 12, todos os motoristas que terminam suas viagens são removidos da lista $motoristas\_movendo\_agora$.

Tabela 4.1: Comparação entre os MDPs dos agentes motorista e semafórico

MDP	Agentes motorista	Agentes semafórico
Estados	Estado único - par OD	[fase corrente, tempo decorrido da fase corrente, ocupação do link]
Ações	K menores rotas	Manter ou trocar de fase
Recompensa	Negativo do tempo de viagem	tamanho da fila anterior - tamanho da fila corrente

4.3 MDP dos agentes semafóricos

O MDP do semáforo é modelado, principalmente, seguindo trabalhos similares na literatura, e mais especificamente, seguindo (MANNION; DUGGAN; HOWLEY, 2016). Diferentemente do MDP dos motoristas, o MDP de cada semáforo é baseado em estado; portanto, lida-se aqui com um jogo estocástico. A Tab. 4.1, apresenta um resumo do MDP dos motoristas e dos semáforos, bem como suas diferenças em relação aos jogos repetidos e estocásticos. Para cada semáforo, o estado é definido como um vetor $[fase\_corrente, \delta, \lambda_1, ..., \lambda_\rho]$, com $fase\_corrente \in \{1, ...\rho\}$, onde ρ é o número de fases, δ é o tempo decorrido da fase corrente, e λ_i é a ocupação do link para cada fase no timestep em questão.

Por simplicidade, o tempo decorrido da fase corrente δ é discretizado em intervalos de 5 timesteps. A ocupação do link (isto é, número de veículos dividido pela capacidade de armazenamento do link) é então discretizada em 4 intervalos (igualmente distribuídos), sendo $x = \{x \in \mathbb{R} : 0 >= x <= 25\%, 25\% > x <= 50\%, 50\% > x <= 75\%, x > 75\%\}$. Os atributos do estado são comprimidos em um único número utilizando o algoritmo *mixed radix conversion*[3], para servir como índice e acessar as entradas da tabela-Q.

O número de ações é igual ao número de fases, então $|A_t| = \rho$. Seguindo a literatura (em relação à rede viária ser uma rede em grade), há apenas duas fases; portanto as ações são ou manter o verde da fase atual, ou permitir o verde para outra fase. Como costume, chamam-se essas ações de *manter* e *trocar*.

Em relação à recompensa, em cada intersecção, a recompensa é definida como a

[3]O *mixed radix conversion* transforma um vetor para uma representação em um único número.

diferença entre o tamanho médio da fila (AQL) atual e anterior nas vias de aproximação, ou seja, para cada semáforo a recompensa é definida como $R(s, a, s') = AQL_s - AQL_{s'}$.

4.4 Algoritmo dos agentes semafóricos

Uma visão geral do algoritmo dos agentes semafóricos é apresentada em Alg. 3. O algoritmo recebe, como entrada, a duração máxima do tempo verde para uma fase $maxVerde$ (para prevenir *starvation*), o número de fases ρ e duas variáveis que referem ao estados das filas; elas são necessárias para computar a recompensa. Para o propósito da explicação, assume-se que todos os semáforos possuem o mesmo número de fases. No entanto, essa suposição pode ser facilmente relaxada. Na linha 4, cada agente computa seu estado atual e escolhe uma ação usando a estratégia de exploração ε-*greedy* (linha 5): similar aos motoristas, com probabilidade $1 - \varepsilon_t$, a ação $a = \underset{a}{max} Q(s, a)$ é escolhida. O procedimento descrito nas linhas 4 e 5 não deve ser realizados em todos os timesteps da simulação ou a decisão é feita em um período de tempo muito curto. Em vez disso, a literatura sugere que o semáforo compute as filas (definição de estado) e escolha uma ação após um certo intervalo de tempo. Usa-se Δ para denotar esse intervalo (veja a linha 3). Além disso, deve-se respeitar também o tempo mínimo de verde $minVerde$, senão a troca entre as fases pode ser muito rápida e não dar tempo suficiente para o semáforo observar o fluxo local e as mudanças do mesmo para realizar suas decisões.

Se o tempo decorrido δ da fase corrente é maior que o tempo máximo de verde permitido ($maxVerde$), o semáforo é obrigado a trocar de fase (linha 7). Nas linhas 10 e 12, se a ação do agente é *trocar*, ele troca de fase; caso contrário, o tamanho da fila atual é computado, o qual é usado, então, para computar a recompensa (linha 14), como detalhado na Seção 4.3. Finalmente, a tabela-Q é atualizada usando o estado, ação e recompensa, com a Eq. 2.1. Na linha 16, cada agente decai seu ε_t usando a taxa de decaimento d_t.

O procedimento *Computa_estado_corrente*, basicamente, define a fase corrente, o tempo decorrido da fase corrente e a ocupaçãos dos links para cada fase do semáforo (linhas 22 a 26), e mapeia esses números para um inteiro utilizando o *mixed radix conversion*.

Algorithm 3 Algoritmo dos agentes semafórico e a função que computa o estado atual

procedure AGENTE_SEMAFORICO_ALG($maxVerde$, ρ, fila_step_anterior, fila_step_atual)
▷ cada agente computa seu estado and seleciona uma ação a cada Δ intervalo:
    for all tl $\in$ lista_semaforos **do**
        if $tl.\delta$ mod Δ == 0 **and** $tl.\delta >= minVerde$ **then**
            estado_atual=COMPUTA_ESTADO_CORRENTE(tl, ρ)
▷ escolhe uma ação utilizando ϵ-greedy:
            $tl.action$ = EscolheAção(estado_atual,ε_t)
            if $tl.\delta >= maxVerde$ **then**
                $tl.action =' trocar'$
            end if
▷ agente mantém ou troca de fase:
            if $tl.action ==' trocar'$ **then**
                muda a fase do semáforo
            else
                fila_step_atual[tl]= calcula_tamanho_fila(tl)
            end if
            calcula_recompensa(tl, fila_step_aterior, fila_step_atual)
            atualiza tabela-Q utilizando estado, ação e recompensa
            decai ε utilizando a taxa de decaimento d_t
        end if
    end for
end procedure
function COMPUTA_ESTADO_CORRENTE(tl, ρ)
    filas $\leftarrow \emptyset$
▷ o estado é definido pela fase corrente, duração da fase corrente e o tamanho da fila para cada fase:
    idFase = getFase(tl)
    duração = getTempoDecorrido(tl)
    for all fase $\in$ tl **do**
        filas[fase] = calcula_ocupação_fila
    end for
    return encode(idFase, duração, filas)
end function

4.5 Resumo

Neste capítulo o co-aprendizado entre motoristas e semáforos é apresentado. Os agentes motoristas e agentes semafóricos são modelados individualmente. Cada um desses agentes possui objetivos diferentes e a natureza do aprendizado de cada um também é diferente. Os motoristas escolhem uma rota para viajar a cada episódio e objetivam minimizar o seu tempo de viagem individual. Enquanto isso, os semáforos adaptam seus tempos de verde ao tráfego durante todo o tempo de simulação e objetivam melhorar o seu fluxo local.

O modelo proposto é avaliado em um cenário de trânsito microscópico. Nos experimentos, cada agente é avaliado individualmente e as configurações dos parâmetros de cada um são testadas. Após, o co-aprendizado é avaliado e comparado com o caso quando nenhum agente está aprendendo, nesse caso os motoristas escolhem sempre sua menor rota e os motoristas executam uma política fixa.

5 EXPERIMENTOS

No Capítulo anterior são apresentados o co-aprendizado e o funcionamento dos agentes motoristas e semafóricos. Nesse capítulo, a abordagem é avaliada separadamente, em cada um de seus aspectos. A rede viária usada nos experimentos é detalhada, assim como cada experimento relacionado aos motoristas, semáforos e ao co-aprendizado é avaliado para validar a abordagem proposta.

5.1 Cenário estudado

Para realizar a avaliação experimental da abordagem proposta, foi utilizado a rede viária G em grade 6×6, similar à utilizada em (BAZZAN et al., 2007) e maior que a rede em grade utiliza em (MANNION; DUGGAN; HOWLEY, 2016). Em (BAZZAN et al., 2007), os autores explicam porque esse é um cenário realista e complexo, argumentando que há um grande número de rotas que vão de A a B, o que não acontece de verdade. Por exemplo, para alguns pares OD, não havia mais que 3 caminhos possíveis. A razão é que naquele trabalho as vias possuíam apenas um sentido e por isso, menos alternativas. Portanto, foram feitas algumas alterações no cenário. Primeiro, todos os links possuem dois sentidos (o que permite trabalhar com um número maior de rotas possíveis (K) e também é mais realista). Outra alteração, para tornar o cenário mais realista, é em relação aos pares OD, que não são distribuídos de forma aleatória, mas concentram-se em três pares OD: A1 até E5, A6 até E5 e F1 até E5 (veja Fig. 5.1). Dessa forma, mantém-se a ideia subjacente em (BAZZAN et al., 2007), a qual é simular uma situação do pico da manhã, onde a maioria das viagens destinam-se a um distrito comercial.

Vale lembrar que, mesmo que o cenário original fosse mantido, uma comparação completa não poderia ser possível, uma vez que a modelagem é muito diferente. Como foi mencionado, o modelo proposto em (BAZZAN et al., 2007) é baseado em filas (cada link é representado por uma fila; uma vez que há capacidade, o veículo atravessa o link) que abstrai a maioria das interações físicas reais, pelo menos a nível do veículo. Além disso, não é detalhado como o excedente das filas é resolvido; também não reporta quantos veículos estão utilizando a rede viária a cada timestep. Finalmente, na abordagem de (BAZZAN et al., 2007) é apenas reportado o tempo de viagem sobre as últimas cinco viagens e os motoristas não implementam uma abordagem RL; assim, também, impedindo uma comparação direta.

Portanto, para fins de avaliação, foram usados os seguintes casos: (i) os semáforos executam uma política fixa e os motoristas escolhem a sua menor rota entre a origem e o destino; (ii) os semáforos como em (i) e os motoristas aprendem utilizando o Q-learning; (iii) os motoristas como em (i) e os semáforos aprendem utilizando o Q-learning; (iv) motoristas e semáforos aprendem utilizando o Q-learning. Tais casos vão ser referidos como: (i) **Fixo+MenorRota**; (ii) **Fixo+MotorQL**; (iii) **SemafQL+MenorRota** e (iv) **Co-apren**, daqui em diante.

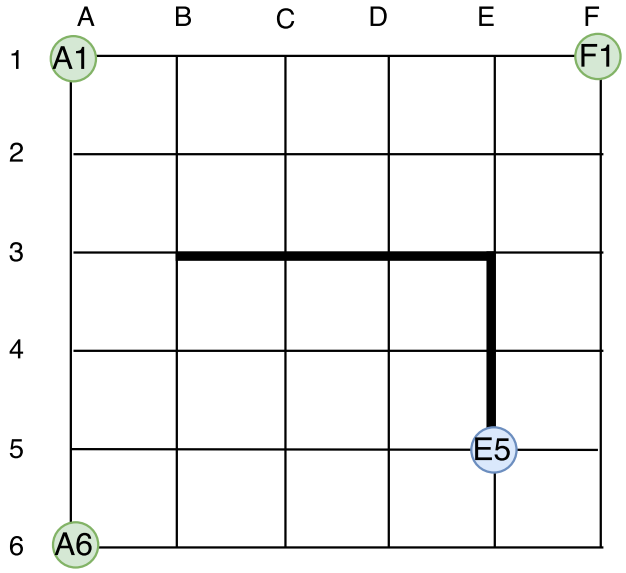
O cenário e a simulação serão descritos a seguir, observando-se que todos os casos de simulação foram repetidos 30 vezes. Assim como mostrado na Fig. 5.1, a rede viária G possui $|V| = 36$ vértices e $|L| = 60$ links, cada link possui 300m. Foram mantidos os valores padrões do SUMO para o tamanho dos veículos e outros parâmetros físicos. O tamanho do veículo é de 5m e a distância mínima entre os veículos é de 2,5m. Portanto, cada link possui uma capacidade de armazenamento de 40 veículos por faixa. Todos os links possuem vias nos dois sentidos com uma faixa para cada sentido, exceto os links principais (linhas destacadas da Fig. 5.1) que são formados pelos links entre os vértices B3 a E3 e E3 a E5, os quais possuem três faixas nos dois sentidos. Consequentemente, sua capacidade é triplicada. Cada intersecção da rede viária, exceto os quatro cantos, possui um semáforo, sendo o número de semáforos $TL = 32$.

A respeito da demanda, como já foi mencionado no Capítulo 4, uma das dificuldades quando se usa um simulador completamente microscópico é de manter um certo fluxo de veículos durante toda a simulação (o que é uma suposição razoável quando se lida com um horário de pico que dura por algum tempo). Note-se que o fato de visar um fluxo quase constante de veículos na rede viária não significa que, localmente, esse fluxo permanecerá constante. Em vez disso, devido à natureza da tarefa (semáforos aprendendo a lidar com a demanda local e os motoristas aprendendo seu caminho na rede), além de o fato de se lidar com um pico da manhã (onde alguns links são muito mais exigidos), a distribuição do fluxo entre os links é muito diferente. De qualquer forma, para manter uma demanda relativamente constante, a abordagem proposta é a seguinte: no início da simulação a rede viária está vazia. Foram realizados experimentos para determinar quando o fluxo estabiliza. Nesses experimentos, nenhuma classe de agentes aprende. Os motoristas utilizam a sua rota mais curta e os semáforos executam uma política fixa que aloca 30 segundos de verde para cada uma das duas fases. Obviamente, essa não é a melhor forma de utilizar a rede viária. Na verdade, o congestionamento impede de outros veículos entrarem na rede (porque os links associados aos vértices origem estão cheios). No

entanto, tal experimento exprime a ideia de quando o fluxo começa a ser quase constante. Conforme mostrado na Fig. 5.2, isso ocorre aproximadamente após uma hora.

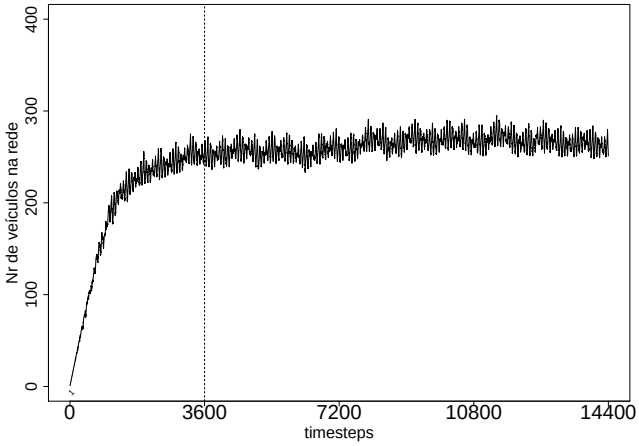
O número de motoristas foi configurado em $N = 400$, mas nem todos os motoristas estão realizando suas viagens ao mesmo tempo nesse caso, pois como todos os motoristas escolhem sempre a mesma rota, alguns links estão muito cheios e os motoristas não conseguem entrar na rede. Como pode-se visualizar na Fig. 5.2 aproximadamente 300 motoristas realizam suas viagens simultaneamente.

Figura 5.1: Rede viária em grade 6×6: todas as vias possuem dois sentidos, com uma faixa em cada sentido. Linhas em destaque indicam uma maior capacidade das faixas



Fonte: A Autora

Figura 5.2: Fluxo de veículos na rede viária ao longo do tempo. Motoritas viajam sempre pela menor rota e os semáforos executam sua política fixa. Linha pontilhada vertical indica 1 hora de simulação



5.2 Métricas

Para avaliar o desempenho da abordagem, utilizou-se o tempo médio de viagem sobre todos os veículos, após cada um finalizar sua viagem. Essa é uma métrica mais global e serve para avaliar o objetivo final, que é minimizar os tempos de viagem. Assim, as métricas-padrão utilizadas na literatura que se concentram somente nos semáforos como, por exemplo, tamanho médio da fila, número de veículos parados ou atraso nos cruzamentos, embora usadas pelos semáforos para determinar seus estados e alterar suas políticas, foram consideradas apenas como locais; portanto, não são usadas como métrica geral. Além disso, no final, os tempos de viagem, de alguma forma, levam em consideração o tempo que os veículos permanecem parados nas filas e os atrasos nos cruzamentos.

Note-se que, mesmo que exista um fluxo quase constante de veículos entrando na rede viária, a cada timestep diferentes quantidades de veículos terminam suas viagens. Pelo menos no início, esse número varia devido a diversas razões. Primeiro, as viagens individuais de diferentes pares de OD consomem tempos de viagem significativamente diferentes. Segundo, o tempo de viagem varia de acordo com a escolha da k-ésima rota (ação) selecionada (por um motorista). Terceiro, esse tempo também varia de acordo com a política que cada um dos semáforos executa quando o veículo percorre sua rota. Se um veículo experimenta muitas filas ou sinais vermelhos nos semáforos, o tempo de viagem aumenta em comparação com outra viagem em que isso não ocorreu.

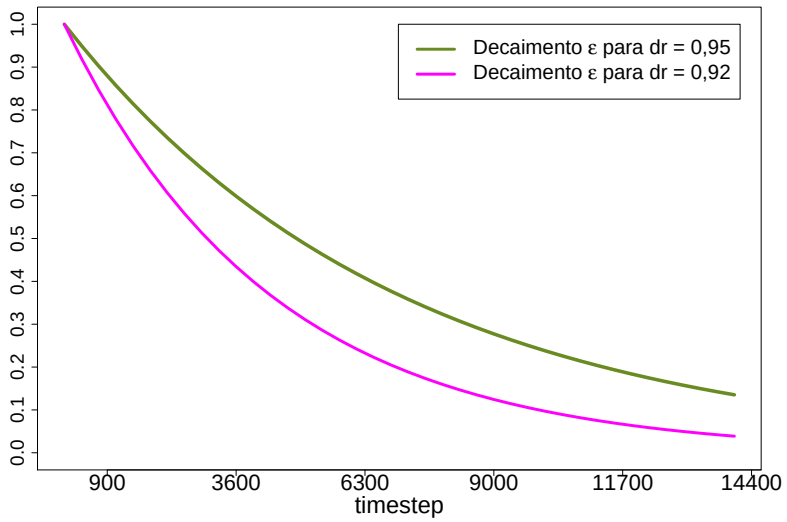
5.3 Configuração

Nessa seção serão explicados os parâmetros listados na lista de símbolos, ao valor referente ao tempo de simulação é atribuído $timesteps$ = 14400 (equivalente a 4 horas). Para investigar como cada valor de cada parâmetro da lista de símbolos influencia os resultados, foi mantido o valor dos outros parâmetros fixos. Além disso, o aprendizado foi desligado nas outras classes de agentes. Por exemplo, para investigar os valores do número de rotas K, a taxa de aprendizado α_d e a taxa de decaimento dr_d para o aprendizado dos motoristas, os semáforos executam sua política fixa. Pela literatura, sabe-se que o valor de K tem uma maior influência e a taxa de aprendizado α_d possui o efeito menor em um longo período de tempo. Portanto, testou-se $K = \{4, 5, 6, 7\}$, notou-se que com valores de K maiores que 7, o tempo médio de viagem aumentava; o mesmo acontecia com valores menores que $K = 4$, para essa rede em específico. Para a taxa

de aprendizado testou-se $\alpha_d = \{0,2; 0,4; 0,6; 0,8\}$. Para a taxa de decaimento, testou-se $dr = \{0,93; 0,95; 0,97\}$. Os melhores valores foram obtidos com $dr = 0,95$ em comparação com os outros dois testados. Então, não foi necessário testar outros valores da taxa de decaimento dr_d. A curva verde da Fig. 5.3 mostra o decaimento de ε_d quando aplicado a taxa de decaimento $dr_d = 0,95$, sendo após uma hora de simulação $\varepsilon_d \approx 0,60$ e após três horas de simulação $\varepsilon_d \approx 0,2$. Portanto, no início há mais exploração das ações e no final mais aproveitamento das melhores ações.

Em relação aos parâmetros que afetam o algoritmo de aprendizado dos semáforos, do acordo com os parâmetros citados na lista de símbolos, o número de fases ρ = 2, pois é comum na literatura quando são usadas redes em grade. Os valores dos outros parâmetros são: δ = 30, τ = 300, $minVerde$ = 10, $maxVerde$ = 180 e $minVeloc$ =10km/h. O valor de τ foi empiricamente calculado de acordo com o tempo que demora para ter veículos em diversos setores da rede viária. Os parâmetros $minVerde$ = 10 e $minVeloc$ = 10km/h, pois são valores comumente usados na literatura. Não é recomendado que veículos fiquem parados muito tempo na fila, para isso é usado o parâmetro $maxVerde$. Em relação aos parâmetros do aprendizado, para a taxa de decaimento foi testado $dr_t = \{0,9; 0,92; 0,94; 0,96; 0,98\}$. Em relação à Equação 2.1, devido à presença de um alto número de agentes, o que torna o problema altamente não estacionário, foi usado $\alpha_t = 0,1$. Pela mesma razão, o fator de desconto precisa ser alto para considerar as recompensas obtidas nos estados futuros, então $\gamma = 0,8$.

Figura 5.3: Variação de ε para taxa de decaimento $dr = 0,92; 0,95$



5.4 Resultados experimentais

Os resultados apresentados representam o tempo médio de viagem nos últimos 30 minutos de simulação, pois assim consegue-se observar o que acontece com o sistema quando semáforos e motoristas estão aproveitando suas políticas. Todos os resultados são comparados com o caso (i) Fixo+MenorRota, no qual os motoristas escolhem sua rota mais curta e os semáforos executam uma política fixa. O tempo médio de viagem para o caso (i) é 379,09 e não possui desvio padrão, pois os veículos e semáforos se comportam de maneira determinística.

Primeiro, serão apresentados os resultados quando somente os motoristas aprendem, caso (ii) Fixo+MotorQL. Após, os resultados referentes ao aprendizado dos semáforos, caso (iii) SemafQL + MenorRota. A melhor configuração de cada um desses casos será utilizada para determinar os parâmetros na co-aprendizagem, caso (iv) Co-apren.

5.4.1 Resultados - motoristas

A Tab. 5.1 mostra o tempo médio de viagem para diferentes valores da taxa de aprendizagem α_d, da taxa de decaimento dr_d e das K rotas, para quando somente os motoristas estão aprendendo.

Os resultados da Tab. 5.1 destacados em negrito são os que apresentaram ser melhores que o resultado obtido no caso (i). Para quase todos os valores apresentados na Tab. 5.1, os melhores resultados obtidos são com $K = 6$. Esses valores são comparados com o resultado que tem o menor tempo médio de viagem que é com a taxa de decaimento $dr_d = 0,95$, taxa de aprendizado $\alpha_d = 0,8$ e $K = 6$; para fins de comparação esse resultado será chamado de P_1. Aplicou-se o teste estatístico t de *Student*, para verificar se os valores são significativamente diferentes. Para o teste estatístico, a hipótese nula se refere às médias serem iguais, ou seja, a média de P_1 é igual a média de outro valor; e a hipótese alternativa que a média de P_1 é menor que o outro resultado comparado. Os valores comparados com P_1 são os valores destacados em negrito da Tab. 5.1. Para todos esses valores, exceto $\{dr_d = 0,93,\ \alpha_d = 0,6$ e $K = 6\}$ e $\{dr_d = 0,95,\ \alpha_d = 0,6$ e $K = 6\}$, o teste resultou que a amostra P_1 é menor com um nível de significância de 5%. Mesmo que o valor de P_1 não tenha sido significativo para os dois valores citados anteriormente, a configuração de P_1 será utilizada para os motoristas no co-aprendizado.

Tabela 5.1: Resultados obtidos para o caso (ii) Fixo+MotorQL, quando somente os motoristas aprendem. Tempo médio de viagem (e desvio padrão) para diferentes valores da taxa de aprendizado α_d, taxa de decaimento dr_d e K

	$\alpha_d = 0,2$	$\alpha_d = 0,4$	$\alpha_d = 0,6$	$\alpha_d = 0,8$
		$dr_d = 0,93$		
K = 4	458,37 (20,37)	465,60 (22,08)	463,23 (15,29)	457,46 (17,15)
K = 5	469,05 (26,46)	431,32 (33,61)	409,52 (52,20)	397,73 (48,46)
K = 6	**324,54 (16,83)**	**320,25 (10,43)**	**316,63 (8,10)**	**326,52 (13,10)**
K = 7	467,61 (31,02)	449,20 (20,82)	434,64 (29,12)	421,74 (48,31)
		$dr_d = 0,95$		
K = 4	454,50 (19,62)	445,83 (16,30)	448,34 (14,24)	446,12 (14,84)
K = 5	487,40 (34,84)	469,58 (36,25)	437,35 (32,22)	434,69 (28,33)
K = 6	**328,97 (15,86)**	**322,25 (9,31)**	**316,91 (8,24)**	***316,35 (5,51)***
K = 7	464,80 (29,85)	454,59 (22,87)	436,27 (34,27)	412,32 (43,61)
		$dr_d = 0,97$		
K = 4	449,73 (11,69)	445,50 (12,12)	442,87 (11,80)	435,77 (10,45)
K = 5	518,02 (39,69)	488,90 (34,79)	484,42 (35,50)	457,69 (37,07)
K = 6	383,08 (29,11)	**368,03 (33,91)**	**348,85 (31,83)**	**326,03 (19,11)**
K = 7	471,70 (27,67)	462,30 (28,75)	449,28 (28,53)	431,62 (28,31)

Tabela 5.2: Resultados obtidos para o caso (iii) SemafQL+MenorRota quando somente os semáforos aprendem. Tempo médio de viagem (e desvio padrão) para $\alpha_t = 0.1$, $\gamma = 0.8$ e diferentes valores da taxa de decaimento dr_t

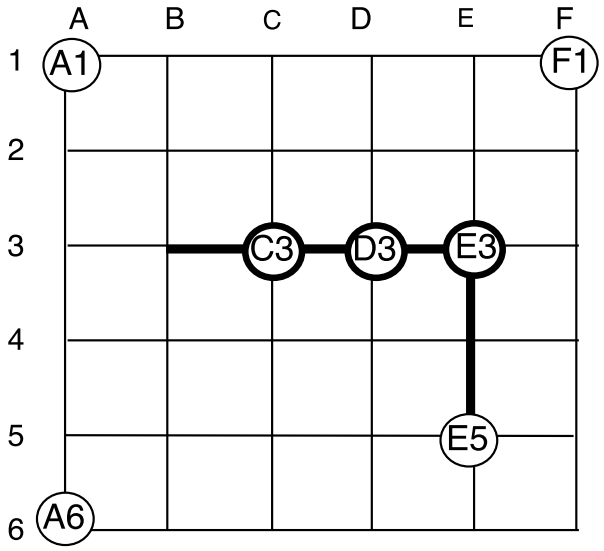
$dr_t = 0,9$	$dr_t = 0,92$	$dr_t = 0,94$	$dr_t = 0,96$	$dr_t = 0,98$
401,53 (50,05)	**388,21 (53,46)**	421,01 (28,13)	465,48 (26,29)	531,81 (20,04)

5.4.2 Resultados - semáforos

A respeito do aprendizado dos semáforos, caso (ii), a Tab. 5.2 apresenta os resultados obtidos com diferentes valores da taxa de decaimento dr. Como explicado anteriormente, os valores da taxa de aprendizado e fator de desconto foram $\alpha_t = 0,1$ e $\gamma = 0,8$, respectivamente, devido ao fato do problema ser altamente não estacionário. O melhor resultado obtido para os semáforos, em tempo médio de viagem, está destacado em negrito é $dr_t = 0,92$. A curva magenta da Fig. 5.3 mostra o decaimento de ε_t quando aplicado a taxa de decaimento $dr_t = 0,92$, mostrando novamente o balanceamento entre exploração e aproveitamento das ações. Para fins de comparação será chamado de P_2 a combinação do melhor resultado obtido pelos semáforos. No entanto, ele não é melhor que o resultado do caso base (i) 379,09.

Na literatura, como citado na Subseção 3.2, quando utilizado algum tipo de algoritmo de aprendizado nos semáforos, há melhoras no fluxo na rede viária. No caso investigado, não houve um melhor desempenho por parte dos semáforos. Isso se deve

Figura 5.4: Rede viária em grade 6 $\times$ 6: cruzamentos destacados C3, D3 e E3 possuem maior fluxo de veículos



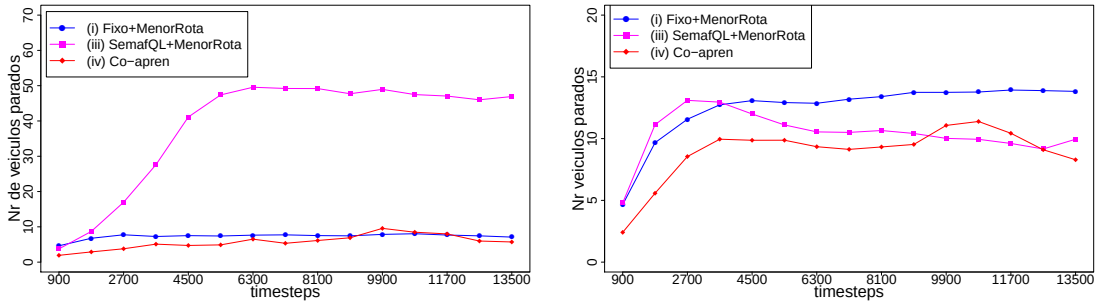
Fonte: A Autora

ao fato de que quando os veículos escolhem sempre sua menor rota e não possuem uma distribuição pela rede viária, algumas partes da rede ficam muito congestionadas a ponto dos semáforos não conseguirem melhorar seu desempenho. Como mostrado na Fig. 5.4, os semáforos nos vértices C3, D3 e E3 são os que possuem maior fluxo de veículos, ficando muitas vezes saturados. Para melhor visualizar o que acontece com os semáforos ao longo da simulação, a Fig. 5.5 mostra o número de veículos parados nas filas dos semáforos; a Fig. 5.5a mostra os semáforos que possuem maior congestionamento e saturação dos links, com mais de 20 veículos na fila. Já a Fig. 5.5b, mostra os semáforos que possuem um congestionamento razoável, menos de 20 veículos na fila. Na curva com quadrados (magenta) do lado 5.5a, nota-se que os semáforos com aprendizado (caso(iii)) não conseguem melhorar o desempenho se comparados com a curva com círculos azuis (caso (i)). Por outro lado, a Fig. 5.5b demonstra como os semáforos com Q-learning, curva com quadrados (magenta), conseguem diminuir o tamanho das filas em relação à curva com círculos (azul). A curva com losângos (vermelho) será explicada na próxima seção.

5.4.3 Resultados - co-aprendizado

Em relação ao co-aprendizado, os parâmetros usados no experimento, para os motoristas e para os semáforos foram aqueles que alcançaram o melhor resultado individu-

Figura 5.5: Número de veículos em fila para duas classificações dos semáforos.



(a) Cruzamentos em que há um grande volume de tráfego

(b) Cruzamentos em que o volume de tráfego é médio

Tabela 5.3: Resultados obtidos nos quatro casos investigados. Valores em tempo médio de viagem.

	Tempo médio de viagem
(i) Fixo+MenorRota	379,09 (0,0)
(ii) Fixo+MotorQL	316,35 (5,51)
(iii) SemafQL+MenorRota	388,21 (53,46)
(iv) Co-apren	285,28 (72,76)

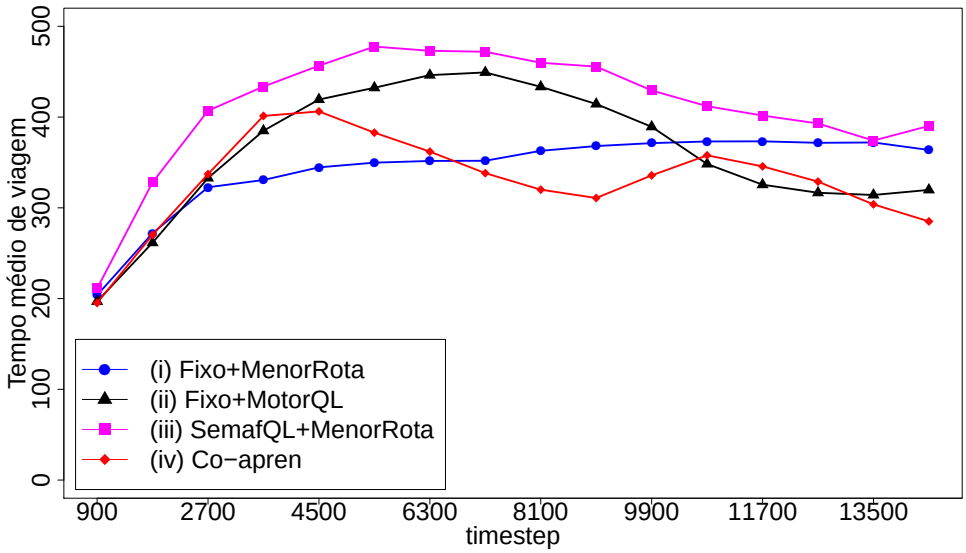
almente. Então, para os motoristas foi utilizada a configuração P_1 e para os semáforos a configuração P_2, detalhadas anteriormente.

A Tab. 5.3 lista os melhores resultados obtidos em relação aos casos (ii) e (iii), detalhados anteriormente, e o resultado em relação ao co-aprendizado. A partir do teste estatístco t de *Student*, testou-se se os valores da Tab. 5.3 eram significativamente diferentes. Com um nível de significância de 5%, o teste resultou que o caso (iv) Co-apren é diferente dos demais. Além disso, o co-aprendizado, caso (iv), é 32,88% melhor que o caso (i), 10,89% melhor que o caso (ii) e 38,08% melhor que o caso (iii).

A Fig. 5.6 compara os quatro casos investigados em relação ao tempo médio de viagem, ao longo da simulação. A curva com círuclos (azul) mostra o caso (i), no qual os semáforos executam a sua política fixa e os motoristas escolhem sempre a menor rota para viajar. Na curva com triângulos (preta) é mostrado o caso (ii); pode-se notar que no início ele é pior que o caso (i), mas ao longo do tempo torna-se melhor. A curva com quadrados (magenta) apresenta o caso (iii), que se mostra pior em relação aos outros casos investigados. Já a curva com losângos (vermelha) mostra o co-aprendizado (caso (iv)), o qual, ao longo da simulação, se mostra melhor que todos os outros casos.

Além de avaliar o tempo médio de viagem dos veículos, que foi escolhido como uma métrica global, avaliou-se também o número de veículos parados nas filas dos semáforos. Como foi explicado anteriormente, os vértices C3, D3,e E3 são os que pos-

Figura 5.6: Comparação entre os quatro casos investigados em termos de tempo médio de viagem



suem um maior fluxo. Na Fig. 5.5a nota-se que a curva com losangos vermelhos que representa o co-aprendizado possui um desempenho quase sempre melhor que o caso (i) Fixo+MenorRota. Já na Fig 5.5b a curva com losangos vermelhos do co-aprendizado possui sempre um desempenho melhor que o caso (i) Fixo+MenorRota (quadrados azuis). Portanto, o co-aprendizado é melhor que os outros casos, não só em termos de tempo médio de viagem, mas também há uma melhora significativa em relação ao tamanho das filas nos semáforos.

6 CONSIDERAÇÕES FINAIS

Este capítulo apresenta uma visão geral das questões discutidas ao longo dos capítulos desta dissertação, com o intuito de oferecer uma visão panorâmica dos tópicos aqui abordados. As principais considerações e perspectivas de continuidade são discutidas.

Os congestionamentos são um fenômeno presente nas redes viárias urbanas. Eles prejudicam os deslocamentos, além de causar vários problemas. Para lidar com esse problema, a literatura utiliza formas clássicas de otimização da rede de transporte, até formas mais complexas de otimizar a demanda ou a oferta do sistema de transporte. Pelo lado da demanda temos o problema de escolha de rotas, no qual os veículos escolhem a melhor rota para trafegar. Já a oferta inclui os controladores semafóricos, os quais se ajustam à demanda para melhorar o fluxo localmente. Muitos trabalhos lidam com uma ou outra forma para melhorar o fluxo na rede de transporte.

O presente trabalho possui uma abordagem diferenciada e mais realista, pois ataca as duas frentes do sistema de transporte: oferta e demanda. Através de sistemas multiagente e aprendizado por reforço multiagente, duas classes de agentes representando a oferta e a demanda se adaptam a fim de melhorar os congestionamentos na rede de transporte. Nesta dissertação, afirma-se que uma abordagem em que ambas as classes de agentes estão aprendendo simultaneamente é mais efetiva.

No entanto, esta abordagem está relacionada a alguns desafios em termos de aprendizado por reforço multiagente. Além do conhecido problema de ações altamente acopladas, nesse tipo de cenário (não importa se apenas motoristas ou semáforos aprendem), um outro desafio considera o fato de que a natureza dessas tarefas de aprendizado é diferente; o objetivo de um é minimizar o tempo de viagem individual e do outro é melhorar as filas localmente. Neste trabalho, propõe-se uma abordagem que mistura a formulação do MDP de jogos repetidos com jogos estocásticos para ambas as classes de agentes, além de discutir como lidar com a questão de aprendizado de horizonte infinito e em episódios não sincronizados (em nível dos agentes motoristas).

Além disso, discute-se vários problemas em relação ao modelo de simulação microscópico. Na verdade, esse modelo de simulação é muito mais desafiador, uma vez que a simples função de custo usada em abordagens macroscópicas ou modelos baseados em filas é substituída pelo movimento físico real dos veículos.

Os experimentos foram realizados utilizando o simulador microscópico SUMO e uma rede viária em grade 6×6 com 32 semáforos, além de uma demanda quase contínua.

Claramente, a abordagem proposta pode ser utilizada em outras redes viárias; no entanto, mais testes seriam necessários.

6.1 Contribuições

Apesar de utilizar duas técnicas já conhecidas na literatura para otimizar a escolha de rotas dos veículos e melhorar o fluxo local dos semáforos, este trabalho apresenta resultados importantes quando combina as duas abordagens. Primeiro, ressalta-se que este trabalho é o primeiro a utilizar duas classes de agente atuando na mesma rede viária, em um cenário microscópico de trânsito. Como apresentado na Seção 3.3, nenhum trabalho utiliza o modelo de simulação microscópico. Isso, além de possuir episódios não síncronos nos agentes motoristas e tarefa de aprendizado de horizonte infinito por parte dos agentes semafóricos.

Dentro desse contexto, a abordagem proposta mostrou-se mais eficiente em termos de tempo médio de viagem, comparado a quando nenhum agente está aprendendo ou quando somente uma classe de agente está aprendendo. Além disso, o co-aprendizado também apresentou melhores resultados quando se compara o tamanho das filas de veículos nos semáforos, mostrando que, assim, os motoristas ficam menos tempo parados.

6.2 Perspectivas de continuação

Como continuação desse trabalho seria interessante investigar uma formulação de jogos estocásticos para a escolha de rotas dos veículos. Dessa forma, ao invés de cada veículo possuir K rotas para escolher, a escolha seria feita a cada vértice da rede viária.

Outra ideia é utilizar outras funções de recompensa para os semáforos. No presente trabalho, o tamanho da fila é usado para calcular a recompensa. Uma função de recompensa interessante para investigar seria o tempo de espera dos veículos no semáforo ou até mesmo uma combinação de ambas as recompensas, como proposto em (MANNION; DUGGAN; HOWLEY, 2016). Dentro dessa ideia de outras funções de recompensa, os semáforos poderiam receber um sinal de feedback dos seus vizinhos, como proposto em (PRABUCHANDRAN; KUMAR; BHATNAGAR, 2015) com o objetivo de haver colaboração entre eles ou gerar uma onda verde.

Por fim, testar a abordagem proposta em outra rede viária é prioridade. No en-

tanto, encontrar uma rede em que se consiga simular veículos e semáforos não é uma tarefa fácil, pois uma rede viária muito grande torna inviável simular todos os semáforos. Além disso, muitas das redes conhecidas na literatura são projetas para serem utilizadas em uma modelagem macroscópica e não conseguem ser transcritas para a modelagem microscópica por falta de dados.

ANEXO A - MODELAGEM DE SIMULAÇÃO SÍNCRONA

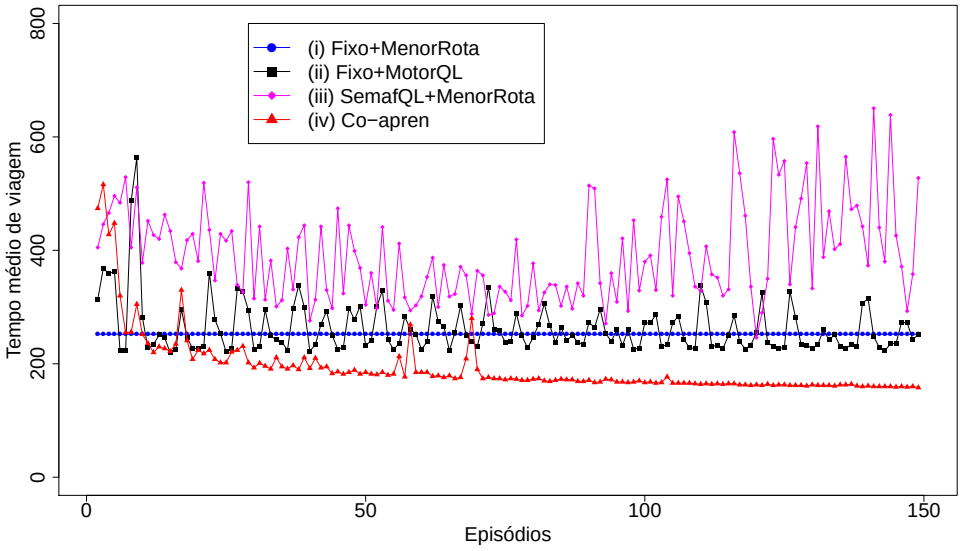
Na abordagem apresentada, a tarefa de aprendizado dos veículos é realizada em episódios assíncronos. Dessa forma, o episódio do motorista i é diferente do episódio do motorista j. Considera-se parte de um episódio o motorista escolher uma rota (ação), viajar nessa rota e, quando ele chegar ao destino e receber a recompensa, o episódio é finalizado. A diferença, aqui, é que os motoristas possuem episódios síncronos. Então, a cada episódio todos os motoristas escolhem uma rota para viajar e realizam as suas viagens. O episódio só termina quando todos os motoristas chegarem em seu destino. Após, eles calculam a recompensa e atualizam as tabelas-Q. No próximo episódio, todos escolhem suas rotas novamente e realizam suas viagens. Neste modelo não há comunicação entre os veículos, uma vez que todos escolhem suas ações ao mesmo tempo.

Nesse contexto, experimentos foram realizados somente para investigar como o co-aprendizado se comportaria mudando a forma dos episódios dos motoristas e removendo a comunicação entre eles. Para isso, utilizou-se a rede em grade 6 $\times$ 6 e os mesmos pares OD explicados anteriormente na Seção 5.1. Os parâmetros do aprendizado dos motoristas usados para esse experimento foram: α_d = 0,8; dr_d = 0,90 e K = 4. Os parâmetros do aprendizado dos semáforos usados foram: α_t = 0,8; dr_t = 0,999 e γ = 0,8. O número de episódios simulados foram 150, pois notou-se que o co-aprendizado estabilizava.

Estudos mais aprofundados são necessários para fazer afirmações precisas. No entanto, pela Fig. 6.1 nota-se que o co-aprendizado (curva com triângulos vermelhos) possui um comportamento melhor que outros casos e, consequentemente, um menor tempo médio de viagem. É similiar ao que acontece na abordagem proposta nesta dissertação.

O caso síncrono e o caso assíncrono não são comparáveis, principalmente, porque a demanda é diferente. Enquanto no caso síncrono a demanda é fixa dentro dos episódios, tendo o mesmo número de veículos que começam e terminam viagens e, além disso, no início e final da simulação existem poucos veículos trafegando pela rede viária. No caso assíncrono, os episódios são diferentes para cada motorista e há um número variado de motoristas que começam e terminam suas viagens ao longo da simulação. Além disso, a demanda é relativamente constante. Outra diferença é o tempo de simulação; no caso síncrono, há um número fixo de episódios e cada episódio simula um pico da manhã. No caso assíncrono, há um tempo fixo de simulação que simula um pico da manhã, em que os motoristas realizam diversos episódios dentro desse tempo.

Figura 6.1: Comparação entre os quatro casos investigados em termos de tempo médio de viagem. Motoristas possuem episódios síncronos



REFERÊNCIAS

ARAGHI, S.; KHOSRAVI, A.; CREIGHTON, D. A review on computational intelligence methods for controlling traffic signal timing. **Expert Systems with Applications**, Elsevier, v. 42, n. 3, p. 1538–1550, 2015.

BARTHÉLEMY, J.; CARLETTI, T. A dynamic behavioural traffic assignment model with strategic agents. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 85, p. 23–46, 2017.

BAZZAN, A. L. C.; GRUNITZKI, R. A multiagent reinforcement learning approach to en-route trip building. In: **2016 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2016. p. 5288–5295.

BAZZAN, A. L. C.; KLÜGL, F. A review on agent-based technology for traffic and transportation. **The Knowledge Engineering Review**, v. 29, n. 3, p. 375–403, 2013. ISSN 1469-8005.

BAZZAN, A. L. C. et al. Adapt or not adapt – consequences of adapting driver and traffic light agents. In: **Proceedings of the 7th Adaptive Learning Agents and Multi-Agent Systems Symposium (ALAMAS)**. [s.n.], 2007. p. 1–8. Available from Internet: <www.inf.ufrgs.br/maslab/pergamus/pubs/alamas07BazzanEtAl.pdf>.

BEHRISCH, M. et al. SUMO - simulation of urban mobility: An overview. In: **SIMUL 2011, The Third International Conference on Advances in System Simulation**. Barcelona, Spain: [s.n.], 2011. p. 63–68.

BURIOL, L. S. et al. A biased random-key genetic algorithm for road congestion minimization. **Optimization Letters**, v. 4, p. 619–633, 2010.

BUŞONIU, L.; BABUSKA, R.; SCHUTTER, B. D. A comprehensive survey of multiagent reinforcement learning. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, IEEE, v. 38, n. 2, p. 156–172, 2008.

CLAES, R.; HOLVOET, T.; WEYNS, D. A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. **IEEE Transactions on Intelligent Transportation Systems**, v. 12, n. 2, p. 364–373, March 2011. ISSN 1524-9050.

CLAUS, C.; BOUTILIER, C. The dynamics of reinforcement learning in cooperative multiagent systems. In: **Proceedings of the Fifteenth National Conference on Artificial Intelligence**. [S.l.: s.n.], 1998. p. 746–752.

DAVIS, S. C.; WILLIAMS, S. E.; BOUNDY, R. G. **Transportation Energy Data Book: Edition 35**. [S.l.], 2016.

DIA, H.; PANWAI, S. **Intelligent Transport Systems: Neural Agent (Neugent) Models of Driver Behaviour**. LAP Lambert Academic Publishing, 2014. ISBN 9783659528682. Available from Internet: <http://books.google.com.br/books?id=fPXpoAEACAAJ>.

DIAS, J. C. et al. An inverted ant colony optimization approach to traffic. **Engineering Applications of Artificial Intelligence**, v. 36, n. 0, p. 122–133, 2014. ISSN 0952-1976.

DUSPARIC, I.; MONTEIL, J.; CAHILL, V. Towards autonomic urban traffic control with collaborative multi-policy reinforcement learning. In: IEEE. **19th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.], 2016. p. 2065–2070.

EL-TANTAWY, S.; ABDULHAI, B.; ABDELGAWAD, H. Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): Methodology and large-scale application on downtown toronto. **Intelligent Transportation Systems, IEEE Transactions on**, v. 14, n. 3, p. 1140–1150, Sept 2013. ISSN 1524-9050.

GALIB, S. M.; MOSER, I. Road traffic optimisation using an evolutionary game. In: **Proceedings of the 13th annual conference companion on Genetic and evolutionary computation**. New York, NY, USA: ACM, 2011. (GECCO '11), p. 519–526. ISBN 978-1-4503-0690-4. Available from Internet: <http://doi.acm.org/10.1145/2001858.2002043>.

GIPPS, P. A behavioural car-following model for computer simulation. **Transportation Research Part B: Methodological**, v. 15, n. 2, p. 105–111, 1981.

KRAUSS, S. **Microscopic Modelling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics**. Thesis (PhD) — DLR (Hauptabteilung Mobilität und Systemtechnik), 1998.

LÄMMER, S.; HELBING, D. Self-control of traffic lights and vehicle flows in urban road networks. **Journal of Statistical Mechanics: Theory and Experiment**, IOP Publishing, v. 2008, n. 04, p. P04019, 2008.

LE, T. et al. Decentralized signal control for urban road networks. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 58, p. 431–450, 2015.

MANNION, P.; DUGGAN, J.; HOWLEY, E. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In: MCCLUSKEY, L. T. et al. (Ed.). **Autonomic Road Transport Support Systems**. [S.l.]: Springer, 2016. p. 47–66.

NARENDRA, K. S.; THATHACHAR, M. A. L. **Learning Automata: An Introduction**. Upper Saddle River, NJ, USA: Prentice-Hall, 1989. ISBN 0-13-485558-2.

ORTÚZAR, J.; WILLUMSEN, L. G. **Modelling Transport**. 3rd. ed. [S.l.]: John Wiley & Sons, 2001.

PRABUCHANDRAN, K. J.; KUMAR, A. N. H.; BHATNAGAR, S. Decentralized learning for traffic signal control. In: **Proceedings of the 7th International Conference on Communication Systems and Networks (COMSNETS)**. [S.l.: s.n.], 2015. p. 1–6. ISBN 9781479984398.

PRASHANTH, L. A.; BHATNAGAR, S. Reinforcement learning with average cost for adaptive control of traffic lights at intersections. In: IEEE. **Procedings of 14th International Conference on Intelligent Transportation Systems (ITSC)**. [S.l.], 2011. p. 1640–1645.

RAMOS, G. de. O.; GRUNITZKI, R. An improved learning automata approach for the route choice problem. In: KOCH, F.; MENEGUZZI, F.; LAKKARAJU, K. (Ed.). **Agent**

Technology for Intelligent Mobile Services and Smart Societies. [S.l.]: Springer Berlin Heidelberg, 2015, (Communications in Computer and Information Science, v. 498). p. 56–67. ISBN 978-3-662-46240-9.

ROESS, R. P.; PRASSAS, E. S.; MCSHANE, W. R. **Traffic Engineering**. 3rd. ed. [S.l.]: Prentice Hall, 2004. 816 p.

SHARON, G. et al. Real-time adaptive tolling scheme for optimized social welfare in traffic networks. In: DAS, S. et al. (Ed.). **Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)**. São Paulo: IFAAMAS, 2017. p. 828–836.

SMITH, M. A local traffic control policy which automatically maximize the overall travel capacity of an urban road network. **Traffic Engineering & Control**, v. 21, n. HS-030 129, 1980.

SMITH, M. Traffic signal control and route choice: A new assignment and control model which designs signal timings. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 58, p. 451–473, 2015.

SUTTON, R.; BARTO, A. **Reinforcement Learning: An Introduction**. Cambridge, MA: MIT Press, 1998.

TAALE, H.; van Kampen, J.; HOOGENDOORN, S. Integrated signal control and route guidance based on back-pressure principles. **Transportation Research Procedia**, Elsevier, v. 10, p. 226–235, 2015.

TASSIULAS, L.; EPHREMIDES, A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. **IEEE Transactions on Automatic Control**, IEEE, v. 37, n. 12, p. 1936–1948, 1992.

TUYLS, K.; WEISS, G. Multiagent learning: Basics, challenges, and prospects. **AI Magazine**, v. 33, n. 3, p. 41–52, 2012.

WATKINS, C. J. C. H.; DAYAN, P. Q-learning. **Machine Learning**, Kluwer Academic Publishers, Hingham, MA, USA, v. 8, n. 3, p. 279–292, 1992. ISSN 0885-6125.

WIERING, M. Multi-agent reinforcement learning for traffic light control. In: **Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)**. [S.l.: s.n.], 2000. p. 1151–1158.

WOOLDRIDGE, M. J. **An Introduction to MultiAgent Systems**. Chichester: John Wiley & Sons, 2009. 461 p. Second edition.

YEN, J. Y. Finding the k shortest loopless paths in a network. **Management Science**, v. 17, n. 11, p. 712–716, 1971. Available from Internet: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.17.11.712>.