

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO**

Uso de Aprendizado por Reforço para
Controle Dinâmico de Tráfego Urbano

Pedro Julio Matuck

Itajubá, 29 de julho de 2018

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO**

Pedro Julio Matuck

**Uso de Aprendizado por Reforço para
Controle Dinâmico de Tráfego Urbano**

Dissertação submetida ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciências em Ciência e Tecnologia da Computação.

Área de Concentração: Inteligência Artificial

Orientador: Prof. Dr. Guilherme de Sousa Bastos

**29 de julho de 2018
Itajubá**

Pedro Julio Matuck

Uso de Aprendizado por Reforço para Controle Dinâmico de Tráfego Urbano/
Pedro Julio Matuck. – Itajubá, 29 de julho de 2018-
69 p.

Orientador: Prof. Dr. Guilherme de Sousa Bastos

Dissertação (Mestrado)

Universidade Federal de Itajubá

Programa de Pós-Graduação em Ciência e Tecnologia da Computação, 29 de julho
de 2018.

1. Aprendizado por Reforço. 2. Processo Decisório de Markov. I. Prof. Dr. Gui-
lherme de Sousa Bastos. II. Universidade Federal de Itajubá. III. Uso de Aprendi-
zado por Reforço para Controle Dinâmico de Tráfego Urbano

CDU 07:181:009.3

Agradecimentos

Há um provérbio africano que diz: "Se quiser ir rápido, vá sozinho. Se quiser ir longe, vá acompanhado". Pois bem, nenhum sonho dessa grandeza se sonha sozinho. Num primeiro momento, esta é uma montanha que o pico está acima das nuvens e você não faz ideia do quanto nem como terá que caminhar. No fundo você sabe que existem desafios, mas não imagina o quão grande eles serão.

Pela minha sorte, existe uma família maravilhosa ao meu lado que me ajudou em cada obstáculo, em cada tempestade vivida durante este longo tempo. Em especial, queria agradecer do fundo do meu coração, minha mãe Lúcia, que sempre esteve pronta para segurar minha mão, oferecer conforto em dias difíceis e me apoiar incondicionalmente em qualquer loucura que eu acredite. Minha irmã Luciene, que muitas vezes me guia e me ajuda a ser uma pessoa melhor a cada dia e meu pai Pedro, que também me apoiou incontestavelmente. Não posso deixar de lado a estrelinha que veio do céu e hoje esta em nossas vidas, meu sobrinho Lucca, uma luz de alegria para toda família. Aos tios, tias e primos que sempre estão ao nosso lado oferecendo uma palavra de conforto, momentos de encontro e alegria, e segurando a barra para nos manter unidos.

Do que seria um homem sem amigos, não é? Sem eles a caminhada seria bem mais árdua. Durante esse tempo, muitas coisas aconteceram e a cada momento lá estava um, oferecendo apoio, motivação e fé. As vezes parece injusto deixar com que nossos amigos dividam nossas angústias, tristezas, sendo que este é o preço que eventualmente pagamos pelas nossas escolhas, mas é exatamente nessa hora que os verdadeiros amigos se colocam ao nosso lado e nos ajudam a carregar o fardo. Nunca me esquecerei daqueles que torceram e sofreram junto a mim, mesmo que no meio da caminhada as coisas tenham dado errado e seguir o próprio caminho foi a melhor opção. Sem cada um de vocês eu não chegaria até aqui.

Mais do que nunca este trabalho me mostrou que além do destino, o importante é a jornada. O aprendizado, a luta pessoal e o amadurecimento, são valores que acima de tudo me ajudarão a enfrentar novos desafios ainda maiores. Obrigado Prof. Guilherme Bastos pelos caminhos apontados ao longo deste desafio e por viabilizar a realização deste projeto.

Tenho o mais claro e profundo sentimento de gratidão por todos vocês!

*"O futuro pertence àqueles que acreditam na beleza de seus sonhos."
(Eleanor Roosevelt)*

Resumo

Motivado por decisões políticas acerca do trânsito da cidade de São Paulo, este trabalho apresenta uma nova forma de se pensar a respeito do tráfego viário urbano. A estrutura de ruas e avenidas das cidades, em muitos casos se mostra propensa a apresentar gargalos e formar congestionamentos, assim, esta tese propõe o uso da técnica de Aprendizagem por Reforço para ajudar uma infraestrutura viária a encontrar um ponto de equilíbrio através do controle dinâmico das velocidades máximas permitidas em cada via. Este controle tem como foco o aumento da vazão de veículos, dado que a malha abstrata utilizada possui um ponto de partida e outro de chegada. Um simulador de mobilidade urbana foi utilizado para auxiliar nesta atividade e, para atender todos os requisitos da pesquisa, uma Aplicação também foi desenvolvida para gerenciar a simulação. Em seguida, o processo de aprendizado é avaliado diante diferentes circunstâncias e uma análise é feita sobre o resultado apresentado. Ao final, pontos positivos e negativos a respeito do trabalho são discutidos.

Palavras-chaves: Processo Decisório de Markov. Aprendizado por Reforço. Tráfego. Veículos.

Abstract

Motivated by politics decisions related to traffic in São Paulo city, this work presents a new way to think about urban road traffic. The cities structure of streets and avenues, frequently show itself prone to have bottlenecks and traffic jams, then, the Reinforcement Learning technique was proposed to help the road infrastructure to find an equilibrium through dynamic control of maximum allowed speed in each road. This control have focus on rising the vehicles flow rate, given that the abstract road network used have an origin and destination points. A urban mobility simulator was used to support on this activity and in order to meet all requirements, also an Aplication was developed to manage the simulation. Then, the learning process was evaluated under different circumstances and an analisys was done about the results. At the end, positive and negative points regarding this work were discussed.

Key-words: Markov Decision Process. Reinforcement Learning. Traffic. Vehicles.

Lista de ilustrações

Figura 1 – Número de habitantes por veículos	16
Figura 2 – Genealogia dos modelos de tráfego.	20
Figura 3 – Transição entre estados a partir de uma ação	25
Figura 4 – Sistema com vários estados e ações	26
Figura 5 – Ciclo envolvendo o ambiente e a política em um MDP	28
Figura 6 – Malhas abstratas construídas a partir da aplicação <i>netgenerate</i> ; da esquerda para direita: grade, teia e aleatória.	33
Figura 7 – Preparação da malha viária a partir das aplicações <i>netgenerate</i> e <i>netconvert</i>	33
Figura 8 – Relação entre tamanho do veículo e distância para outro veículo.	36
Figura 9 – Representação Gráfica da Malha Viária no SUMO	38
Figura 10 – Representação Gráfica das Zonas de Velocidade.	39
Figura 11 – Definição dos elementos formadores da malha viária. A) Via; B) Faixa da Via; C) Segmento de Via.	40
Figura 12 – Estrutura lógica da Aplicação desenvolvida	42
Figura 13 – Dois nós conectados por duas arestas direcionadas representando um Segmento de Via, ou seja, dois cruzamentos conectados por duas vias com sentidos opostos.	46
Figura 14 – Estrutura de dados da tabela-Q.	48
Figura 15 – Número de carros no destino x fluxo de entrada	52
Figura 16 – Número de carros no destino x quantidade de rotas utilizadas	53
Figura 17 – Gráfico comparativo entre os Testes 01, 02 e 03.	55
Figura 18 – Gráfico comparativo entre os Testes 04, 05 e 06.	56
Figura 19 – Gráfico comparativo entre os Testes 07, 08 e 09.	56
Figura 20 – Gráfico comparativo entre os Testes 01, 04 e 07.	57
Figura 21 – Gráfico comparativo entre os Testes 02, 05 e 08.	58
Figura 22 – Gráfico comparativo entre os Testes 03, 06 e 09.	58
Figura 23 – Gráfico comparativo das execuções do sistema sem AR.	59
Figura 24 – Comparação entre os cenários com e sem Aprendizagem por Reforço	60
Figura 25 – Comparação entre as médias das velocidades entre os cenários com e sem AR	61
Figura 26 – Avaliação do desempenho do uso de um sistema treinado em outro ambiente sem treinamento.	62

Lista de tabelas

Tabela 1 – Lista de aplicações da suíte SUMO.	32
Tabela 2 – Tabela de Velocidades por Zona	38
Tabela 3 – Elementos formadores da malha viária.	39
Tabela 4 – Valor de cada ação	41
Tabela 5 – Parâmetros iniciais da aplicação.	43
Tabela 6 – Configuração de parâmetros separados por testes	54

Lista de abreviaturas e siglas

ABNT	Associação Brasileira de Normas Técnicas
API	Application Programming Interface
AR	Aprendizado por Reforço
DLR	Centro Aeroespacial Alemão
GPL	General Public License
GPSS	General Purpose Simulation System
ICT	Information and Communication Technologies
IDE	Integrated Development Environment
ITS	Intelligent Transportation System
MDP	Markov Decision Process
RNA	Rede Neural Artificial
SUMO	Simulator of Urban Mobility
TAZ	Traffic Assignment Zones
TCP	Transmission Control Protocol
TraCI	Traffic Control Interface
UNIFEI	Universidade Federal de Itajubá
VANET	Vehicular Ad Hoc Network
V2I	Vehicles to Infraestructure
V2V	Vehicles to Vehicles
XML	Extensible Markup Language

Lista de símbolos

s	Estado do MDP
S	Conjunto de estados
a	Ação do MDP
A	Conjunto de ações
T	Função de Transição
r	Recompensa
t	Época de Decisão
z	Horizonte
n	Número de nós (cruzamentos) da malha viária
π	Política
Q	Função-Q do algoritmo Q-Learning
α	Taxa de Aprendizagem
γ	Fator de Desconto
ϵ	Política ϵ -greedy

Sumário

1	INTRODUÇÃO	14
1.1	Visão Geral	14
1.2	Motivação	16
1.3	Organização do trabalho	17
2	REVISÃO TEÓRICA	19
3	TRABALHOS CORRELACIONADOS	23
4	APRENDIZADO POR REFORÇO	24
4.1	Definição	24
4.2	Processo Decisório de Markov	25
4.2.1	Definição de um MDP	26
4.2.2	Épocas de Decisão	27
4.2.3	Horizonte	27
4.2.4	Política	27
4.2.5	CrITÉrios de desempenho	28
4.3	Algoritmo Q-Learning	29
4.3.1	Características Principais do Algoritmo	30
4.3.1.1	Exploração versus Investigação	30
4.3.1.2	Fator de desconto	30
4.3.1.3	Condições iniciais da função Q	30
5	MODELAGEM DO PROBLEMA	31
5.1	SUMO - <i>Simulator of Urban Mobility</i>	31
5.1.1	Malha Viária	31
5.1.2	Veículos e Rotas	34
5.1.3	Simulação	34
5.1.3.1	Modelo de Dinâmica de Tráfego	35
5.1.3.2	Modelo de Cruzamento	36
5.1.3.3	Modelo de Mudança de Faixa	36
5.1.4	TraCI - Interface de Controle de Tráfego	37
5.2	Criação da Malha Viária	37
5.2.1	Definição dos Estados	39
5.2.2	Definição das Ações	40
5.2.3	Regras de Decisão	40

5.2.4	Épocas de Decisão	41
5.2.5	Horizonte	41
5.2.6	Episódios	41
5.2.7	Recompensa	41
5.3	Aplicação para controle dinâmico de tráfego	42
5.3.1	Estrutura Lógica da Aplicação	42
5.3.2	Init	42
5.3.3	State	44
5.3.4	Action	45
5.3.5	Routes Manager	45
5.3.6	Vehicles Manager	46
5.3.7	Q-Learning Engine	47
5.3.8	API Connector	47
5.3.9	Log Generator	48
5.4	Configuração do Ambiente	49
5.4.1	Sumo	49
5.4.2	Python	49
5.4.3	IDE Pycharm	49
6	RESULTADOS	51
6.1	Considerações Iniciais	51
6.1.1	Veículos	51
6.1.2	Rotas	52
6.1.3	Política E-Greedy	54
6.2	Teste de Aprendizado pela variação de Parâmetros	54
6.2.1	Testes em função do Fator de Desconto	55
6.2.2	Testes em função da Taxa de Aprendizagem	57
6.3	Teste comparativo em um cenário com e sem Aprendizagem por Reforço	59
6.4	Uso de um sistema treinado em um ambiente sem treinamento	61
7	CONCLUSÃO	63
	REFERÊNCIAS	65

1 Introdução

1.1 Visão Geral

Em 1800, a população mundial era estimada em 1 bilhão de pessoas. Durante 150 anos, até meados do século passado, o número de pessoas no planeta seguiu uma constante crescente, porém lenta, e alcançou 2,5 bilhões de indivíduos. Entretanto, com o desenvolvimento tecnológico, que trouxe avanços na medicina, nutrição e na agricultura, houve uma forte queda na taxa de mortalidade e também um aumento na expectativa de vida humana, e assim, o planeta atingiu o número de aproximadamente 7,5 bilhões de pessoas no ano de 2016 resultando em uma expressiva expansão demográfica. (BONGAARTS, 2009).

Juntamente com esse crescimento populacional, o processo de industrialização ocasionou a migração das populações de áreas rurais para os centros urbanos, favorecendo a expansão física e demográfica das cidades. De acordo com as Nações Unidas, a população urbana se tornou maior que a população rural no ano de 2008. Esse processo evidenciou inúmeros problemas urbanos e também gerou novos desafios para o desenvolvimento sustentável das cidades. A rápida aglomeração de pessoas em ambientes urbanos, criam inevitavelmente espaços desordenados que afetam, dentre inúmeros fatores, a qualidade de vida dos indivíduos. Problemas como: gerenciamento de resíduos, escassez de recursos naturais, poluição e congestionamentos de veículos (CHOURABI et al., 2012), estão na lista dos desafios enfrentados durante o processo de urbanização. Aliado a esses problemas, existem outros desafios de natureza social, econômica e de governança a respeito do desenvolvimento das cidades.

Como forma de enfrentar esses problemas e formalizar soluções para os mesmos, no início do século 21 surgiu o conceito de *smartcities*. Apesar deste conceito ter sido utilizado primeiramente no ECO-92, uma conferência das Nações Unidas ocorrida no Rio de Janeiro para tratar de assuntos relacionados ao meio ambiente, até o presente momento não existe uma definição universal e aplicável de *smartcity*. Inicialmente, as definições se fundamentavam diretamente ao acesso e uso da tecnologia, assim como Hall et al. (2000) definiu que uma cidade inteligente é aquela que “monitora e integra a malha viária, infraestrutura de comunicação, recursos hídricos e energéticos, afim de otimizá-los, planejando atividades de manutenção preventiva e aspectos de segurança, enquanto maximiza o serviço oferecidos aos cidadãos”. Já em 2011, Harrison e Donnelly (2011) sugeriram que uma *smartcity* conecta as infraestruturas física, social, econômica e também a infraestrutura de tecnologia da informação com o intuito de elevar a inteligência coletiva de uma cidade. De acordo com Kourtit, Nijkamp e Partridge (2013)

Smartcities são o resultado de intenso conhecimento e estratégias criativas com foco no aumento do nível socioeconômico, ecológico, logístico e no desempenho competitivo das cidades, assim como, *smartcities* são baseadas na miscigenação de capital humano, social, estrutural e empreendedor.

É notório que a cada definição de *smartcity* novos agentes e conceitos são agregados de forma que, as definições evoluíram da dimensão tecnológica, para também a dimensão social e institucional, envolvendo o capital humano e a governança das cidades. Desta forma, as definições mais recentes estão alinhadas com uma abordagem mais antropocêntrica, sugerindo que, cidades que almejam um crescimento econômico e aumento da qualidade de vida, necessitam investir no desenvolvimento humano e social, além da infraestrutura de tecnologia da informação (IRUNGBAM, 2016).

Apesar das variadas tentativas de definição do que é uma *smartcity*, a aplicação destes conceitos se baseia em estratégias para atender as necessidades e desafios enfrentados por cada cidade. O termo *smart* pode adotar diferentes conotações de acordo com o contexto no qual é inserido. Por exemplo, quando está relacionado a economia, uma *smartcity* é aquela que possui indústrias de tecnologia, gera muitos empregos e favorece fortemente a renda per capita da cidade. Por outro lado, quando se trata de educação, uma *smartcity* está relacionada com o nível educacional e cultural de seus cidadãos. Este termo também pode ser aplicado na política, aferindo a capacidade de um governo em gerir a cidade de forma transparente. Segundo Giffinger et al. (2007), uma *smartcity* possui seis características em comum:

- *Smart Economy*
- *Smart People*
- *Smart Governance*
- *Smart Mobility*
- *Smart Environment*
- *Smart Living*

Cada uma dessas características possuem fatores que permitem a avaliação de estratégias adotadas quando aplicadas a problemas urbanos. Paralelamente aos pesquisadores, grandes empresas de tecnologia, como CISCO (CISCO, 2017), IBM (IBM, 2017) e Ericsson (ERICSSON, 2017), também adotaram definições particulares de *smartcities* com o propósito de apresentarem soluções para os desafios encontrados.

Dentre todos os aspectos de uma *smartcity*, a mobilidade é um problema presente na vida de quase todos os cidadãos que dependem do trânsito para trabalhar, estudar,

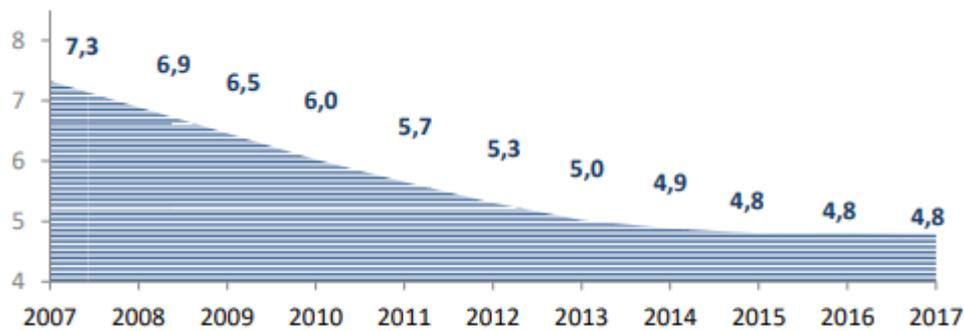


Figura 1 – Número de habitantes por veículos

Fonte: <<https://www.sindipecas.org.br/area-atuacao/?co=s&a=frota-circulante>>

entre outras atividades. O que era particularidade de grandes cidades, o congestionamento já faz parte da vida de pequenos e médios centros urbanos.

Atualmente, a frota brasileira é composta por mais de 97 milhões de veículos (DENATRAN, 2018). Segundo (SINDIPECAS, 2018), aproximadamente 45% do total de veículos registrados estão em circulação pelas cidades resultando em aproximadamente um veículo para cada 5 pessoas conforme visto na Figura 1.

A atual malha viária construída nas cidades certamente não suporta o volume de veículos existente nas ruas, sem contar que ruas e avenidas não são facilmente modificadas para suportar mais veículos, devido aos custos e a restrição do espaço urbano. Esses fatores causam congestionamentos que aumentam o tempo de cada indivíduo no trânsito além de favorecer a elevação do número de acidentes.

Em face destes problemas, este trabalho tem como foco alterar dinamicamente a velocidade máxima permitida de cada via que compõem uma malha viária abstrata, através de um agente controlado pela técnica de Aprendizagem por Reforço. A técnica de aprendizagem utilizada tem como propósito avaliar e escolher as melhores configurações de velocidades das estruturas de vias com o objetivo de otimizar o fluxo de veículos nas ruas e diminuir o tempo médio em trânsito.

Para realização deste experimento, um simulador de trânsito foi utilizado. A escolha deste simulador se deu pelo fato de ser uma ferramenta *opensource*, acessível e projetada para atender diferentes tipos de problemas de mobilidade. Assim, associado ao simulador adotado, uma aplicação foi desenvolvida para implementar a técnica de Aprendizagem por Reforço ao cenário proposto.

1.2 Motivação

É latente a quantidade de problemas causados aos ambientes urbanos com o aumento do número de veículos circulando pelas ruas, além de diversos serviços críticos

à sociedade, que dependem de um trânsito funcional, como: Departamentos de Polícia, Corpos de Bombeiros e Hospitais, através do uso de ambulâncias, outras pessoas utilizam as ruas para irem ao trabalho, à escola, ou mesmo para momentos de lazer.

Quando cabe aos governantes tomarem decisões para melhorarem o funcionamento do trânsito, em muitas oportunidades as mudanças realizadas se mostram desastrosas por carecerem de uma análise detalhada da situação (USP, 2017). Decisões desta natureza em muitos casos possuem efeitos contrários a real necessidade da população, como pode ser visto em G1 (2017), Folha (2017) e REDE BRASIL ATUAL (2017).

O mundo tem presenciado o surgimento de inúmeras tecnologias que visam melhorar o funcionamento do trânsito como um todo. Por um lado, empresas como a TESLA, VOLVO e Google, oferecem carros semiautônomos que auxiliam o condutor a tomar decisões durante o trajeto. Aliado a estes carros, tecnologias como V2V e V2I têm sido cada mais discutidas no meio acadêmico, tendendo assim a se tornarem populares em alguns anos. E um terceiro elemento, não menos importante, é a popularização de técnicas de Inteligência Artificial que estão cada dia mais presentes na vida das pessoas. Um exemplo disso é o fato de que atualmente grande parte dos condutores dirigem sob o auxílio dos aplicativos de navegação através dos *smartphones*.

Os artigos publicados referentes ao tráfego urbano, principalmente aqueles que visam associar técnicas de computação a elementos de trânsito, em sua grande maioria se dedicam à alternativas aos Sistemas de Semáforos atuais ou mesmo formas de aumentar a autonomia dos veículos, esquecendo assim da infraestrutura das vias. Diante do surgimento de inúmeras tecnologias, associadas as já existentes atualmente, seria possível então alterar dinamicamente a velocidade máxima permitida das ruas a fim de encontrar um melhor equilíbrio ao trânsito, reduzindo gargalos e aumentando o fluxo de veículos? Com o propósito de responder esta indagação, esta dissertação apresenta uma nova forma de se pensar em trânsito.

1.3 Organização do trabalho

Este trabalho inicia com uma revisão histórica e contextual das pesquisas relacionadas a definição, análise e simulação de tráfego, apresentados no Capítulo 2. Uma amostra de trabalhos são citados dando a dimensão das aplicações compreendidas nesta área. No Capítulo 3, há uma revisão teórica a respeito da técnica de Aprendizagem por Reforço assim como uma breve discussão sobre Processos Decisórios de Markov. O Capítulo 4 trata da modelagem do problema a fim de construí-lo com o propósito de atender a indagação desta dissertação. Para modelá-lo, foi necessário criar uma Aplicação que, em conjunto com o simulador utilizado, oferecesse todos os recursos necessários para o desenvolvimento da pesquisa. O Capítulo 5 apresenta os resultados e uma análise a respeito.

Por último, o Capítulo 6, trata das considerações finais e a conclusão deste trabalho.

2 Revisão teórica

As primeiras pesquisas relacionadas com a descrição, simulação e previsão de tráfego surgiram na década de trinta com o Diagrama Fundamental do Modelo de Fluxo de Tráfego proposto por Greenshields et al. (1934). Este modelo tinha como principal preocupação descrever a relação de distância e velocidade entre os veículos. No ano seguinte, em 1935, o mesmo autor adicionou a este modelo o que ele chamou de densidade, ou seja, a relação entre o tamanho da via e o número de carros ocupantes (GREENSHIELDS et al., 1935).

A partir do Diagrama Fundamental, houve a necessidade de considerar nos modelos os fenômenos físicos pertinentes ao tráfego de veículos. Assim, surgiram outras três famílias de modelos: Microscópica, Mesoscópica e Macroscópica (WAGENINGEN-KESSELS et al., 2015). Cada família apresenta um modelo de fluxo distinto, onde o modelo Microscópico tem como finalidade descrever de forma discreta os elementos referentes ao tráfego, tais como: veículos, vias, semáforos e a relação espaço-temporal entre eles. O modelo Macroscópico descreve o tráfego como um fluxo contínuo onde apenas variáveis agregadas são consideradas, como: densidade, fluxo e velocidade. Na lacuna existente nos modelos anteriores, encontra-se o modelo Mesoscópico, que por sua vez descreve os veículos de forma contínua, através de distribuições de probabilidade, porém permite com que regras de comportamento sejam estabelecidas.

O modelo Microscópico tem como base a “Dinâmica de Tráfego de Veículos” (*car-following model*). A primeira proposição de dinâmica foi apresentada por Pipes (1953). Neste modelo, os outros veículos pertencentes à frota devem ajustar suas próprias velocidades baseados em uma distância segura do veículo líder. Outros modelos que utilizaram a distância segura a outro veículo também foram apresentados em (GIPPS, 1981; NEWELL, 2002).

Uma segunda vertente da Dinâmica de Tráfego consiste nos modelos baseados em estímulo-resposta. Este modelo prega que o veículo pode ser modelado baseado em três estímulos (CHANDLER; HERMAN; MONTROLL, 1958):

1. Sua própria velocidade;
2. Distância para o líder;
3. Velocidade relativa ao líder.

Apesar das inúmeras tentativas de aprimorar e refinar este modelo, Brackstone e McDonald (1999) concluíram que a dificuldade e a ambiguidade dos parâmetros que definiam

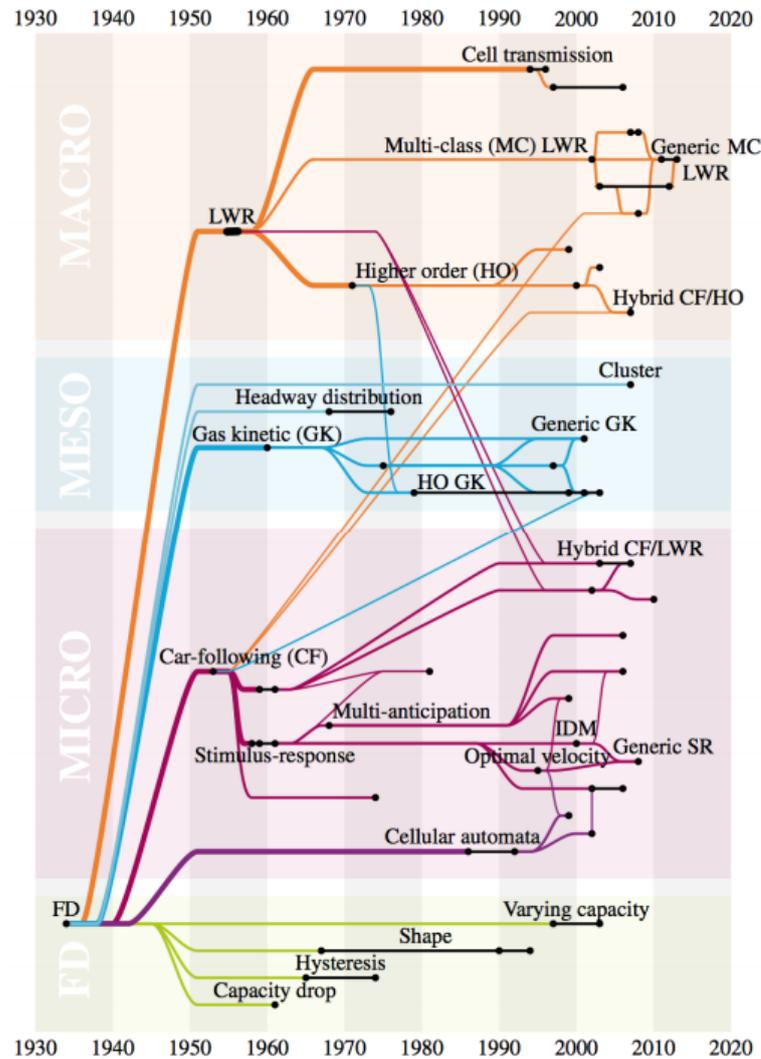


Figura 2 – Genealogia dos modelos de tráfego.
 Fonte: (WAGENINGEN-KESSELS et al., 2015)

os estímulos fizeram desta abordagem pouco utilizada.

A partir da criação dos modelos de tráfego, que podem ser observados na figura 2, foi possível analisar o comportamento dos veículos sob aspectos matemáticos e simular estas informações a fim de obter respostas. No entanto, a complexidade destes sistemas exigia que um grande volume de cálculos fossem realizados ao mesmo tempo. Desta forma, o desenvolvimento da simulação de tráfego só foi possível com o surgimento dos primeiros computadores de propósito geral, por volta da década de 50 (LIEBERMAN, 2014). Nos anos seguintes, diferentes linguagens de programação surgiram para atender diferentes propósitos, inclusive, linguagens destinadas a simulação como: GPSS (*General Purpose Simulation System*), SIMSCRIPT e SIMULA as quais se baseavam em linguagens mais tradicionais: FORTRAN e ALGOL por exemplo.

Na medida em que os computadores evoluíam e se tornavam mais populares, novos modelos de fluxo puderam ser representados e estudados através da simulação. Parale-

lamente, novas Tecnologias de Informação e Comunicação (ICT) foram desenvolvidas e integradas ao estudo de problemas de tráfego. A partir desta união, surgiu o conceito de Sistemas de Transporte Inteligente (ITS) o qual usa tecnologias de comunicação para coletar, analisar e processar informações a partir de cenários reais (DIMITRAKOPOULOS; DEMESTICHAS, 2010). Este conceito está intimamente ligado ao conceito de *smartcities*, dado que, problemas de tráfego afetam a qualidade de vida das pessoas além de causarem impactos sócio-econômicos e ambientais no local onde ocorrem.

Todos os avanços tecnológicos citados permitiram que ferramentas mais robustas fossem desenvolvidas e problemas mais complexos fossem abordados. Nestas circunstâncias, dentro do universo do modelo Microscópico, ferramentas como: Paramics(PARAMICS, 2018), VISSIM(PTV-VISSIM, 2018) e SUMO(SUMO, 2018) são utilizadas atualmente para analisar e prever situações reais de tráfego.

Por sua vez, o SUMO (*Simulator of Urban Mobility*) é um pacote de simulação *opensource* contendo várias ferramentas que auxiliam nessa tarefa. Krajzewicz et al. (2012) destacaram os principais tópicos de pesquisa que utilizam o SUMO como plataforma:

- Comunicação veicular;
- Escolhas de rota e navegação dinâmica;
- Algoritmos para semáforos;
- Avaliação dos sistemas de vigilância de tráfego.

A Comunicação Veicular tem como foco o estudo de uma rede de comunicação V2X (veículo-veículo ou veículo-infraestrutura) aplicada na busca de soluções para problemas de tráfego. Neste contexto, identifica-se uma grande atenção para redes VANET (*Vehicular Ad Hoc Network*) (EMMELMANN; BOCHOW; KELLUM, 2010) e trabalhos que utilizam o SUMO em (PIORKOWSKI et al., 2008; KARNADI; MO; LAN, 2007).

Escolhas de rota e Navegação Dinâmica, que é o foco deste trabalho, avaliam as tomadas de decisões dos veículos baseadas na origem e destino do fluxo de tráfego, tanto no campo teórico como prático, através do desenvolvimento e aplicação de algoritmos para escolhas de rota.

Outra área que recebe grande foco e suporte dos simuladores do tipo microscópico é a análise de Algoritmos para Semáforos. Em um cenário inteligente, espera-se que os semáforos se adaptem a diferentes situações de trânsito. O SUMO em si não é amplamente utilizado para estudos com semáforos em cenários reais, no entanto, a API de interação em tempo-real com a simulação e a rápida execução da ferramenta oferece suporte para testes em cenários abstratos.

Por sua vez, simuladores também são utilizados para Avaliação de Sistemas de Vigilância empregados no trânsito. Neste caso, as ferramentas de simulação fornecem dados a respeito dos veículos e suas respectivas rotas e, integrado a outras ferramentas de análise, é possível avaliar o grau de eficiência do modelo de vigilância utilizado conforme visto em (KEREKES et al., 2009).

3 Trabalhos Correlacionados

Técnicas de Inteligência Artificial são amplamente utilizadas como formas de auxiliar na solução de problemas no que diz respeito ao tráfego urbano. Khatib et al. (2015) e Saha, Roy e Sinha (2015) adotaram Redes Neurais Artificiais (RNA) para avaliar o comportamento e a distribuição de veículos dentro de uma VANET. Preocupado com a segurança da rede formada por carros autônomos, (ALHEETI; GRUEBLER; MCDONALD-MAIER, 2015) propõem uma RNA para detectar ataques e invasões na referida rede de comunicação.

No que diz respeito às rotas dos carros em ambientes urbanos, Dallmeyer et al. (2015) utilizaram algoritmos de Colônia de Formigas como base para propor um roteamento dinâmico de veículos comparando o desempenho com outros algoritmos de busca. Zheng e Liu (2006) aplicaram lógica *Fuzzy* associada a algoritmos genéticos para proposição de um modelo de otimização de rotas.

Silva, Costa e Bastos (2013) e Costa e Bastos (2012) adotaram AR para o estabelecimento de Semáforos Inteligentes visando aumentar o fluxo de carros em cruzamentos. Já em (OH; LEE; CHOI, 2000), a mesma técnica, auxiliada por algoritmos de processamento de imagem, foi utilizada para a proposição de um condutor autônomo de veículos em estradas de alta velocidade. Através do uso de redes V2V, Desjardins e Chaib-Draa (2011) propõem uma nova abordagem para carros autônomos através de Controles Adaptativos de Velocidade (ACC) cooperativos. Medina, Hajbabaie e Benekohal (2010) propôs uma aplicação utilizando AR para melhorar o fluxo de vias arteriais em ambientes urbanos.

Um sistema de tráfego é composto por inúmeros agentes, carros, pedestres, entre outros. Bazzan (2009), analisa os problemas de tráfego urbano a partir do uso de múltiplos agentes adotando o AR como método de aprendizado. A abordagem de múltiplos agentes associada a técnica de Aprendizado por Reforço também se estende a proposição de semáforos inteligentes (ABDOOS; MOZAYANI; BAZZAN, 2011) e (EL-TANTAWY; ABDULHAI; ABDELGAWAD, 2013). Cheng e Nguyen (2011) e Gan et al. (2013) avaliaram as melhores condições para as cobranças de tarifas dos serviços de táxi, considerando as conjunturas pertinentes ao tráfego urbano das cidades. O uso de sistemas multiagentes também podem ser aplicados ao gerenciamento do tráfego e no roteamento dinâmico dos veículos, conforme observado em (ADLER et al., 2005).

4 Aprendizagem por Reforço

4.1 Definição

Dado um determinado ambiente, um conjunto de ações e recompensas relacionadas às ações tomadas, o Aprendizado por Reforço (AR) é uma técnica de Aprendizado de Máquina que avalia como um algoritmo, também denominado agente, escolhe suas ações neste ambiente proposto, visando o acúmulo máximo de recompensas. O agente em questão aprende por interação direta com o ambiente em que se encontra, recebendo reforços positivos ou negativos de acordo com os resultados produzidos pelas ações tomadas (COSTA; BASTOS, 2012). Um dos modelos matemáticos utilizados como base para o domínio dos problemas nos quais se faz pertinente o uso de Aprendizado de Máquina é o Processo Decisório de Markov (SILVA; COSTA; BASTOS, 2013).

O ambiente é modelado através de um conjunto de estados discretos S e um conjunto de ações A . Em muitos casos, o agente assume a capacidade total de observação, ou seja, existe um conhecimento prévio de todos os estados do ambiente, ou, quando não é possível devido a dimensão do problema, o agente deve assumir capacidade parcial de observação. Dado um determinado estado $s \in S$, durante um intervalo discreto t , o agente interage com o ambiente e recebe uma recompensa r_t . Em seguida, uma ação a_t ($a \in A$) é escolhida a partir de um conjunto possível de ações e então o agente se move para o estado s_{t+1} estabelecendo a transição (s_t, a_t, s_{t+1}) . O objetivo do algoritmo de Aprendizagem por Reforço é coletar o máximo de recompensas possíveis. O agente pode visitar estados e escolher qualquer ação do histórico registrado. Para estes casos, uma política pode ser definida a fim de propor certa aleatoriedade na escolha das ações permitindo que o agente tenha um comportamento “guloso” ou não.

O AR requer um mecanismo ágil para exploração do ambiente. Dependendo da dimensão do problema, o que resulta em muitos estados, a escolha puramente aleatória de ações se apresenta como uma alternativa ruim no que diz respeito a *performance*. Uma estratégia adotada para o balanceamento de investigação e exploração é o algoritmo *ϵ -greed*. Este método sempre escolhe uma ação ótima, ou seja, aquela que fornece a maior recompensa. Associado ao *ϵ -greed*, uma outra estratégia que pode ser adotada é a Exploração de *Boltzmann*. Este método escolhe entre uma ação aleatória ou aquela que fornece a melhor recompensa esperada de acordo com uma probabilidade $p \rightarrow [0, 1]$ de escolher a melhor ação, sendo $1 - p$ a probabilidade de tomar uma ação aleatória. O sistema pode inicialmente adotar um valor baixo para p , a fim de garantir maior capacidade de exploração, no entanto, esta mesma probabilidade pode crescer com o tempo, fazendo que o sistema transite de um modo de exploração para um modo de investigação (KAELBLING;

LITTMAN; MOORE, 1996).

Ao final de todo este processo o algoritmo fornece uma política π , a qual é um conjunto de ações tomadas com as recompensas maximizadas. A política é modelada como um mapa dado pelo conjunto de estados S e pelo conjunto de ações A ($\pi : S \times A$) e assim, a partir da política, dado um estado s , é possível determinar qual ação retornará a melhor recompensa possível.

4.2 Processo Decisório de Markov

Existem várias situações no cotidiano onde a tomada de decisões sequenciais é fundamental para a execução e compreensão de uma determinada atividade. Tratamento de doenças (AKHAVAN-TABATABAEI; SÁNCHEZ; YEUNG, 2017), jogos de tabuleiro (FU, 2017), mudanças genéticas (PACIONI et al., 2017), comportamento de preços de uma determinada ação da bolsa de valores (CHANG; LEE, 2017), são exemplos que podem ser analisados a partir de um modelo de tomada de decisões em sequência.

O Processo de Decisão de Markov é um modelo matemático que pode ser aplicado em situações onde se faz necessária a execução de ações consecutivas. O momento prévio à execução de uma ação é chamado de estado. A partir deste estado, uma ação deve ser tomada (de forma estocástica ou determinística) levando a um estado seguinte conforme modelo ilustrado na figura 3.

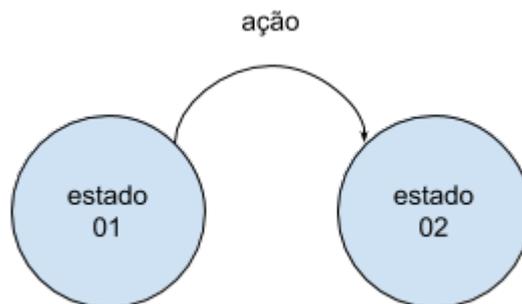


Figura 3 – Transição entre estados a partir de uma ação

Os problemas que podem ser modelados como um Processo de Decisão de Markov consiste em um sistema com vários estados e ações as quais normalmente modificam o estado do sistema. As ações tomadas em um determinado estado conduzem o sistema a um novo estado o qual pode ou não gerar algum resultado. Este resultado normalmente é representado por um custo ou uma recompensa. A solução de problemas desta natureza implica em encontrar uma política que indique qual ação tomar, dado o estado presente, com o objetivo de minimizar o custo esperado ou maximizar a recompensa esperada. Esta política é denominada política ótima.

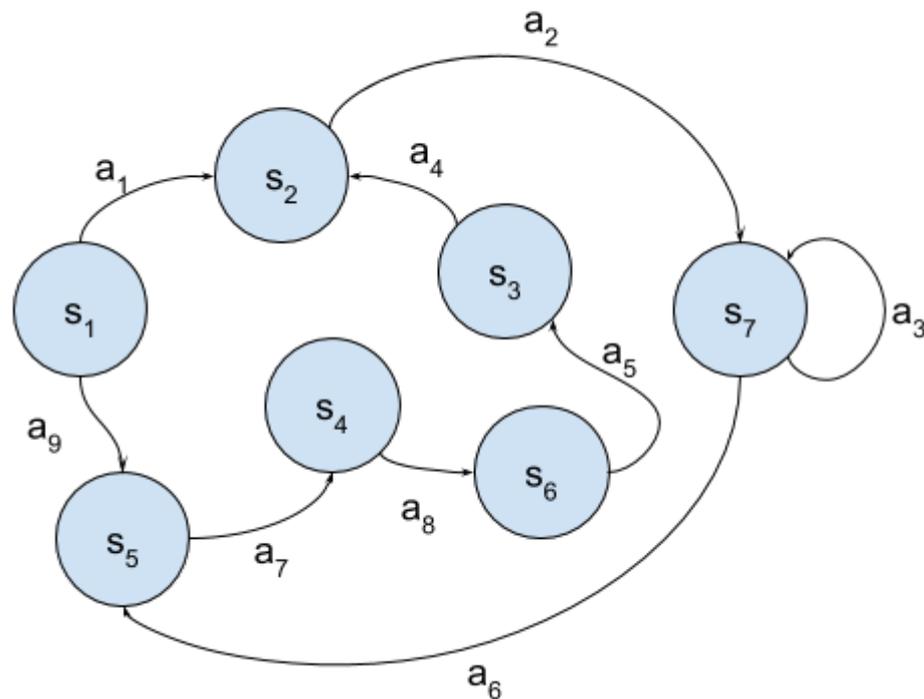


Figura 4 – Sistema com vários estados e ações

O processo de tomada de decisões envolvido neste modelo é denominado de Markov pois obedece a propriedade de Markov, ou seja, o efeito de uma ação em um estado depende apenas da ação e do estado atual do sistema, independentemente de qual caminho o sistema levou para atingir aquele determinado estado (PELLEGRINI; WAINER, 2007). Assim, são denominados processos de decisão pois modelam como um agente, tomador de decisões, pode interferir periodicamente em um sistema através de ações. O Processo Decisório de Markov, aqui será referenciado como MDP (do inglês - *Markov Decision Process*).

4.2.1 Definição de um MDP

Um MDP é definido como um tupla (S, A, T, R) onde:

- S é um conjunto finito de estados $s \in S$;
- A é um conjunto finito de ações $a \in A$;
- $T : S \times A \times S \rightarrow [0, 1]$ é a função de transição que indica a probabilidade do sistema passar para um estado $s' \in S$, dado que o processo estava em um estado s e o agente tomou a ação a , resultado em: $T(s'|s, a)$;

- $R : S \times A$ é a função que fornece o custo ou a recompensa por tomar uma decisão a quando o processo está no estado s .

Existem casos onde os conjuntos de ações disponíveis para cada estado são distintos, assim, definidos como:

- A_s conjunto finito de ações no estado $s \in S$

4.2.2 Épocas de Decisão

O intervalo entre a tomada de duas ações em um MDP é chamado de Época de Decisão. Neste intervalo o sistema permanece no estado, o qual foi conduzido pelo par estado-ação tomado anteriormente, por um determinado período a fim de validar os custos ou recompensas potencialmente obtidos.

4.2.3 Horizonte

O número de épocas de decisões disponíveis em um MDP é chamado de horizonte, definido como z . Este horizonte pode ser finito, infinito ou indefinido:

- finito: existe um número fixo de decisões a serem tomadas até que o processo encerre ou reinicie.
- infinito: as decisões são tomadas repetidamente sem possibilidade de parar.
- indefinida: as decisões são tomadas repetidamente até que o processo atinja um estado denominado como estado final (s_f).

4.2.4 Política

Pode-se dizer que o conjunto de estados e ações presentes em um MDP são definidos pelo ambiente no qual o agente se encontra. O ambiente propõe um determinado estado e as possibilidades de alterar este estado através de ações, assim, propondo novos estados. A cada época de decisão, a relação estado-ação cria uma regra de decisão denominada por $d : S \times A$. O conjunto de ações tomadas nos estados visitados (uma regra de decisão para cada época de decisão) é denominado de política. A cada execução do MDP, uma política é criada. Quando esta política já existe, o agente pode consultá-la para auxiliá-lo na tomada de decisões, destacando que, em ambientes determinísticos, o custo ou recompensa fornecido pelo próximo estado já será conhecido. O ciclo de interação entre o ambiente e as ações pode ser representado conforme figura 5.

Uma política pode ser caracterizada como:

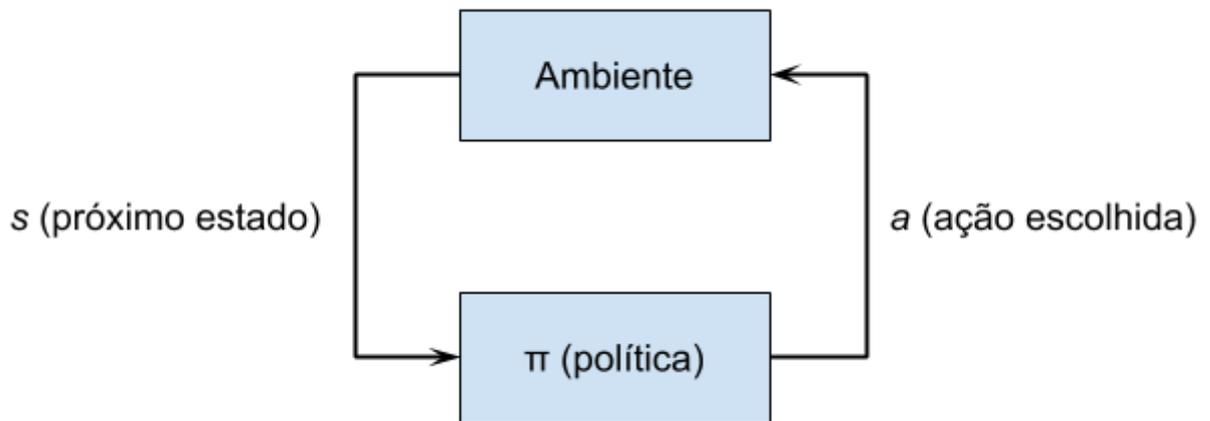


Figura 5 – Ciclo envolvendo o ambiente e a política em um MDP

- Total: quando cada estado do MDP possui uma regra de decisão definida;
- Parcial: quando parte dos estados de um MDP possuem regras de decisão definidas;
- Estacionária: quando a ação indicada pela política independe da época de decisão;
- Não-estacionária: quando a ação indicada depende da época de decisão no qual o MDP se encontra;
- Determinística: quando cada estado é relacionado a uma única ação;
- Não-determinística: quando cada estado é relacionado a um conjunto de ações que podem ser escolhidas de forma estocástica;
- Markoviana: quando a escolha da ação depende apenas do estado no qual o MDP se encontra;
- Não-Markoviana: quando a escolha de uma ação é dependente das decisões tomadas anteriormente no sistema até o ponto em que se encontra.

4.2.5 Critérios de desempenho

Apesar do objetivo de um MDP seja encontrar uma política ótima, ou seja, aquela que apresenta as maiores recompensas pela relação estado-ação, um sistema pode possuir inúmeras outras políticas. Para comparar o “grau de otimalidade” entre duas políticas, é preciso definir um critério de desempenho para o MDP. Dado uma época de decisão k , a recompensa obtida nesta época é denotada por r_k . Entre os critérios de desempenhos mais conhecidos estão:

- A recompensa média por época de decisão:

$$\frac{1}{z} \sum_{k=0}^{z-1} r_k \quad (4.1)$$

- A recompensa esperada total:

$$E \left[\sum_{k=0}^{z-1} r_k \right] \quad (4.2)$$

O fator de desconto é denotado por γ e aplicado para garantir a convergência da recompensa esperada total nos casos em que o MDP possui horizonte infinito.

- A recompensa esperada descontada:

$$E \left[\sum_{k=0}^{z-1} \gamma^k r_k \right] \quad (4.3)$$

Devido a natureza do problema estudado, este trabalho adotou o algoritmo *Q-Learning* como base de aprendizado da AR.

4.3 Algoritmo Q-Learning

O *Q-Learning* é uma técnica de Aprendizado de Máquina que adota uma função Q responsável por armazenar e fornecer o "valor" aprendido pelo agente. Este valor pode se alterar na medida com que o aprendizado avança e encontra estados melhores que os anteriores, e então, a função é atualizada até que se encontre os melhores valores de aprendizado possíveis.

O Algoritmo possui uma função que calcula a qualidade do par estado-ação:

$$Q : S \times A \rightarrow R \quad (4.4)$$

Antes do aprendizado iniciar, a função Q é inicializada com um valor arbitrário escolhido pelo usuário (geralmente inicia-se com zeros). Após a inicialização, a cada tempo t , o agente seleciona uma ação a_t , observa uma recompensa r_t , e então se move para o próximo estado s_{t+1} (a partir do estado anterior s_t e da ação tomada a_t). Ao final desses passos, a função Q é atualizada da seguinte forma:

$$Q(s_t, a_t) \leftarrow (1 - \alpha) \cdot Q(s_t, a_t) + \alpha \cdot [r_t + \gamma \cdot \max_a Q(s_{t+1}, a)] \quad (4.5)$$

O cerne do algoritmo é uma expressão de atualização por iteração de valor, usando a média ponderada do valor aprendido anteriormente com o valor atual, onde α é a taxa de aprendizado ($0 \leq \alpha \leq 1$) e γ é o fator de desconto ($0 \leq \gamma \leq 1$) que afeta a avaliação das recompensas recebidas no início do aprendizado ou no fim.

Existem dois cenários nos quais o aprendizado através do *Q-Learning* se encerra. O ambiente pode fornecer um estado final s_f , sendo que $Q(s_f, a)$ nunca é atualizado ou ainda em um cenário que contém infinitos laços de iteração. Para esse último caso, o fator

de desconto deve ser menor que um para garantir a convergência dos valores obtidos no par estado-ação.

4.3.1 Características Principais do Algoritmo

Este algoritmo se destaca por três características:

4.3.1.1 Exploração versus Investigação

A taxa de aprendizado α determina até que ponto as informações recém adquiridas substituem as informações antigas na função Q . Se a taxa é 0, o agente não aprenderá nada e o algoritmo se tornará exclusivamente exploratório. Por outro lado, se a taxa for definida como 1, isto faz o agente considerar apenas informações mais recentes, ignorando completamente o conhecimento prévio adquirido.

4.3.1.2 Fator de desconto

O fator de desconto γ determina a importância de recompensas futuras. Quando definido em zero, isto faz do agente míope considerando apenas recompensas atuais. Com o fator próximo de 1, o agente buscará recompensas melhores a longo prazo. Caso o fator de desconto for maior ou igual a 1, em um ambiente sem um estado final, as recompensas agregadas nunca encontrarão a convergência e, assim, tornará o aprendizado extremamente longo.

4.3.1.3 Condições iniciais da função Q

O Q -*Learning* é um algoritmo iterativo e implicitamente necessita de uma condição inicial antes do primeiro ciclo de atualização. Valores iniciais altos encorajam o agente a agir de forma exploratória, dado que geralmente as primeiras recompensas possuem valores baixos, aumentando então a probabilidade do agente escolher ações ainda não tomadas levando em conta o estado no qual se encontra.

5 Modelagem do Problema

A realização deste trabalho passa por criar um ambiente que possa representar as características da infraestrutura urbana através da criação de uma malha viária abstrata com a presença de carros e rotas. Além do mais, estes elementos devem respeitar as diretrizes do Processo Decisório de Markov para que seja possível utilizar a técnica de Aprendizagem de Máquina e, por fim, a avaliação dos resultados obtidos.

5.1 SUMO - *Simulator of Urban Mobility*

SUMO (*Simulation of Urban MObility*) é uma plataforma de simulação de tráfego de características microscópicas, inter e multimodal, onde a representação dos veículos adotam medidas de espaço-contínuo e tempo-discreto (BEHRISCH et al., 2011). Foi desenvolvida em sua maior parte pelo Instituto de Sistemas de Transportes (DLR) situado no Centro Aeroespacial Alemão. O início de seu desenvolvimento deu-se em 2001 e tão logo, em 2002, foi disponibilizada uma versão *opensource* à comunidade com a licença GPL (GNU *Public Licence*). A ferramenta foi desenvolvida com dois propósitos: ser aberta a comunidade para que fosse acessível a estudantes e pesquisadores, e também ser um simulador que recebesse suporte de outras instituições.

Ao longo dos anos, o SUMO se transformou em uma suíte de aplicações completa para modelagem de sistemas viários, incluindo a capacidade de se comunicar com outras aplicações como Vissim, OpenDRIVE e OpenStreetMaps, entre outros, e ainda uma interface programável, nomeada de TraCI (*Traffic Communication Interface*). A tabela 1 apresenta um breve descritivo das aplicações contidas na suíte deste simulador.

A característica microscópica deste simulador permite com que os veículos sejam modelados individualmente dado uma malha viária. Cada veículo possui sua própria rota e movem-se individualmente pelas vias determinadas. Apesar de utilizar elementos determinísticos, é possível empregar certa aleatoriedade na simulação.

Todas essas características apresentadas dividem o SUMO em quatro grandes módulos.

5.1.1 Malha Viária

A simulação de tráfego necessita que a estrutura de ruas e vias sejam criadas ou importadas de outros sistemas. Estas estruturas têm como propósito representarem as redes de ruas, avenidas e estradas do mundo real na forma de um grafo, onde os nós são as interseções das vias, ou seja, os cruzamentos, e as arestas são as vias em si.

Tabela 1 – Lista de aplicações da suíte SUMO.

Aplicação	Descrição
sumo	Aplicação principal do simulador sem interface gráfica, apresentando apenas informações por linha de comando no <i>prompt</i> de comando.
sumo-gui	Aplicação principal do simulador com interface gráfica.
netconvert	Capaz de gerar e importar malhas viárias de outras aplicações com diferentes formatos e convertê-los para o formato SUMO.
netedit	Editor gráfico de malha viária.
netgenerate	Gera malhas viárias abstratas a partir de parâmetros pré-definidos para o formato SUMO.
duarouter	Calcula rotas mais rápidas através da malha.
jtrouter	Calcula as rotas baseadas nos cruzamentos.
dfrouter	Calcula rotas baseadas em medidas feitas por laços de indução.
od2trips	Decompõe matrizes de origem e destino (O/D) em jornadas individuais para cada veículos.
polyconvert	Importa pontos de interesse e polígonos em diferentes formatos, traduzindo-os em elementos que podem ser utilizados e visualizados pelo SUMO-GUI.
activitygen	Módulo gerador de demanda de tráfego a partir de zonas populacionais.
emissionmap	Gera mapas de emissão de gases.
emissiondrivingcycle	Calcula valores de missões baseados em uma determinada rota.

As arestas são conexões unidirecionais entre dois cruzamentos onde, cada aresta representa uma faixa da via. Assim, quando deseja-se criar uma via com mais de uma faixa, cada faixa será representada como uma aresta entre os referentes nós. Além do mais, é possível definir características geométricas das vias como tamanho e posição, e em cada faixa existe a capacidade de permitir o acesso de determinados tipos de veículos e a velocidade máxima permitida.

A malha viária do SUMO pode ser gerada de forma procedural através da aplicação *netgenerate*, ou ainda ser importada de um mapa digital usando a aplicação *netconvert*. É possível gerar três tipos de malhas usando o *netgenerate* com os seguintes formatos (Figura 6):

- grade (*manhattan-like grid*);
- teia (*spider-net*);
- aleatória (*random*).

Cada tipo de estrutura possui uma série de propriedades que podem ser ajustáveis

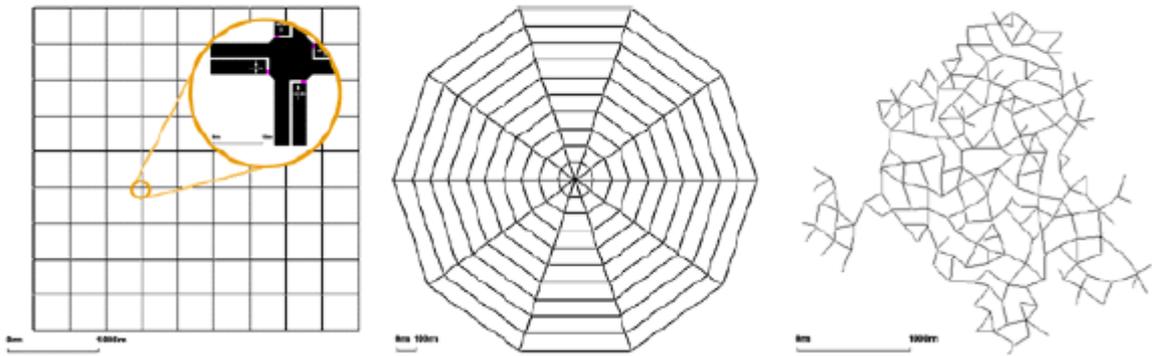


Figura 6 – Malhas abstratas construídas a partir da aplicação *netgenerate*; da esquerda para direita: grade, teia e aleatória.

Fonte: (KRAJZEWICZ et al., 2012)

no momento da geração. Com a aplicação *netconvert*, é possível utilizar malhas de outros simuladores de tráfego como o VISUM, Vissim e o MATSim, assim como interpretar mapas com formato digital do *OpenStreetMaps*.

Tanto a aplicação *netconvert* como a aplicação *netgenerate* adotam um processo semelhante de geração de malha viária. Este processo passa por etapas de pré-processamento com o intuito de otimizar e eliminar redundâncias dentro da estrutura pretendida (Figura 7).

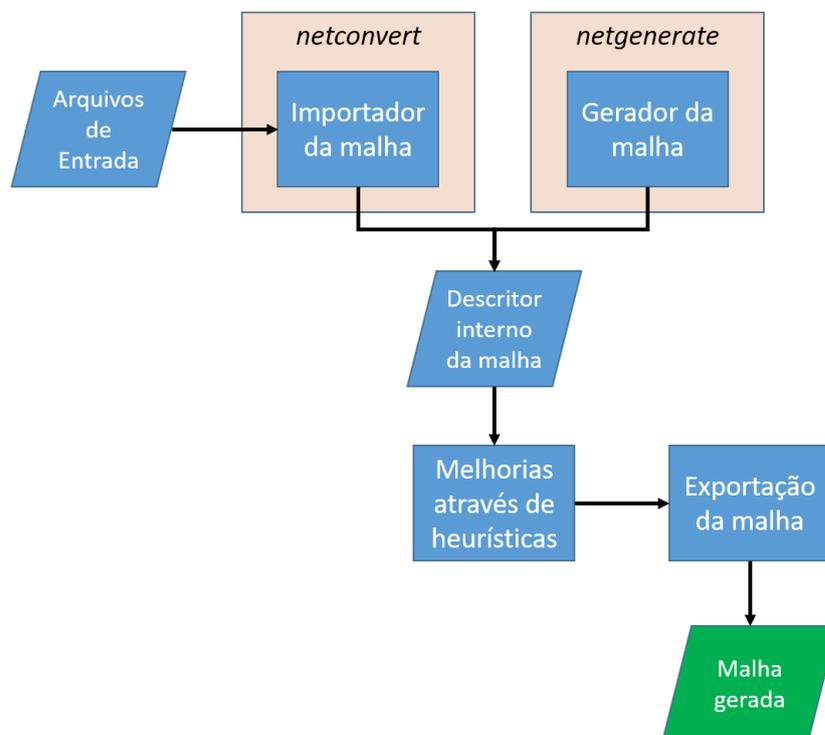


Figura 7 – Preparação da malha viária a partir das aplicações *netgenerate* e *netconvert*.

Fonte: Adaptado de (KRAJZEWICZ et al., 2012)

O formato nativo da estrutura viária do SUMO é representado por arquivos do

tipo XML simples (*plain*) e descrevem todos os nós, as arestas, tipos de arestas (quando aplicáveis), os cruzamentos e o sistema de semáforo quando utilizado. A malha gerada pode ser editada com uma outra aplicação da suíte chamada *netedit*, a qual permite a edição das vias através de uma interface gráfica, facilitando e agilizando a criação de estruturas mais simples para pequenos projetos.

5.1.2 Veículos e Rotas

A característica microscópica do SUMO exige que cada veículo seja definido de forma explícita, ou seja, possua pelo menos um atributo identificador do ponto de vista da simulação. Além deste atributo obrigatório, os veículos também possuem, entre outras características, o tempo de partida (em passos de simulação) e uma rota, sendo esta uma lista completa de vias (arestas) interligadas entre a origem e o destino deste veículo.

Os veículos podem ser definidos por um tipo, o que descreve suas características físicas como: tamanho, velocidade máxima atingida, aparência, entre outros atributos.

Este simulador foi projetado para atender demandas de diferentes dimensões, desde a análise de poucos veículos, até a utilização de dezenas de milhares ou milhões deles. Para estes casos, é possível utilizar matrizes de origem e destino, ou ainda, a definição de zonas de tráfego chamadas TAZ (*Traffic Assignment Zones*). Estas zonas devem ser utilizadas em pares, sendo que uma assume o ponto de origem e outra o destino de uma rota.

As rotas podem ser geradas através de aplicações auxiliares ou definidas explicitamente quando necessário. Cada rota possui um atributo identificador e as arestas que a compõem entre a origem e o destino. Opcionalmente é possível atribuir uma cor a esta rota, caso deseje-se identificar graficamente os veículos que a utilizarem. Há também a possibilidade de definir as rotas utilizando apenas informações de origem e destino, seja utilizando arestas ou zonas de tráfego. Neste caso, o simulador adota algoritmos de busca de caminho mínimo para identificar implicitamente as rotas completas baseando-se nas condições atuais da simulação.

Adicionalmente as rotas, existe a alternativa de definir fluxos de tráfego. Os fluxos são definidos a partir de informações de tempo de início e de fim, em passos de simulação e também informações relativas ao volume de tráfego. Este volume pode assumir características determinísticas ou estocásticas.

5.1.3 Simulação

Toda simulação no SUMO possui as características de ser discreta no tempo e contínua no espaço. O passo de simulação padrão é de 1s no tempo real, ou seja, todos os cálculos referentes a movimentação dos carros na malha viária utilizam o tempo de um

segundo a cada passo da simulação. A definição de posição de um veículo se dá pela faixa da via que ele está ocupando e a distância do início desta via.

Quando os veículos se deslocam pelas vias, as respectivas velocidades são calculadas baseadas no Modelo de Dinâmica de Veículos (*car-following model*). O modelo utilizado por padrão neste simulador foi definido por Stefan Krauss devido a sua simplicidade e desempenho para a realização dos cálculos. Além deste modelo, a simulação de um sistema de tráfego exige também a implementação de um Modelo de Cruzamento e um Modelo de Mudança de Faixa. Estes três modelos são o coração do modelo de simulação microscópico.

5.1.3.1 Modelo de Dinâmica de Tráfego

Os veículos em uma via tendem a tomar decisões baseados nos outros carros que dividem o espaço próximo. Quando se trata de um fluxo de veículos, o posicionamento e a velocidade adotada por cada um está relacionada com o veículo adiante. Em geral, um fluxo é orientado por um veículo que, por inúmeras condições, assume a dianteira de uma fila de carros e todos os carros anteriores adotam uma velocidade de deslocamento média. Além desta velocidade, outros fatores como aceleração positiva e negativa, e a distância para o próximo veículo, influenciam em todo o sistema de tráfego. Assim, este tipo de modelo procura apresentar uma forma de descrever o relacionamento entre os veículos em um determinado fluxo.

O modelo proposto por Krauß (1998) parte do pressuposto que, em geral, os veículos se movem sem colidir. Desta forma, a dinâmica de comportamento dos veículos fica condicionada a capacidade de aceleração e frenagem.

Apenas para critério de conhecimento, o funcionamento adequado deste modelo depende de três características:

- Tamanho do veículo (*length*);
- Distância entre veículos (*minGap*); e
- Imperfeição quanto a aceleração do motorista (σ).

Estas características influenciam na formação de filas de veículos e consequentemente em congestionamentos (Figura 8). Quando os veículos são criados para serem inseridos na malha, estes valores podem ser configurados para diferentes situações, no entanto, para este trabalho em específico, os valores foram mantidos conforme estabelecido de forma padrão pelo simulador.

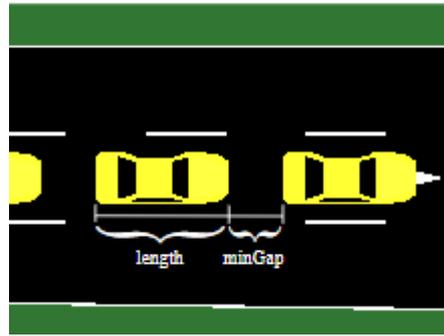


Figura 8 – Relação entre tamanho do veículo e distância para outro veículo.

Fonte: <http://sumo.sourceforge.net/userdoc/Definition_of_Vehicles,_Vehicle_Types,_and_Routes.html>

5.1.3.2 Modelo de Cruzamento

O comportamento dos veículos quando se aproximam e atravessam um cruzamento em uma malha viária é de extrema importância para sistemas microscópicos, pois os veículos necessitam evitar colisão com qualquer outro que também está cruzando o mesmo caminho. Existem diferentes esquemas para tratar essa situação como: regra de prioridade ao veículo da direita, uso de semáforos, entre outras situações.

5.1.3.3 Modelo de Mudança de Faixa

É comum encontrar em cenários reais vias com múltiplas faixas. O modelo de mudança de faixa tem como princípio propor o comportamento de um veículo em cenários assim. No simulador SUMO, há quatro diretivas que influenciam a mudança de faixa de um veículo:

- Estratégia: usar outra faixa para continuar a rota até o destino;
- Cooperativo: mudar de faixa para deixá-la livre para outros veículos prioritários;
- Ganho de velocidade: utilizar faixas mais rápidas, menos ocupadas;
- Se manter na direita: se manter na direita para os veículos mais rápidos usarem a esquerda.

Além de gerenciar o motivo pelo qual o veículo deverá mudar de faixa, este modelo também é responsável por adaptar a distância segura entre a frota e adaptar a velocidade dos carros.

5.1.4 TraCI - Interface de Controle de Tráfego

É possível interagir em tempo real com a simulação através de uma interface chamada TraCI (*Traffic Control Interface*). Esta interface se conecta com a simulação via soquete, no formato de cliente-servidor. Através do TraCI é possível acessar diferentes recursos e atributos pertencentes a simulação como: cruzamentos, vias, faixas das vias, sistemas de semáforo, aspectos da própria simulação, além da possibilidade de alterar os atributos de determinados objetos, como por exemplo, forçar um carro a mudar de faixa ou alterar a velocidade máxima de uma via. É possível também iniciar mais de uma simulação ao mesmo tempo e controlá-las através da mesma conexão. A API TraCI é distribuída oficialmente com a suíte de aplicativos do SUMO e foi escrita na linguagem Python, no entanto, é possível acessar a API com qualquer outra linguagem de programação que permite conexões TCP via soquete.

Existem dois modos para executar as simulações no simulador, o modo na forma de linha de comando (*console*) e no formato visual, utilizando uma interface gráfica que permite renderizar a malha viária e exibir os veículos em movimento utilizando *OpenGL*. Ao final da simulação, é possível gerar diferentes relatórios que descrevem, por exemplo, a jornada de cada veículo, a posição em cada passo da simulação, inclusive dados relacionados a emissão de CO_2 e consumo de combustível. Todos os relatórios são gerados em formato *XML*.

5.2 Criação da Malha Viária

O escopo deste estudo visa propor um sistema escalável, que possa ser utilizado para testar diferentes tamanhos de cenários. Para isso, todas as instâncias dos objetos presentes na estrutura da malha como: cruzamentos, ruas, faixas das ruas e carros, foram criados respeitando um padrão de nomenclatura, baseada em um plano cartesiano. Esta padronização, além de permitir o uso de malhas com tamanhos diferentes, facilita também o acesso as informações dos objetos, dado a possibilidade de acessar as instâncias através da API.

Um conjunto interligado de vias pode ser representado através de um grafo direcionado, onde os nós representam os cruzamentos e as arestas, as ruas que conectam um cruzamento ao outro.

A criação da malha viária utilizada foi auxiliada pela aplicação *netgenerate*. Esta aplicação permite que diferentes malhas sejam criadas a partir da definição de alguns parâmetros. Por exemplo, assumindo que um modelo de malha em grade possa ser representado por uma matriz quadrada, a ordem dessa matriz é o parâmetro indicador do tamanho da malha que será criada. Dessa forma, três nós geram uma malha com nove cruzamentos, quatro nós com dezesseis, cinco nós com vinte cinco e assim por diante.

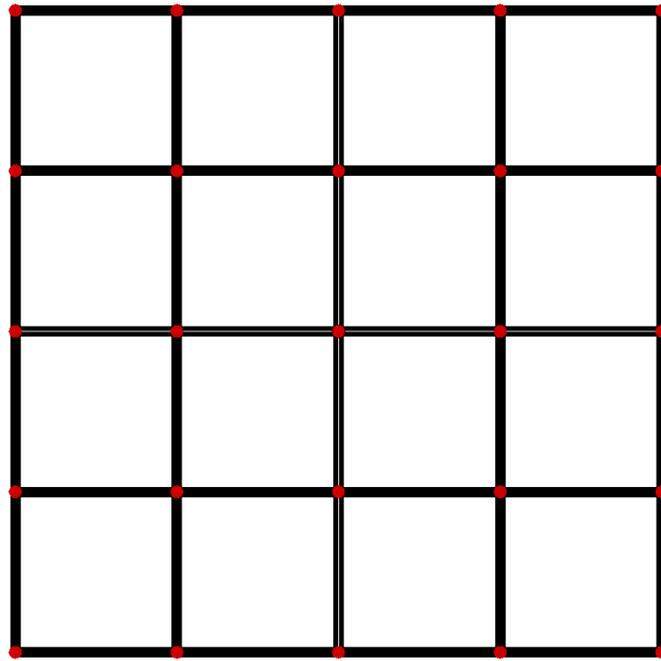


Figura 9 – Representação Gráfica da Malha Viária no SUMO

A nomenclatura dos nós é composta por dois termos separados por barra “x/y”. O nó inicial é nomeado como “0/0”, sendo este posicionado no canto inferior esquerdo a partir da representação gráfica da malha. Na medida em que os nós progredem para a direita, o termo “x” é acrescido em um. Da mesma forma que ao progredir para cima, o termo y é acrescido da mesma forma. Desta maneira, o último nó da malha, posicionado no canto superior direito, tanto o termo “x” como o termo “y” é igual a ordem da matriz que gerou a malha - 1. Por exemplo, em uma malha com vinte e cinco cruzamentos, o último nó será o “4/4”.

A partir da malha criada (Figura 9), a aplicação *netedit* foi utilizada para configurar as velocidades permitidas das vias com o propósito de aproximar o modelo a um cenário real. Então as vias passaram a adotar velocidades permitidas diferentes seguindo o seguinte modelo baseado no espectro de cores visíveis conforme figura 10.

Cada cor utilizada representa uma zona, sendo que as cores mais próximas de azul são as zonas de velocidade permitida mais alta, e as zonas com cores próximas ao vermelho são de velocidade permitida mais baixa. Cada zona pode assumir até três velocidades permitidas diferentes de acordo com a Tabela 2.

Tabela 2 – Tabela de Velocidades por Zona

Zona	Cor	Padrão RGB	Velocidade (Km/h)
01	Azul Ciano	(0, 255, 255)	[60, 70, 80]
02	Verde	(0, 255, 0)	[40, 50, 60]
03	Laranja	(255, 165, 0)	[0, 20, 30]

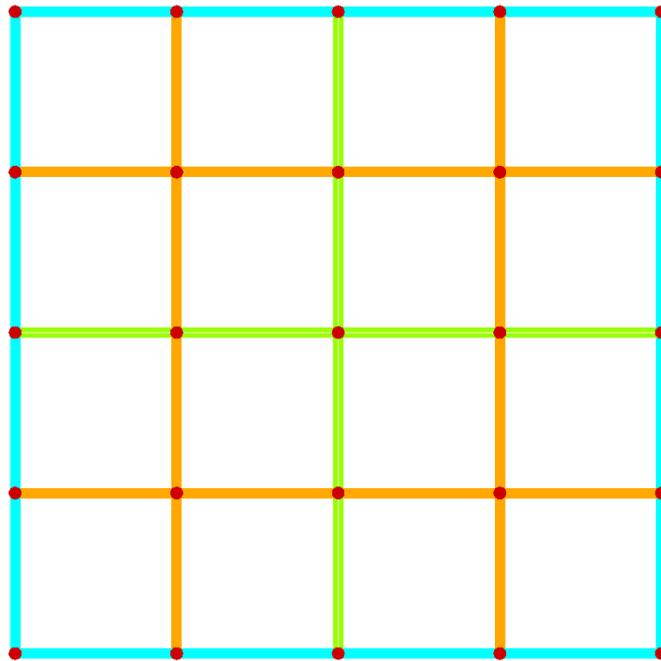


Figura 10 – Representação Gráfica das Zonas de Velocidade.

No que se refere aos elementos da malha, este trabalho os discrimina da conforme os dados apresentados na Tabela 3.

Tabela 3 – Elementos formadores da malha viária.

Elemento	Descrição
Via	É formada por todos os nós e respectivas arestas conectados em uma mesma direção. Podem estar dispostas na vertical ou horizontal.
Faixa da Via	Formada por todos os nós e respectivas arestas conectados em um mesmo sentido. Assim como as vias, podem estar dispostas na vertical ou horizontal mas seguem um único sentido (Norte, Sul, Leste ou Oeste).
Segmento de Via	São duas arestas conectoras de dois nós pertencentes a uma mesma via.

5.2.1 Definição dos Estados

O controle dinâmico de tráfego é feito através da variação das velocidades permitidas em cada via ou em cada faixa da via. Para isso, cada estado s do MDP foi definido como um vetor que contém a velocidade máxima permitida de cada aresta. Considerando n a quantidade de nós existentes na malha, o tamanho deste vetor é da seguinte ordem:

$$4 * (n - \sqrt{n}) \quad (5.1)$$

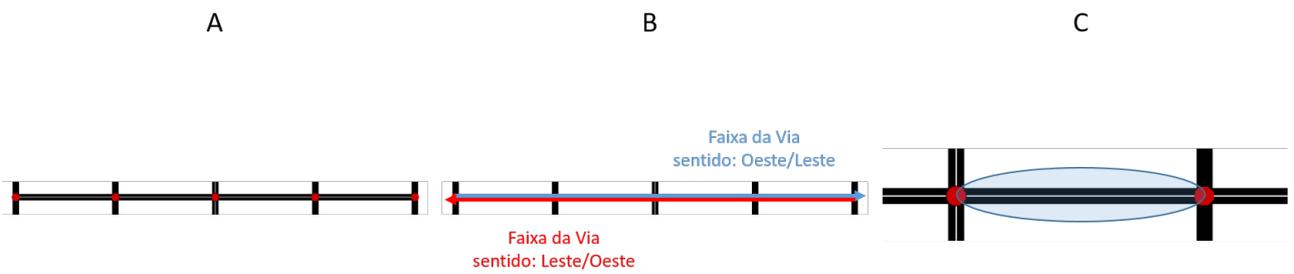


Figura 11 – Definição dos elementos formadores da malha viária. A) Via; B) Faixa da Via; C) Segmento de Via.

Desta forma, o conjunto de estados S é um vetor bidimensional e dinâmico onde cada posição armazena um outro vetor s na medida em que eles são visitados no MDP.

5.2.2 Definição das Ações

Cada ação a é definida como um vetor de tamanho fixo baseado na quantidade de arestas presentes no ambiente seguindo o mesmo padrão apresentado pelo vetor de estados (Equação 5.1).

Cada posição deste vetor contém um valor $i \in [-1, 1]$ descritos com mais clareza na seção Regras de Decisão.

Após executar uma ação, o valor da velocidade permitida em uma faixa da via deve obrigatoriamente respeitar o intervalo de valores definidos na Tabela de Velocidades.

As ações são agrupadas por Faixas das Vias, ou seja, um grupo de arestas é atualizado ao mesmo tempo após a execução de uma ação. Sendo assim, o conjunto de ações A é definido como um vetor bidimensional, de tamanho fixo, que armazena uma ação a em cada posição. Diferentemente do vetor S que se expande de acordo com a execução do MDP, o vetor A é processado e preenchido antes do início do aprendizado.

5.2.3 Regras de Decisão

As Regras de Decisão definidas neste sistema consiste em afetar a velocidade das vias de três modos (Tabela 4):

- Aumentando a velocidade;
- Diminuindo a velocidade;
- Mantendo a velocidade atual.

Caso a via já esteja na velocidade máxima, mesmo com a regra de aumentar velocidade, o valor permanece inalterado. O oposto ocorre com a velocidade mínima. Se o

valor da velocidade permitida se encontre no mínimo definido, a velocidade será mantida mesmo com a ação de diminuição.

Cada regra possui um número inteiro associado para que em seguida a Aplicação de Controle de Vias verifique de que forma a via será modificada.

Tabela 4 – Valor de cada ação

Regra	Valor
Diminuir	-1
Manter	0
Aumentar	+1

5.2.4 Épocas de Decisão

A partir de testes e análises realizadas, foi definido 300 passos de simulação como o tamanho adequado para uma Época de Decisão no ambiente deste trabalho. Em tempo real, o número de passos de simulação estipulados representa 5 minutos.

Esse tempo foi pensado para que o sistema tivesse tempo para se estabilizar com as alterações na malha viária e refletir os efeitos causados pela mudança no ambiente.

5.2.5 Horizonte

Devido a dimensão do ambiente, no que diz respeito ao número de estados, a opção de utilização de um horizonte finito se fez necessária na intenção de limitar o tempo de execução do MDP.

O tamanho do horizonte nesta aplicação foi definido como $100 \times$ o tamanho da época de decisão. Neste caso, 30000 passos de simulação. Portanto, a cada horizonte z , 100 ações são tomadas.

5.2.6 Episódios

O ciclo de vida do aprendizado da simulação adotada neste trabalho se encerra ao final de 100 episódios. Cada episódio consiste na execução do total de horizontes estipulados. Considerando uma iteração a cada época de decisão, todo o sistema realiza 10000 iterações ao final de todo o processo.

5.2.7 Recompensa

Um meio encontrado para medir a eficiência da técnica aplicada foi simplesmente avaliar a quantidade de carros que chegaram ao destino dentro de uma mesma Época de Decisão. Este número foi utilizado como recompensa para o par estado-ação que conduziu o sistema ao estado em que o MDP estará.

5.3 Aplicação para controle dinâmico de tráfego

5.3.1 Estrutura Lógica da Aplicação

Segue abaixo uma descrição a respeito da organização lógica da Aplicação para Controle de Velocidades nas Vias representado pela Figura 12. A aplicação foi construída de forma modular e parametrizada a fim de facilitar alterações e adição de novas funcionalidades. A linguagem de programação utilizada para desenvolvimento desta aplicação foi a *Python*, devido a API disponível do próprio simulador.

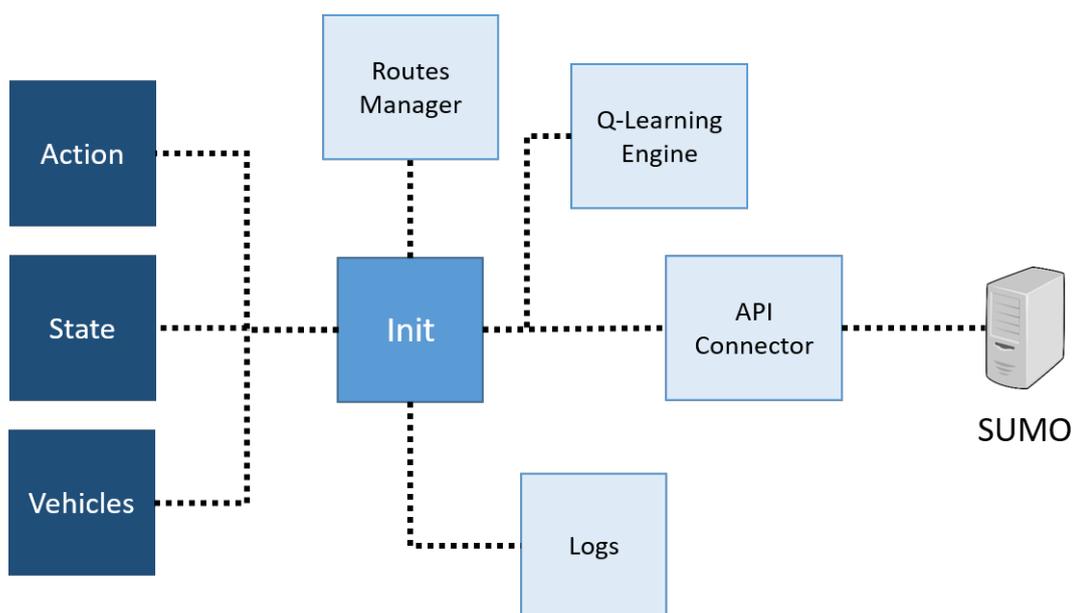


Figura 12 – Estrutura lógica da Aplicação desenvolvida

5.3.2 Init

Init é o bloco central da aplicação responsável por interoperar todos os outros blocos adjacentes. É onde a Aplicação se inicia e gerencia todo o ciclo de funcionamento relacionado ao Aprendizado por Reforço.

Primeiramente, todos os parâmetros da aplicação são inicializados a partir deste módulo antes de iniciar a simulação (Tabela 5).

Os parâmetros devem ser configurados diretamente no código da aplicação. Após esta etapa, o sistema pode ser inicializado através de uma IDE ou mesmo de um compilador *Python*.

Antes de iniciar a comunicação com o simulador, a aplicação faz uma leitura no arquivo referente a malha viária com o propósito de obter a seguinte lista de objetos: cruzamentos (nós) e arestas. Em seguida, o usuário pode escolher a execução do simulador em modo gráfico ou modo console, e então, a aplicação estabelece uma conexão com o simulador através da API TraCI.

Tabela 5 – Parâmetros iniciais da aplicação.

Parâmetro	Descrição	Valor
RL_ON	Define se malha viária será atualizada conforme a execução da Aprendizagem por Reforço.	<i>true/false</i>
N_NODES	Raiz quadrada do número de nós (cruzamentos) existentes na malha.	$Z \geq 2$
HORIZON	Tamanho do horizonte z .	$Z > 0$
EPISODES	Número de episódios que o sistema será executado.	$Z > 0$
HORIZON_SIZE	Tamanho da Época de Decisão dado em passos de simulação (steps).	$Z > 0$
EXPLORATION_RATE	Porcentagem do número de episódios que será utilizada para o balanceamento da Exploração x Investigação do MDP com a atualização dos valores da política ϵ -greedy.	$R > 0$
LANES_SHOULD_NOT_CHANGE	Nome das faixas que não pode ser atualizadas pelo Aprendizado de Máquina para não impossibilitar a execução da simulação.	Vetor de <i>strings</i>
ROUTE_MODE	Modo no qual será definido o roteamento de carros na malha.	<i>string</i>
NUMBER_OF_CARS_PER_ITERATION	Número de carros a serem inseridos na malha por Época de Decisão.	$Z > 0$
NUMBER_OF_ROUTES	Caso o parâmetro ROUTE_MODE seja “Route”, NUMBER_OF_ROUTES define quantas rotas diferentes serão adotadas na simulação.	$Z > 0$
FORWARD_ONLY	Caso o parâmetro ROUTE_MODE seja “Route”, FORWARD_ONLY define se o sentido das rotas adotadas para os carros será apenas em caráter de caminho mínimo até o destino, considerando apenas o tamanho das vias e não a velocidade permitida.	<i>true/false</i>
ACTIONS_BY_LANE	Define se o conjunto A de ações será composto por ações a que afetar as vias no dois sentidos ou cada faixa das vias individualmente.	<i>true/false</i>

Ao iniciar a simulação, este módulo também é responsável por gerenciar o ciclo de vida da simulação, incluindo as etapas pertinentes ao Aprendizado por Reforço. Estas etapas adotam a seguinte ordem:

1. Configuração do estado inicial s_0 e atualização do conjunto S ;
2. Escolha de uma ação a do conjunto A de acordo com a política ϵ -greedy;
3. Atualização da tabela-Q do algoritmo *Q-Learning* com o estado s e ação a obtidas no passo 1 e 2;

4. Definição do próximo estado s' , a partir da atualização da malha, baseada no estado anterior s e a ação a . E então, o estado s' é adicionado ao conjunto S ;
5. Atualização da função de transição $T(s, a, s')$.
6. Atualização da malha viária de acordo com o estado resultante do passo 4;
7. Verificação de rotas bloqueadas após a atualização da malha e, caso existir, substituição por outra escolhida de forma aleatória;
8. Execução de uma Época de Decisão da simulação pelo número de passos de acordo com o parâmetro $HORIZON_SIZE$.
9. Caso existam carros travados após duas Épocas de Decisão, estes carros são roteados novamente utilizando o algoritmo de caminho mínimo de *Dijkstra*, disponível pelo simulador. Os carros roteados nesta etapa são descontados da recompensa gerada na Época de Decisão em que o sistema se encontra. Se o travamento dos carros persistirem por quatro Épocas, estes carros são eliminados do sistema e também não são contados no cálculo de recompensa.
10. Neste passo a recompensa é calculada a partir da quantidade de carros obtidos no ponto de destino durante a Época de Decisão.
11. Então, o algoritmo atualiza a tabela-Q a partir da transição s, a, s' e a recompensa obtida.
12. O estado s' agora passa a ser o estado s e a simulação retorna ao passo 2 até que o número de passos seja igual a multiplicação dos parâmetros $HORIZON * HORIZON_SIZE$.

Todo o processo descrito acima corresponde a execução de um episódio. O número de episódios é definido pelo parâmetro EPISODES. Assim, ao final de um episódio, o parâmetro do algoritmo *Q-Learning* referente a política ϵ -greedy é atualizado para garantir um equilíbrio entre o tempo de Investigação e Exploração dos estados. Em seguida, valores agregados dos carros que chegaram ao destino e a velocidade média de todas as vias durante o episódio são armazenados para geração de relatórios.

5.3.3 State

O módulo *State* tem como responsabilidade gerenciar as estruturas de dados pertinentes a cada estado s e ao conjunto S de estados.

Devido a característica do ambiente parcialmente observável deste sistema, todos os estados são desconhecidos até que o MDP os visite. O vetor S , o qual armazena o

conjunto de estados s , é populado na medida com que o sistema executa novas ações e visita novos estados. Após a execução de uma ação, o sistema verifica se o estado encontrado já foi visitado alguma vez, e, em caso afirmativo, o estado s' assume o valor do estado já visitado, em caso negativo, uma nova posição é adicionada ao vetor na qual será armazenada o novo estado em questão.

5.3.4 Action

O módulo *Action* tem como finalidade estabelecer as estruturas de dados que representam as ações a e o conjunto de ações A dentro do sistema, assim como, todas as funcionalidades relacionadas ao manuseio e gerenciamento destas estruturas.

Diferentemente dos estados, que são descobertos na medida que são visitados, as ações que atuam no ambiente são definidas na inicialização do sistema a partir da malha utilizada.

Há dois modos para geração de ações:

- Ações por Vias;
- Ações por Faixa da Via.

Ambos os modos de geração de ações partem de uma leitura de todas as arestas existentes na malha. No entanto, no caso de Ações por Vias, as duas arestas que pertencem ao mesmo Segmento de Via são considerados na mesma ação (Figura 13). Por outro lado, as Ações por Faixa de via consideram as arestas de um Segmento de forma independente.

5.3.5 Routes Manager

A fim de propor um funcionamento orgânico ao comportamento dos carros dentro da malha, este módulo foi criado para estabelecer rotas entre os pontos de origem e destino e configurá-las para serem utilizadas dentro da simulação.

Assume-se que dentro de uma malha os veículos nem sempre escolhem rotas diretas ou otimizadas até o destino. Prestadores de serviços, transporte público, entre outros ocupantes das vias urbanas, muitas vezes adotam rotas contrárias ao destino pois consideram outros fatores na escolha do caminho. A partir deste ponto de vista, o trânsito de uma cidade é caótico e a imprevisibilidade no comportamento dos veículos gera padrões de rotas amplamente variados.

Utilizando uma adaptação do algoritmo de Busca em Largura, é possível obter a lista de adjacências do grafo gerado a partir da malha. Este algoritmo foi escolhido por seu caráter completo, ou seja, visita todos os nós de um grafo, e também a facilidade de implementação. No entanto, do ponto de vista desta aplicação, há necessidade apenas

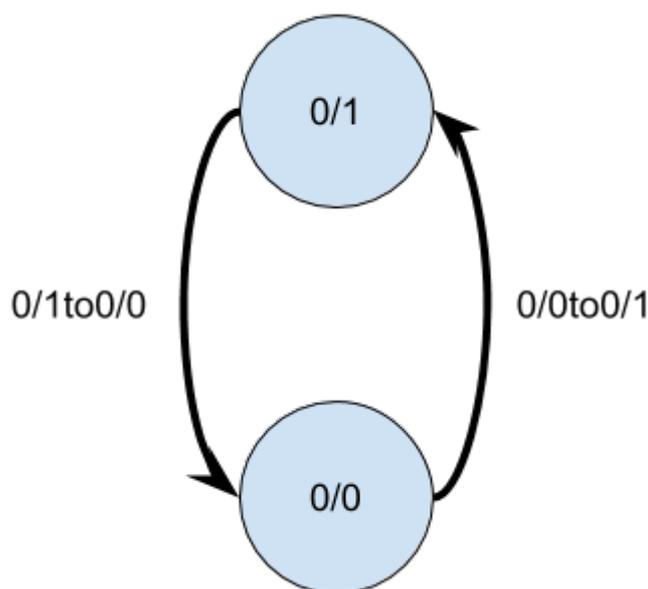


Figura 13 – Dois nós conectados por duas arestas direcionadas representando um Segmento de Via, ou seja, dois cruzamentos conectados por duas vias com sentidos opostos.

de conhecer as possíveis arestas de conexão entre dois nós distintos. Assim, a adaptação aplicada a este algoritmo foi a remoção das estruturas de dados dedicadas a obtenção do caminho mínimo ficando apenas aquelas pertinentes a geração de uma lista de adjacência.

A partir da lista de adjacências, é possível saber todas as arestas existentes entre dois pontos quaisquer, neste caso específico, os pontos de origem e destino da malha. Conectando estas arestas, é possível obter todas as rotas para que sejam adicionadas ao simulador através da API Traci para em seguida serem utilizadas pelo carros inseridos na malha. Caso o usuário tenha definido o parâmetro `FORWARD_ONLY` com o valor *true*, apenas as arestas com sentido ao nó de destino são consideradas, ou seja, aquelas que navegam dos nós da esquerda para a direita e no sentido de baixo para cima. Por outro lado, caso o parâmetro estiver com o valor configurado para *false*, todas as rotas são consideradas.

5.3.6 Vehicles Manager

Uma das características fundamentais dos simuladores de trânsito microscópico é a descrição de cada carro existente na malha viária, ou seja, cada veículo possui uma instância dentro da simulação. Desta forma, para que este elemento exista, se faz obrigatória a configuração de alguns parâmetros como: identificador, tipo de veículo (tipos definidos internamente pelo SUMO) e rota.

O gerenciador de veículos desta aplicação tem a responsabilidade de inseri-los na

malha. A cada Época de Decisão, um determinado volume de veículos são inseridos a partir da definição de um identificador (representado por um número inteiro maior que zero e crescente), o tipo de veículo a ser utilizado (definido pelo próprio simulador) e o identificador de uma rota, obtido a partir do módulo descrito anteriormente. Com estas informações, o veículo passa a existir na simulação e percorrerá a rota destinada a ele.

5.3.7 Q-Learning Engine

Este módulo é responsável por realizar o funcionamento do Aprendizado por Reforço a partir do provimento das estruturas de dados necessárias e também os processos pertinentes a própria inteligência da técnica.

Todo o funcionamento do algoritmo *Q-Learning* parte dos seguintes parâmetros: taxa de aprendizado (α) e o fator de desconto (γ). Neste caso, há também o parâmetro balizador da política gulosa (ϵ). A partir destes parâmetros, todo o processo de aprendizado é realizado.

O cerne do funcionamento da técnica de aprendizado utilizada nesta dissertação é a estrutura que define a tabela-Q. Esta tabela armazena um valor associado a cada par estado-ação do ambiente. Uma das maneiras de definir esta tabela é através da utilização de matrizes onde os índices referente às linhas representam os estados e os índices das colunas representam as ações. No entanto, em MDPs com um grande número de estados e ações, a taxa de esparsidade da matriz pode ser muito alta, elevando o custo computacional do aprendizado, dado que, além desta tabela ser acessada a cada Época de Decisão, ela deve ser inicializada antes de começar a simulação.

Para evitar uma matriz esparsa, a tabela-Q é definida como um vetor multidimensional dinâmico cujas posições se comportam como listas ligadas. Os índices do vetor representam os estados visitados no MDP e o conteúdo de cada posição contém um outro vetor que armazena um par ação-valor de aprendizado (Veja figura 14).

Com a tabela-Q definida, o processo de aprendizado passa por atualizar a tabela e identificar a ação relacionada ao próximo estado com o maior valor de aprendizado no momento em que o ambiente se encontra.

5.3.8 API Connector

Para acessar as funcionalidades do simulador é necessário utilizar a API Traci. A conexão entre a aplicação de controle de vias e o sistema de simulação é realizada conforme uma conexão cliente/servidor.

O módulo API *Connector* é responsável por iniciar e gerenciar a conexão com o servidor SUMO a fim de garantir o funcionamento de toda a aplicação.

5.4 Configuração do Ambiente

Para recriar o ambiente utilizado neste trabalho, a instalação de três ferramentas se faz necessária:

- Simulador SUMO;
- Linguagem de Programação Python;
- IDE com suporte a linguagem Python.

5.4.1 Sumo

Sumo é uma aplicação *opensource*, multiplataforma, distribuída sob a Licença Pública do Eclipse (EPL-2.0). Os pacotes deste simulador são distribuídos para as plataformas *Linux*, *MacOS* e *Windows*, além de suportar o uso no *Docker*. Toda a documentação de instalação, configuração e uso da ferramenta está disponível em sua página principal <<http://sumo.dlr.de/wiki/SUMO>>.

A distribuição deste simulador para a plataforma Windows é feita através de quatro pacotes binários nas versões 32 e 64 bits, sendo que uma versão é para instalação local (quando o usuário possui permissão de administrador da máquina) e outra se dá em formato compactado e portátil, sem a necessidade de acesso de administrador. Os pacotes estão disponíveis no seguinte endereço: <<http://sumo.dlr.de/wiki/Installing>>. A instalação pelo pacote binário ocorre através de um assistente de instalação padrão. O pacote compactado deve ser descompactado em um local escolhido pelo usuário.

O SUMO possui uma interface de comunicação em tempo real com a simulação, e para utilizá-la, é necessário fazer uso da linguagem de programação Python.

5.4.2 Python

Python é uma linguagem de alto-nível e propósito geral. Se encontra disponível no seguinte endereço: <<https://www.python.org/>>. O processo de instalação é também através de pacotes binários e, para instalá-la, basta escolher o pacote adequado na seção “Downloads” do *site* e seguir os passos do assistente de instalação. O uso da linguagem pode ser feita através de um Bloco de Notas padrão disponível no *Windows*, entretanto, para auxiliar na programação e aumentar a eficiência na codificação, é recomendado o uso de uma IDE (Ambiente Integrado de Desenvolvimento) com suporte ao *Python*.

5.4.3 IDE Pycharm

Pycharm é uma ferramenta para criação de aplicações desenvolvida pela *JetBrains* e possui duas versões: *Professional* e *Community*. A versão *Professional* é paga e com

propósito de uso comercial. A versão distribuída sob o nome de *Community é opensource* e gratuita. Esta edição possui a licença *Apache 2*, onde permite que o *software* seja livre. No entanto, os direitos de cópia e os termos de uso devem ser preservados conforme definido pelo desenvolvedor (<<https://www.apache.org/licenses/>>).

Os pacotes de instalação do Pycharm estão disponíveis para as *Windows*, *MacOS* e *Linux* no seguinte endereço: <<https://www.jetbrains.com/pycharm/download/>>. Neste endereço pode ser feito o download da ferramenta através de um pacote executável.

O projeto discutido nesta tese foi concebido através da plataforma *Windows*, assim, as informações pertinentes a configuração do SUMO diz respeito a plataforma em questão. Para informações a respeito de outras plataformas acesse o site oficial do simulador.

A aplicação desenvolvida para este trabalho está disponível para a comunidade no site *Github* no seguinte endereço: <<https://github.com/pjmatuck/mestrado.sumo>>.

6 Resultados

6.1 Considerações Iniciais

A proposta deste trabalho foi avaliar a capacidade de uma estrutura abstrata de vias em se organizar dado diferentes cenários de tráfego urbano. Essa organização foi auxiliada através do uso de uma técnica de Inteligência Artificial, o Aprendizado por Reforço. No entanto, para estabelecer um tráfego, além da malha viária, são necessários veículos, rotas e como estes veículos escolhem estas rotas.

Devido a característica microscópica do modelo de simulação adotado, o volume de carros *versus* o número de rotas disponíveis implica diretamente no custo computacional da simulação, dado que o sistema deve manter na memória uma instância para cada carro e realizar cálculos pertinentes ao seu próprio comportamento na malha.

Na intenção de otimizar o custo computacional e simplificar o modelagem do ambiente, algumas características a respeito dos veículos e as rotas foram adotados.

6.1.1 Veículos

Os carros participantes da malha viária utilizada são inseridos por um único ponto, o vértice “0/0”. A partir deste ponto nascem dois segmentos de vias, um no sentido oeste/leste e outro no sentido sul/norte. Através destes dois trechos, os carros são inseridos no ambiente e percorrem as rotas designadas.

A capacidade de entrada de carros na malha é exatamente igual a capacidade de saída, ou seja, os carros entram por um único vértice e se distribuem através das rotas, as quais se têm como destino, do mesmo modo, um vértice em comum. Neste ponto encontram-se dois segmentos de vias que possuem as mesmas características do ponto de entrada.

Ao utilizar apenas um ponto como origem dos carros, isso conduz a uma condição “física” imposta pelo próprio simulador. Os carros, além de elementos lógicos, possuem atributos espaciais como o tamanho. Desta forma, dois carros não podem ocupar o mesmo espaço na via, sendo que, se isto eventualmente ocorrer, se trata de uma colisão entre veículos, conceito não abordado por este trabalho. Outra condição se refere a própria capacidade de vazão do ponto de origem dos veículos. Um determinado número de carros são configurados para entrarem durante uma Época de Decisão. Quando um veículo a ser inserido não encontra o espaço físico necessário para entrar na via durante o tempo da Época corrente, este é descartado pelo simulador ao final desta Época. O

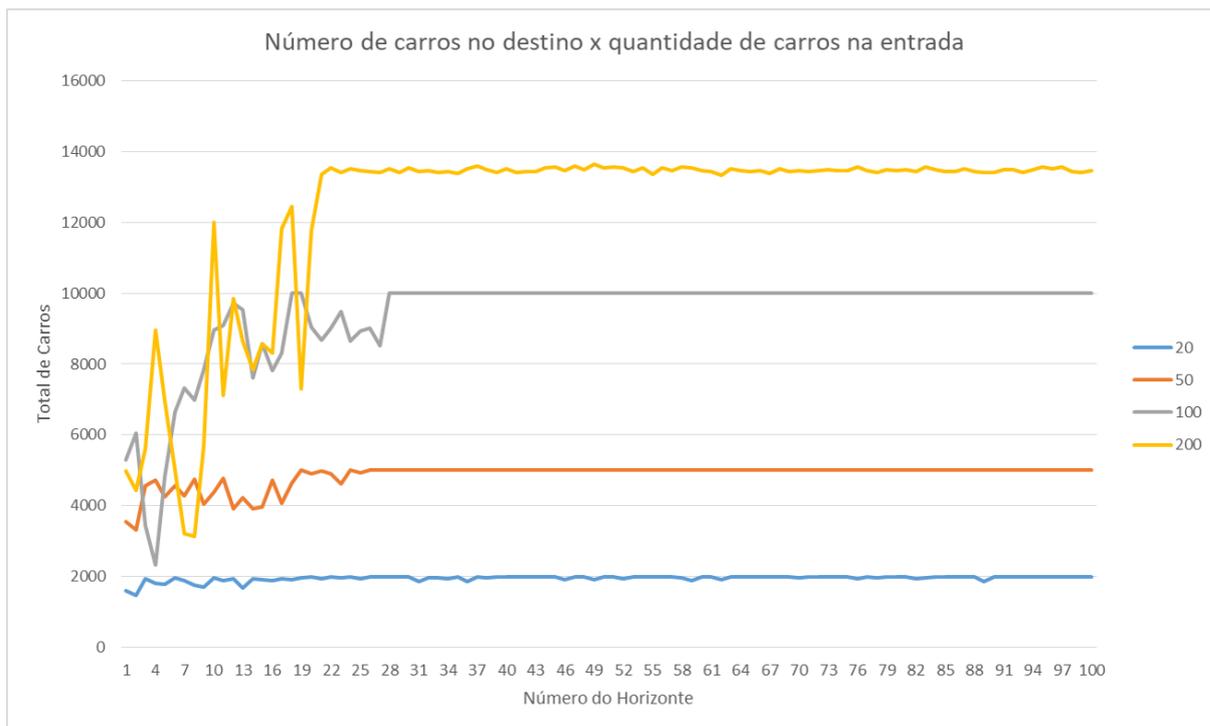


Figura 15 – Número de carros no destino x fluxo de entrada

descarte de carros pode ocorrer em duas situações: A via de entrada está congestionada, não há espaço para inserção e o tempo de espera se esgota; O volume de carros a serem inseridos é muito alto e o tempo máximo de inserção se esgota.

Com base nesta condição, o número de carros médios adotado nesta dissertação foi baseado nos testes de aumento progressivo de veículos.

Ao observar a figura 15, percebe-se que os valores mais baixos do número de carros na entrada da malha convergem para a vazão máxima destes cenários. Com 20 carros na entrada, dado que são 100 Épocas de Decisão por Horizonte, o número máximo de veículos na saída são 2000. Assim, pode-se adotar o mesmo cálculo para 50 e 100 carros, convergindo para a vazão máxima em 5000 e 10000 respectivamente. No entanto, ao elevar este número a 200 carros inseridos por Época, o AR chegou próximo a 14000, sendo que o máximo seria em 20000 veículos. Nesta condição, mesmo considerando o melhor cenário encontrado, há descarte de carros pelo simulador. Diante desta constatação, foi adotado para as demais simulações o número de 200 carros.

6.1.2 Rotas

A maneira com que foi modelado o problema nesta dissertação propõe que as vias tenham suas velocidades máximas permitidas alteradas para melhor acomodar o trânsito. Algumas vias podem adotar a velocidade igual a zero, e assim serem consideradas vias fechadas, impossibilitando o tráfego naquele trecho. Esta condição pode inviabilizar

algumas rotas, pois os carros calculam o caminho a ser percorrido antes de iniciar o trajeto. Assim, caso um carro que tenha escolhido uma rota com algum trecho de velocidade máxima permitida igual a zero, este carro ficará travado na malha (logo antes de adentrar à via) até que esta condição se altere. Se a via na qual algum veículo se encontra tiver sua velocidade alterada para zero, o veículo ficará preso no meio da via.

Para evitar estas situações de incongruência, quando uma rota é inviabilizada, o sistema escolhe uma outra rota, que ainda não foi utilizada, para substituí-la. Esta escolha é aleatória dentro do universo de rotas disponíveis para a malha viária adotada. A nova rota escolhida pode ser mais longa (utilizar mais arestas) ou mais curta (utilizar menos arestas) que a anterior. No entanto, a quantidade de rotas utilizadas durante o teste é constante visando preservar um determinado nível de exploração e ocupação da malha pelos veículos.

Assim como a quantidade de carros, o número de rotas utilizadas durante a simulação contribui para o crescimento do tempo de duração da simulação, haja vista que, quanto mais rotas disponíveis, mais cálculos são realizados no que diz respeito ao comportamento dos veículos durante a aceleração, frenagem e conversões. Desta forma, alguns testes foram realizados para encontrar o número médio de rotas que atendessem as condições de ocupação da malha sem onerar o tempo de execução da simulação (Figura 16).

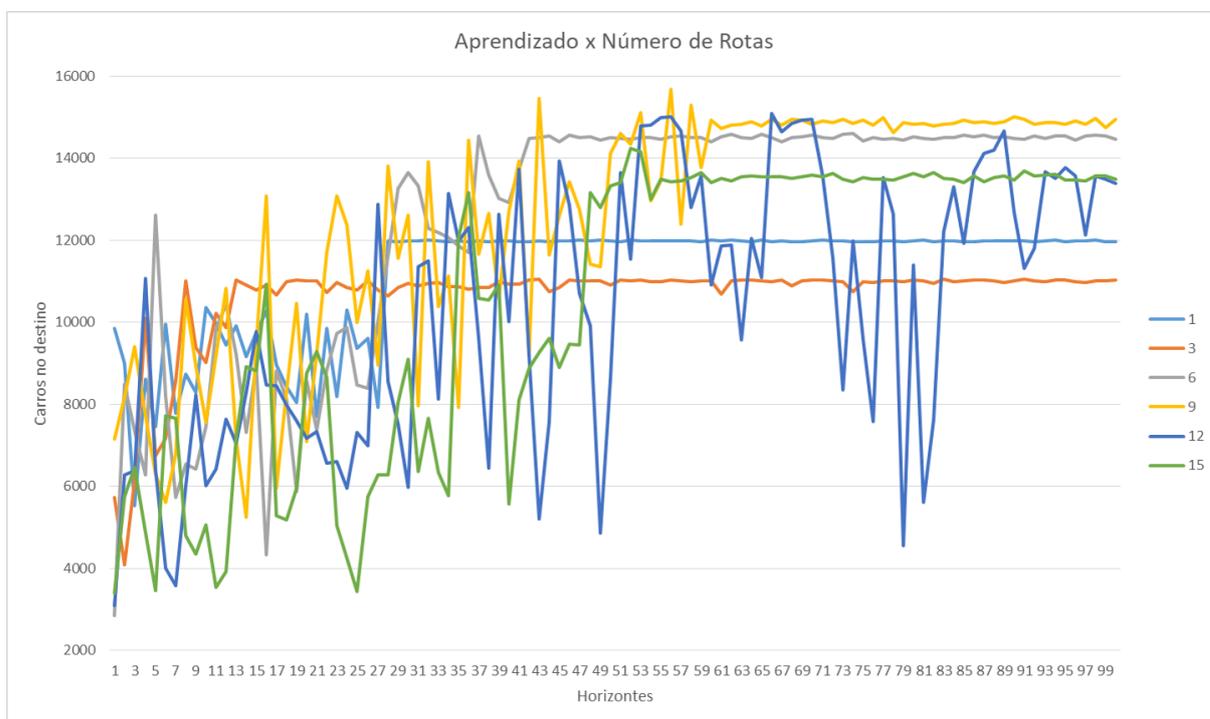


Figura 16 – Número de carros no destino x quantidade de rotas utilizadas

Em todos os casos, o AR se mostrou capaz de melhorar as condições encontrando estados melhores que os iniciais. No entanto, é possível perceber que em cenários com um número elevado de rotas (acima de 9) há mais distúrbio na malha e um tempo maior para

o MDP convergir. Desta forma, para o tamanho da malha viária adotada neste trabalho, o número de rotas utilizadas foi fixado em 6.

6.1.3 Política E-Greedy

Diferentes testes foram realizados para avaliar a melhor configuração da política ϵ -greedy a ser utilizada. De forma empírica, foi decidido dar inicialmente ao sistema uma alta capacidade de exploração e reduzi-la com o tempo, tornando então o agente mais investigativo, conforme o método de Exploração de *Boltzmann*.

Inicialmente, a capacidade do agente em escolher a melhor ação é restrita a 10%. Ao longo de 40/100 episódios, esta probabilidade atinge o nível de 97,5%, ou seja, dado que o sistema possua 100 episódios, no episódio 40, o agente terá 97,5% de chance de escolher a melhor ação disponível. No restante dos episódios o sistema mantém a mesma capacidade investigativa.

6.2 Teste de Aprendizado pela variação de Parâmetros

Com o propósito de avaliar a influência dos parâmetros do algoritmo Q-Learning no aprendizado do sistema, diferentes testes foram realizados com as seguintes configurações apresentados na Tabela 6.

Tabela 6 – Configuração de parâmetros separados por testes

Teste	Parâmetros	
	α	γ
01	0,1	0,9
02	0,5	0,9
03	0,9	0,9
04	0,1	0,5
05	0,5	0,5
06	0,9	0,5
07	0,1	0,1
08	0,5	0,1
09	0,9	0,1

Todos os cenários testados apresentaram relativa evolução, no entanto, em alguns deles o sistema demonstrou certa dificuldade em convergir.

O parâmetro α , pertinente a taxa de aprendizado da AR, refere-se basicamente a importância que novas recompensas possuem perante as recompensas já adquiridas. Valores de α próximos de zero favorecem a informação já existente na tabela-Q em detrimento do novo valor recebido. Por outro lado, valores próximos de 1 permitem a AR dar mais

peso a nova recompensa adquirida, substituindo em maior parte o valor que já existia na tabela-Q.

O fator de desconto γ mede a importância do momento no qual as recompensas são recebidas. Quando o fator assume valores muito baixos, as recompensas previamente adquiridas pelo AR se propagam de maneira “lenta”, ou seja, apenas uma pequena parcela aprendida é transmitida para o par estado-ação anterior que conduziu ao estado presente, valorizando as recompensas a curto prazo. Em contrapartida, valores mais altos permitem ao sistema transmitir a outros estados uma maior parte dos valores que já foram aprendidos, agregando-os a recompensa recebida. Esta situação faz o sistema procurar valores de aprendizado melhores a longo prazo.

O parâmetro γ ganha ainda mais relevância quando o parâmetro α se aproxima de 1 pelo fato do desconto se aplicar ao valor de aprendizado que será agregado a recompensa.

Os testes foram analisados a partir do isolamento de um parâmetro e a variação do outro, agrupando-os de três a três.

6.2.1 Testes em função do Fator de Desconto

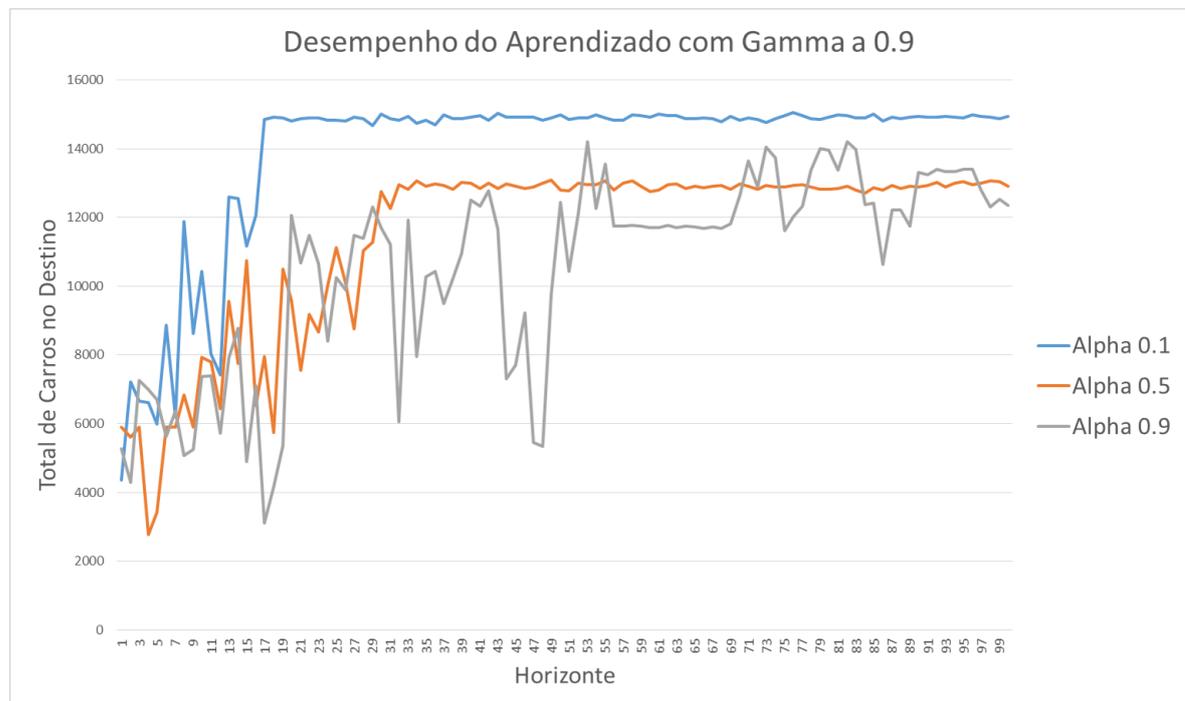


Figura 17 – Gráfico comparativo entre os Testes 01, 02 e 03.

Com base no gráfico da figura 17, o teste configurado com a taxa de aprendizado mais baixa se apresentou com os melhores resultados, encontrando estados com maior vazão de veículos e também uma convergência mais rápida, próxima ao Horizonte 20. Os demais testes se aproximaram de um ponto de convergência, no entanto, quanto maior o valor de α , maior foi a dificuldade de convergir. Isto pode ser compreendido pelo fato de

que taxas de aprendizado próximas de 1 dão ainda mais relevância ao fator de desconto. Quando ambos os parâmetros estão próximos de um, o sistema tende a se comportar de forma instável.

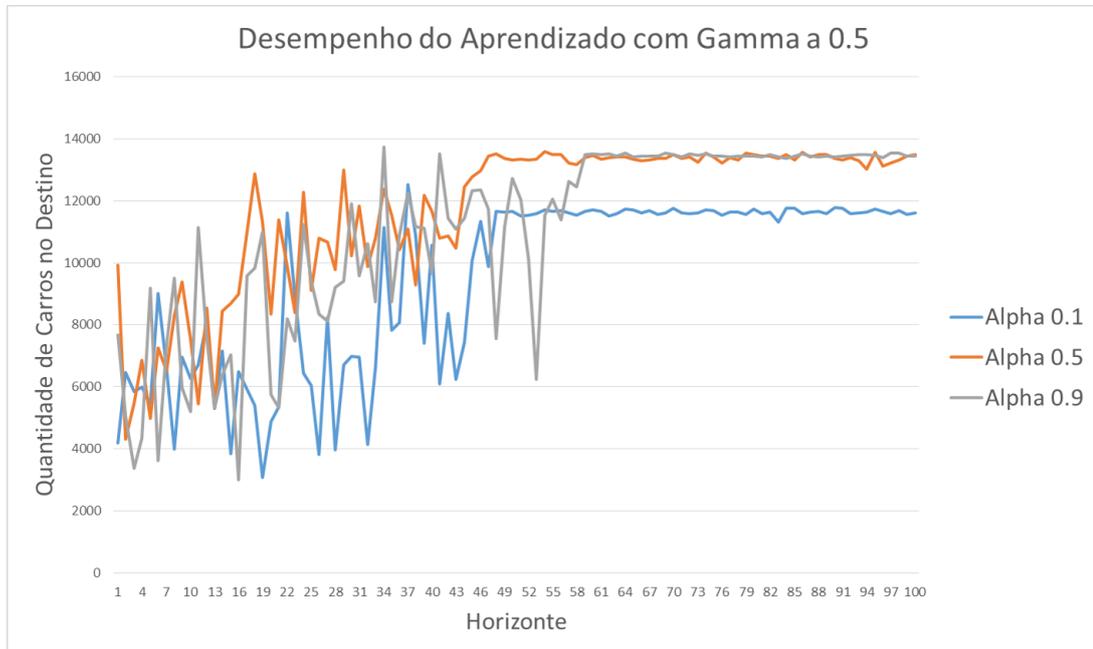


Figura 18 – Gráfico comparativo entre os Testes 04, 05 e 06.

Os testes realizados com o fator de desconto a 0,5 (gráfico da figura 18), em média, se mostraram com relativa dificuldade de convergência, atingindo a estabilização próxima do Horizonte 50. Outro fator observado em comparação aos demais cenários foi a uma maior variação no período anterior a convergência.

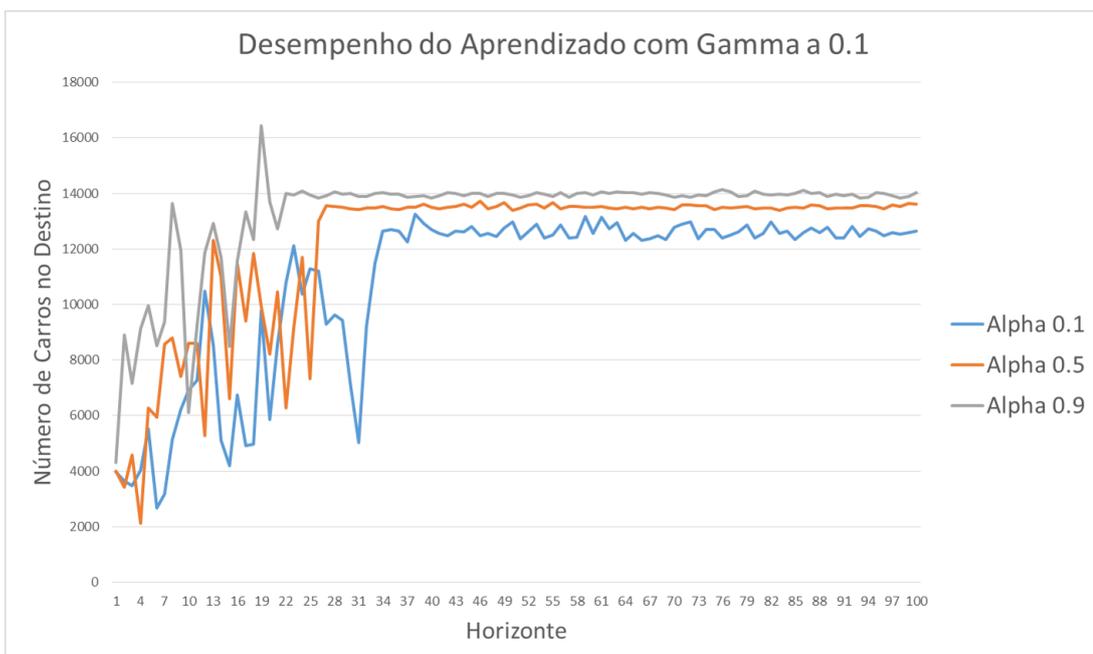


Figura 19 – Gráfico comparativo entre os Testes 07, 08 e 09.

Os testes 07, 08 e 09, observados no gráfico da figura 19, com o parâmetro γ em 0,1, apresentaram comportamento semelhante. A diferença entre os testes ficou a cargo do tempo de convergência que se comportou de forma inversa ao valor da taxa de aprendizado. Quanto maior o valor do parâmetro α , mais alto foi o Horizonte no qual o teste convergiu nos melhores estados encontrados.

6.2.2 Testes em função da Taxa de Aprendizagem

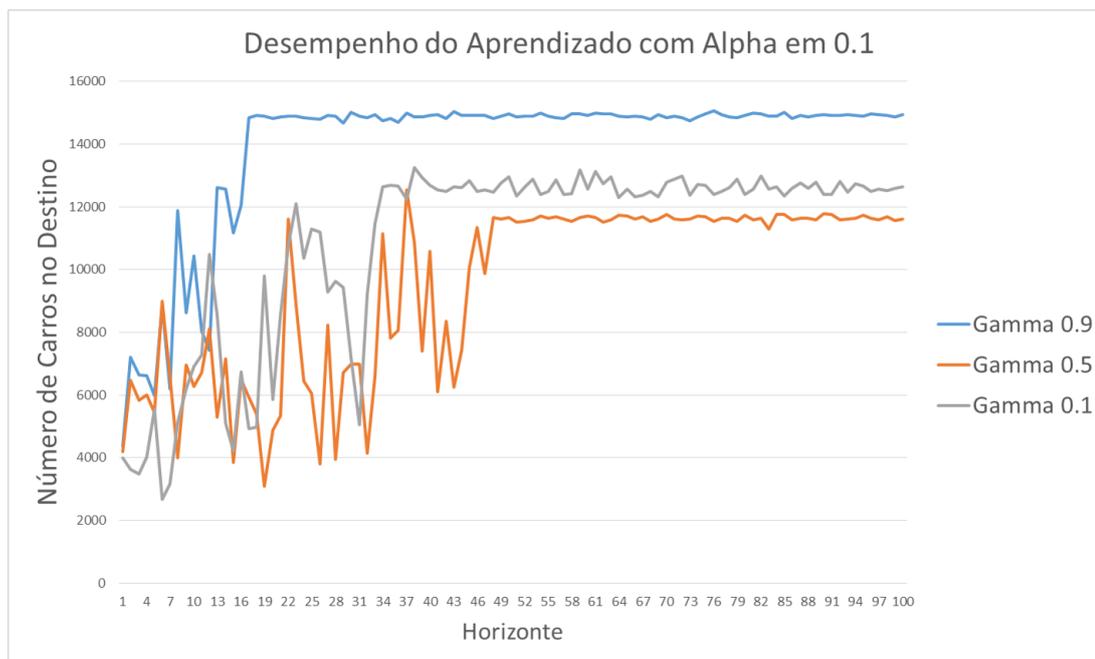


Figura 20 – Gráfico comparativo entre os Testes 01, 04 e 07.

Conforme observado anteriormente, o Teste 01 obteve o melhor desempenho em número de carros no destino que os demais, além de apresentar uma estabilização mais rápida, obteve êxito em encontrar os melhores estados comparado aos testes 04 ($\gamma = 0,5$) e 07 ($\gamma = 0,1$). Com Fatores de Desconto mais baixos nos outros testes, o sistema apresentou maior instabilidade, inclusive após a convergência.

Os testes 02, 05 e 08 (gráfico da figura 21, que envolvem a taxa de aprendizado em 0,5, convergiram a níveis próximos a 13000 carros. A variação do número de veículos no destino entre os horizontes no período anterior a estabilização também se mostrou menor que nos outros cenários, sendo que esta variação aumentou na medida que o parâmetro γ diminuiu.

A convergência dos três testes apresentados no gráfico da figura 22 aconteceu entre 13000 e 14000 veículos, no entanto, com certa dificuldade e em horizontes espaçados. Com o parâmetro α próximo de 1, os efeitos do Fator de Desconto influenciam de forma mais clara no sistema. Com uma Taxa de Aprendizagem alta, é perceptível que o ambiente se mostra mais estável com o fator de desconto menor.

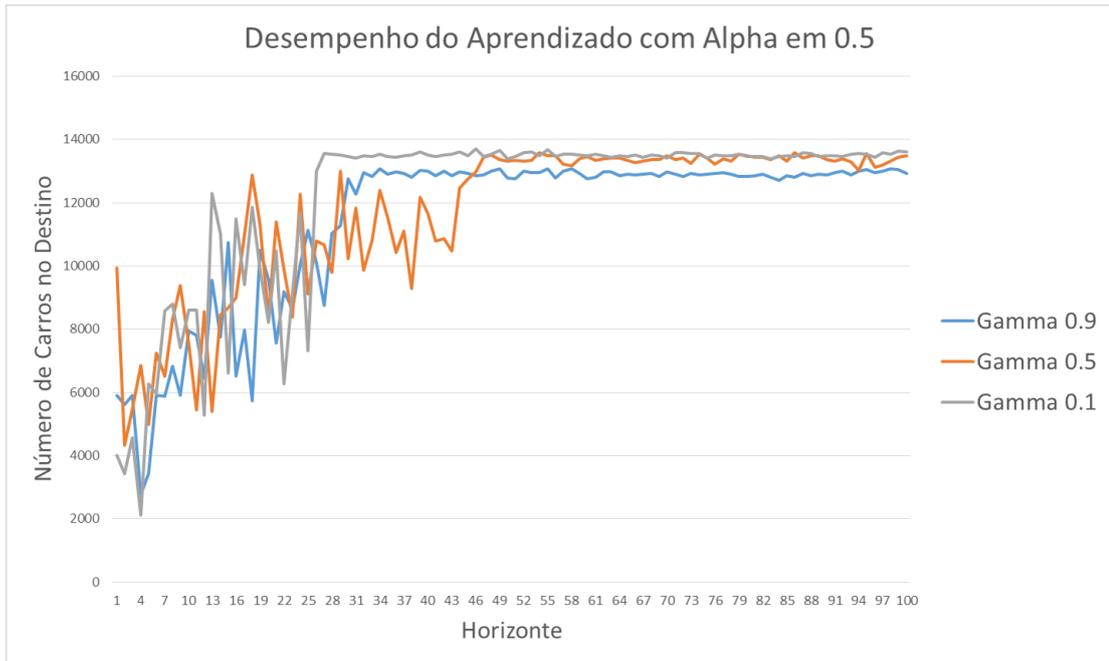


Figura 21 – Gráfico comparativo entre os Testes 02, 05 e 08.

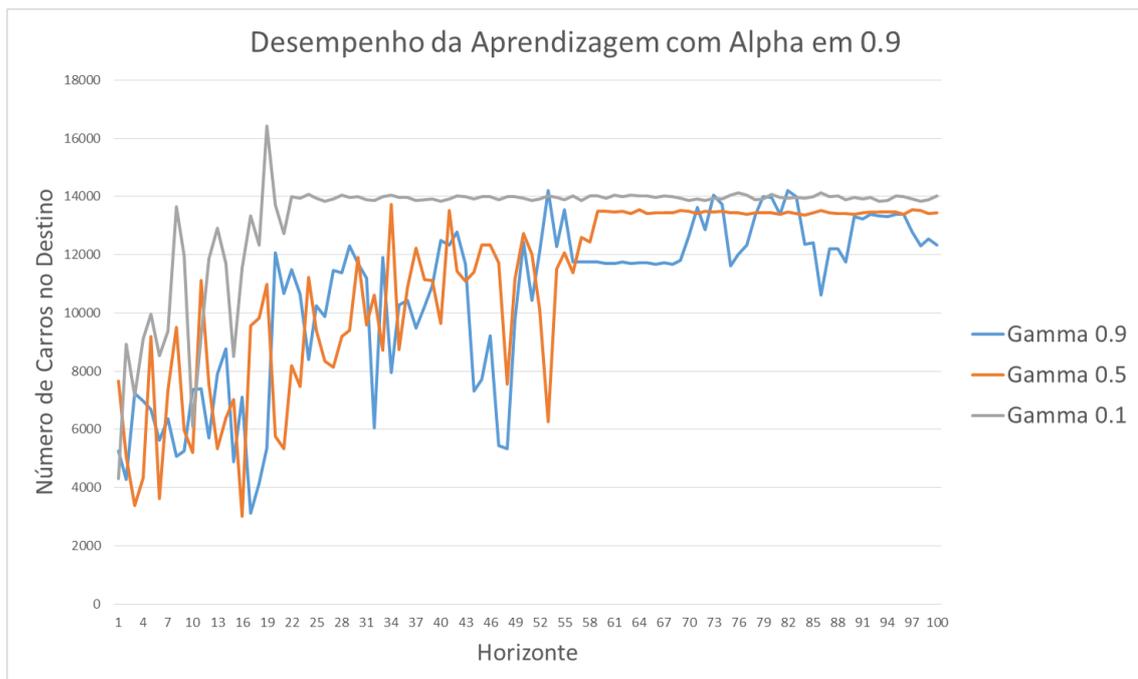


Figura 22 – Gráfico comparativo entre os Testes 03, 06 e 09.

O cenário que apresentou melhor resultado foi aquele configurado com um Fator de Desconto alto (0,9) e a Taxa de Aprendizagem baixa (0,1).

6.3 Teste comparativo em um cenário com e sem Aprendizagem por Reforço

A realização do teste comparativo entre os ambientes com a presença e a ausência da técnica de Aprendizado por Reforço foi feita com o cenário que apresentou melhor desempenho quanto a Aprendizagem *versus* o melhor resultado de um cenário sem o emprego da técnica de IA após 8 execuções.

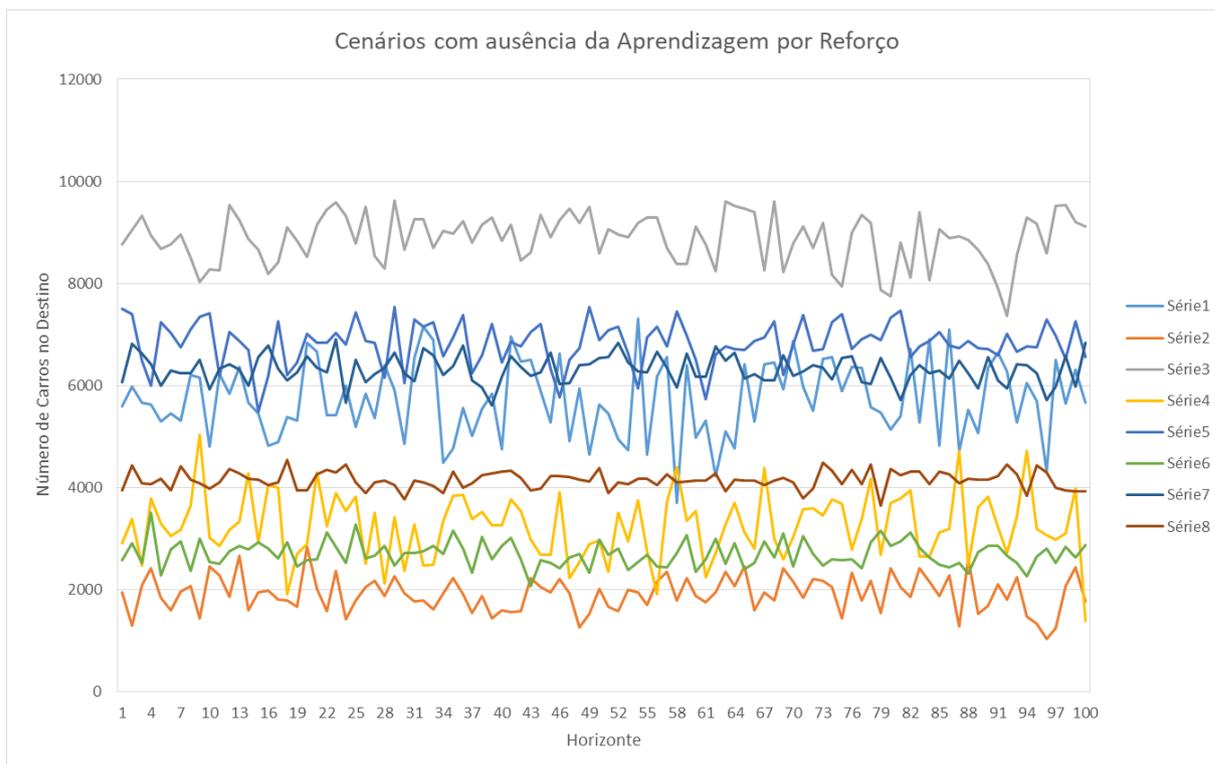


Figura 23 – Gráfico comparativo das execuções do sistema sem AR.

É possível observar no gráfico 24 que o sistema foi capaz de encontrar condições de tráfego melhores que aquelas apresentadas no ambiente sem Aprendizagem por Reforço.

Um trânsito é formado por decisões descentralizadas, ou seja, cada motorista decide a rota que irá seguir diante diferentes circunstâncias. Por mais que aplicativos de navegação acessíveis através *smartphones* auxiliem nesta tomada de decisão, muitas vezes estas informações são imprecisas e geram um efeito contrário, como pode ser observado em (UOL, 2018).

Neste caso, a Inteligência Artificial adotada por esta dissertação funciona como um bloco central capaz de observar todo o estado do ambiente, ou seja, a situação do trânsito em dado momento, e tomar decisões a fim de equilibrá-lo com foco em aumentar a vazão de veículos.

Outro fator observado foi relacionado a média da velocidade máxima permitida de todas as vias da malha viária em ambos os casos. Em ambientes sem o emprego da

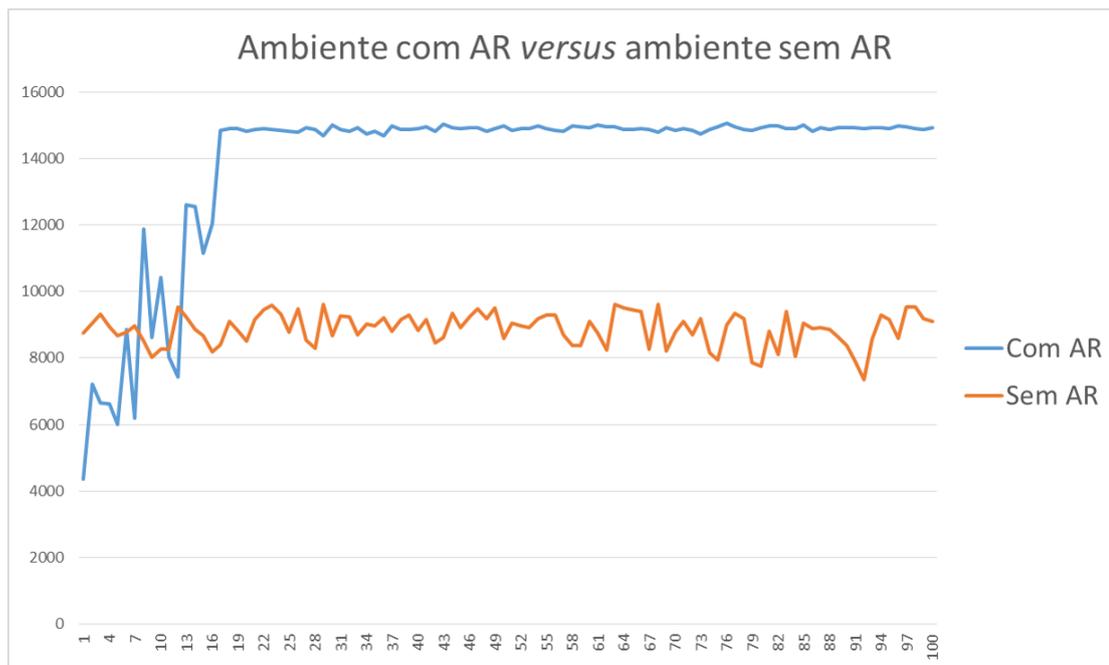


Figura 24 – Comparação entre os cenários com e sem Aprendizagem por Reforço

Aprendizagem por Reforço, a velocidade pré-definida das vias não se altera, então, a média das velocidades permanece em aproximadamente 48 km/h ($13,31 \text{ m/s}^2$).

O processo de aprendizagem do sistema dava a ele a capacidade de alterar a velocidade máxima permitida de uma via na intenção de encontrar um melhor equilíbrio entre as rotas. Ao convergir, foi observado também que a média da velocidade das vias foi mais baixa do que a média observada no ambiente sem a aplicação da AR, como pode ser observado no gráfico da figura 25.

A partir da obtenção das velocidades máximas permitidas de 10 execuções da simulação, a média das velocidades se encontrou em $45,49 \text{ km/h}$, com Variância em 0,196 e Desvio Padrão 0,442. Esta condição sugere que o simples aumento da velocidade máxima permitida de uma via não implica em um maior fluxo de veículos. Na medida em que a vazão de carros aumenta e se estabiliza, a velocidade média reduz e também demonstra uma menor variação.

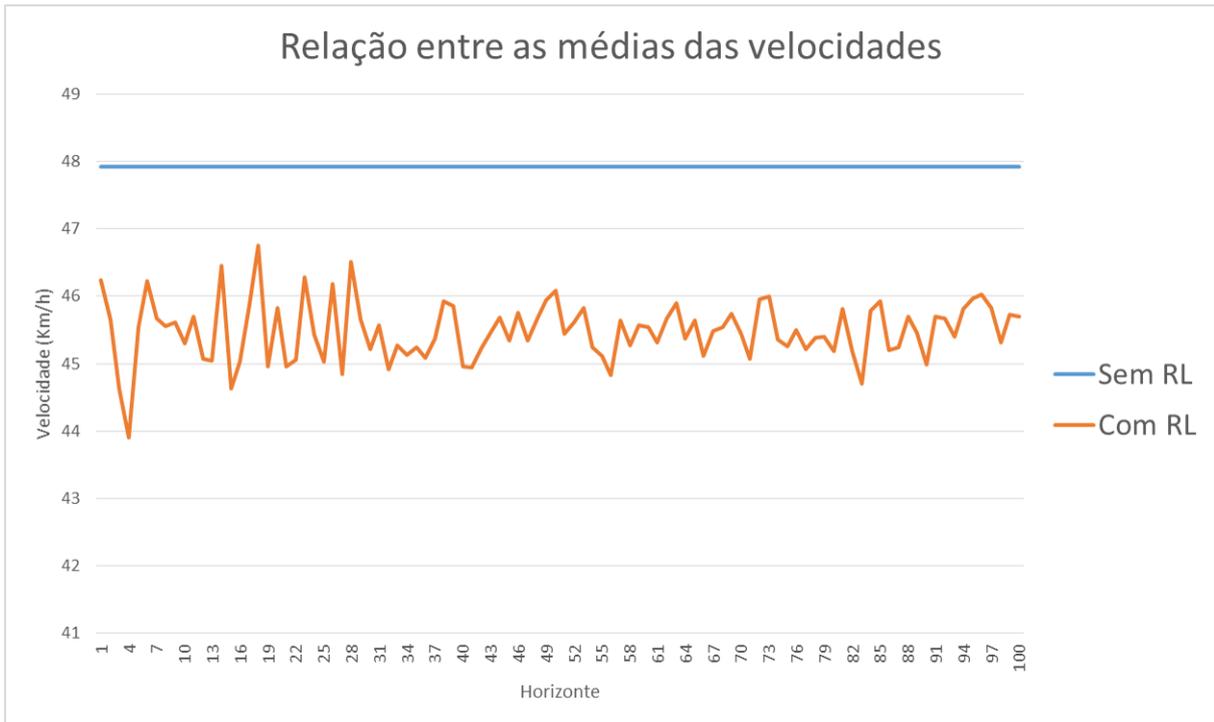


Figura 25 – Comparação entre as médias das velocidades entre os cenários com e sem AR

6.4 Uso de um sistema treinado em um ambiente sem treinamento

Este teste consistiu em aplicar um aprendizado prévio em um ambiente sem treinamento. Uma nova simulação foi realizada, aqui definida por *Teste A*, com os seguintes parâmetros:

- Taxa de Aprendizado: 0,1;
- Fator de Desconto: 0,9;

O Aprendizado realizado no *Teste A* foi empregado em outro teste, denominado de *Teste B*. A aplicação do aprendizado prévio foi feita simplesmente inicializando o segundo teste com as informações da Tabela-Q do primeiro teste. A partir da tabela-Q é possível saber a política resultante do MDP do *Teste A*.

O *Teste A* apresentou convergência próxima ao horizonte 37 com uma vazão de aproximadamente 13000 veículos. O *Teste B* demonstrou capacidade de melhorar os níveis de Números de Carros no Destino apresentado pelo *Teste A*, atingindo um pico acima de 18000 veículos e uma média próxima de 15000 a 16000. No entanto, nota-se uma dificuldade de estabilização e momentos em que o sistema regrediu a níveis muito baixos, ressaltando o momento entre o horizonte 85 e 97 (Figura 26).

Esta instabilidade pode ser compreendida pela diferença das condições existentes entre o *Teste A* e o *Teste B*. Devido a aleatoriedade empregada na concepção do ambiente

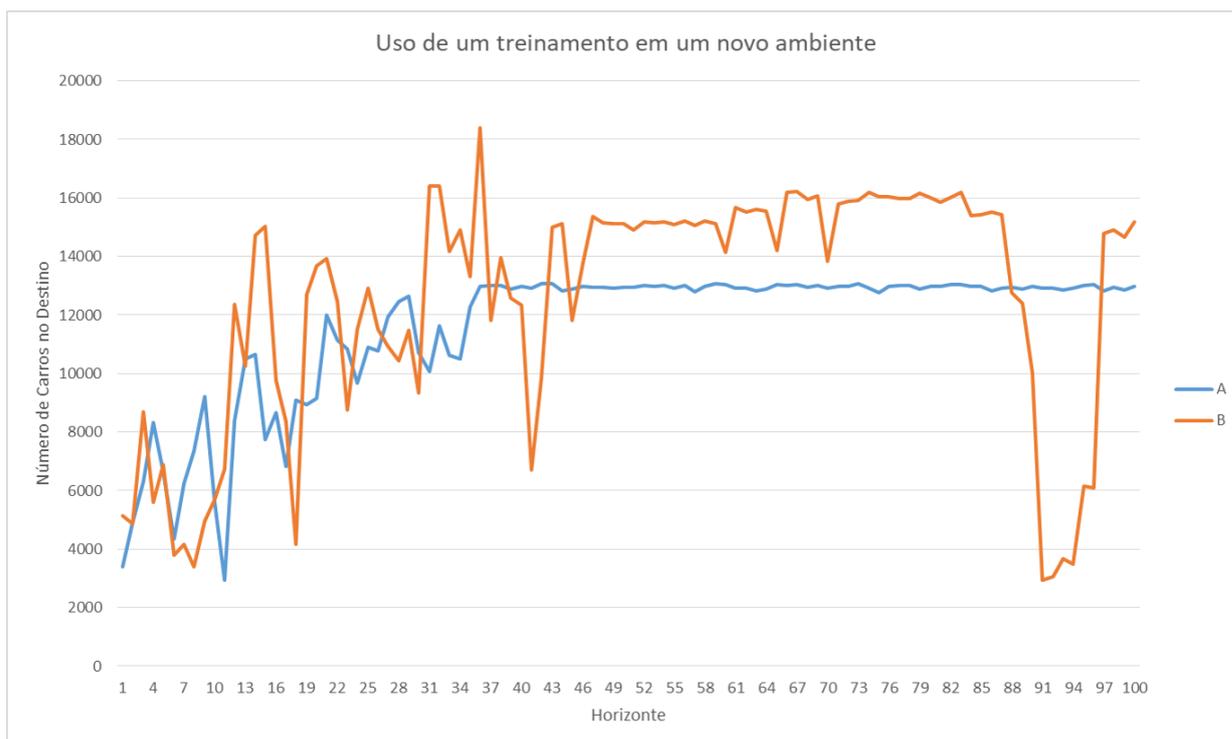


Figura 26 – Avaliação do desempenho do uso de um sistema treinado em outro ambiente sem treinamento.

de ambos os casos, é alta a chance de que rotas diferentes tenham sido escolhidas criando então um comportamento muito distinto entre um caso e outro. Analogamente, é como se o sistema fosse treinado de acordo as condições de trânsito em uma cidade e fosse aplicado em outra, que possui um tráfego diferente.

7 Conclusão

A modelagem de problemas relacionados ao tráfego de veículos se mostrou complexa no que tange a representação dos elementos envolvidos. A forma com que estados e ações são dispostos no ambiente pode inviabilizar o uso do Processo Decisório de Markov. Em face desse primeiro obstáculo, a malha viária utilizada e a forma em como as ações influenciariam no ambiente, foram pensados para que as técnicas adotadas por este trabalho se tornassem aplicáveis.

A complexidade deste problema se estendeu no uso do simulador. Apesar de ser um *software* amplamente utilizado nos meios acadêmicos em diferentes frentes de pesquisa, foi necessário desenvolver uma Aplicação auxiliar para atender a todos os requisitos manifestados a partir do questionamento desta dissertação. Esta aplicação foi responsável por definir todo o ciclo de vida de aprendizagem, desde a conexão com o simulador até a geração de relatórios para análise.

Mesmo com a preocupação quanto a explosão de estados, o MDP resultante do modelo adotado analisado trouxe um certo grau de incerteza quanto ao comportamento dos testes. Esta incerteza se deu pelos próprios recursos aleatórios intrínsecos a técnica utilizada. O uso da política de Exploração x Investigação teve como propósito reduzir esta incerteza, entretanto, em alguns casos o sistema encontrou ações com altas recompensas rapidamente, em outros momentos apresentou uma demora para encontrá-las. Este fato influenciou na quantidade de carros que chegariam ao destino no fim de um horizonte e também no momento em que ocorria a convergência do teste, que, apesar de um comportamento padrão, apresentaram níveis diferentes a cada execução.

No entanto, vale ressaltar que a técnica de Aprendizado por Reforço aplicada ao modelo de tráfego adotado nesta dissertação, se mostrou capaz de "conduzir" o ambiente à melhores pontos de equilíbrio quanto ao fluxo de carros, favorecendo o aumento da vazão de veículos em diferentes circunstâncias, mesmo com configurações diferentes de aprendizado. Associado ao aumento da vazão, outro fator de grande importância foi a redução da média da velocidade máxima utilizada em toda a malha. Este ponto indicou que o aumento de velocidade de uma malha não necessariamente trás ganhos quanto a vazão de veículos, sem contar que é de conhecimento público que velocidades mais altas empregam maiores riscos as pessoas envolvidas.

A diminuição da velocidade média e o aumento da vazão de um conjunto de vias implicam indiretamente na redução da poluição, com menor uso de combustíveis e menor permanência dos veículos nas ruas, além de outros fatores econômicos, sociais e ambientais.

O uso de Tecnologias de Comunicação e Informação, associado aos cenários de

tráfego urbano, apresentam um caminho para trazer melhorias aos sistemas utilizados atualmente. Novas formas de comunicação que integram, além de pessoas, carros e infraestrutura, pode ser um método de centralizar a tomada de decisões em um meio tão caótico e auxiliar de forma mais precisa os motoristas envolvidos. Além do mais, este caminho pode reduzir o impacto das más decisões eventualmente tomadas pelos governantes.

Apesar deste trabalho apresentar um caminho, é necessário compreender que existem outros fatores a serem analisados para que um sistema de trânsito passe a operar envolvido desta maneira por um algoritmo. Como por exemplo:

- Uso de malhas viárias irregulares e mais complexas;
- Utilização de veículos de características diferentes;
- Aplicação do Aprendizado associado ao uso de semáforos;
- Relação entre o fluxo de pedestres e o controle dinâmico de tráfego urbano;
- Entre inúmeros outros fatores.

Outro ponto de destaque é o caráter inovador que este trabalho apresenta. Técnicas de Inteligência Artificial são amplamente utilizadas em problemas de trânsito, no entanto, esta dissertação propõe pensar de uma forma não usual. Nota-se que os trabalhos científicos que embasaram esta pesquisa retratam em parte o cenário quanto ao estudo de problemas de tráfego urbano, empregando a maior parte dos esforços em encontrar novas formas de funcionamento de semáforos, fornecer maior autonomia aos veículos e modelar a rede de comunicação entre os elementos de uma malha viária, carros e estrutura. Obviamente, todos os trabalhos são únicos e de grande valor na contribuição para o avanço da ciência, ainda assim, é de extrema e fundamental importância buscar novas formas de pensar e encarar os problemas que a sociedade enfrenta diariamente.

Referências

- ABDOOS, M.; MOZAYANI, N.; BAZZAN, A. L. Traffic light control in non-stationary environments based on multi agent q-learning. In: IEEE. *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. [S.l.], 2011. p. 1580–1585. Citado na página 23.
- ADLER, J. L. et al. A multi-agent approach to cooperative traffic management and route guidance. *Transportation Research Part B: Methodological*, Elsevier, v. 39, n. 4, p. 297–318, 2005. Citado na página 23.
- AKHAVAN-TABATABAEI, R.; SÁNCHEZ, D. M.; YEUNG, T. G. A markov decision process model for cervical cancer screening policies in colombia. *Medical Decision Making*, SAGE Publications Sage CA: Los Angeles, CA, v. 37, n. 2, p. 196–211, 2017. Citado na página 25.
- ALHEETI, K. M. A.; GRUEBLER, A.; MCDONALD-MAIER, K. D. An intrusion detection system against malicious attacks on the communication network of driverless cars. In: IEEE. *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. [S.l.], 2015. p. 916–921. Citado na página 23.
- BAZZAN, A. L. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multi-Agent Systems*, Springer, v. 18, n. 3, p. 342, 2009. Citado na página 23.
- BEHRISCH, M. et al. Sumo—simulation of urban mobility: an overview. In: THINKMIND. *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. [S.l.], 2011. Citado na página 31.
- BONGAARTS, J. Human population growth and the demographic transition. *Philosophical Transactions of the Royal Society B: Biological Sciences*, The Royal Society, v. 364, n. 1532, p. 2985–2990, 2009. Citado na página 14.
- BRACKSTONE, M.; MCDONALD, M. Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, Elsevier, v. 2, n. 4, p. 181–196, 1999. Citado na página 19.
- CHANDLER, R. E.; HERMAN, R.; MONTROLL, E. W. Traffic dynamics: studies in car following. *Operations research*, Informs, v. 6, n. 2, p. 165–184, 1958. Citado na página 19.
- CHANG, Y.-H.; LEE, M.-S. Incorporating markov decision process on genetic algorithms to formulate trading strategies for stock markets. *Applied Soft Computing*, Elsevier, v. 52, p. 1143–1153, 2017. Citado na página 25.
- CHENG, S.-F.; NGUYEN, T. D. Taxisim: A multiagent simulation platform for evaluating taxi fleet operations. In: IEEE COMPUTER SOCIETY. *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 02*. [S.l.], 2011. p. 14–21. Citado na página 23.

- CHOURABI, H. et al. Understanding smart cities: An integrative framework. In: *IEEE. System Science (HICSS), 2012 45th Hawaii International Conference on*. [S.l.], 2012. p. 2289–2297. Citado na página 14.
- CISCO. *Smart Connected Communities*. 2017. Disponível em: <<https://www.cisco.com/c/en/us/solutions/industries/smart-connected-communities.html>>. Citado na página 15.
- COSTA, G. M.; BASTOS, G. S. Semáforo inteligente—uma aplicação de aprendizagem por reforço. 2012. Citado 2 vezes nas páginas 23 e 24.
- DALLMEYER, J. et al. Don't go with the ant flow: Ant-inspired traffic routing in urban environments. *Journal of Intelligent Transportation Systems*, Taylor & Francis, v. 19, n. 1, p. 78–88, 2015. Citado na página 23.
- DENATRAN. *Frota Total de Veículos - 2018*. 2018. Disponível em: <<https://www.denatran.gov.br/estatistica/635-frota-2018>>. Citado na página 16.
- DESJARDINS, C.; CHAIB-DRAA, B. Cooperative adaptive cruise control: A reinforcement learning approach. *IEEE Transactions on intelligent transportation systems*, IEEE, v. 12, n. 4, p. 1248–1260, 2011. Citado na página 23.
- DIMITRAKOPOULOS, G.; DEMESTICHAS, P. Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, IEEE, v. 5, n. 1, p. 77–84, 2010. Citado na página 21.
- 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. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 14, n. 3, p. 1140–1150, 2013. Citado na página 23.
- EMMELMANN, M.; BOCHOW, B.; KELLUM, C. *Vehicular networking: Automotive applications and beyond*. [S.l.]: John Wiley & Sons, 2010. v. 2. Citado na página 21.
- ERICSSON. *Smart Sustainable Cities*. 2017. Disponível em: <<https://www.ericsson.com/en/industries/smart-sustainable-cities>>. Citado na página 15.
- FOLHA. *Marginais têm aumento de acidentes pelo 4º mês seguido após novos limites*. 2017. Disponível em: <<http://www1.folha.uol.com.br/cotidiano/2017/06/1891548-marginais-tem-aumento-de-acidentes-pelo-4-mes-seguido-apos-novos-limites.shtml>>. Citado na página 17.
- FU, M. C. Markov decision processes, alphago, and monte carlo tree search: Back to the future. In: *Leading Developments from INFORMS Communities*. [S.l.]: INFORMS, 2017. p. 68–88. Citado na página 25.
- G1. *Marginais têm 117 acidentes e duas mortes no segundo mês após aumento da velocidade*. 2017. Disponível em: <<https://g1.globo.com/sao-paulo/noticia/marginais-tem-117-acidentes-e-duas-mortes-no-segundo-mes-apos-aumento-da-velocidade.ghtml>>. Citado na página 17.
- GAN, J. et al. Optimal pricing for improving efficiency of taxi systems. In: *IJCAI*. [S.l.: s.n.], 2013. p. 2811–2818. Citado na página 23.

- GIFFINGER, R. et al. *City-ranking of European Medium-Sized Cities*. [S.l.]: Vienna Univ. of Technology, Delft University of Technology, 2007. Citado na página 15.
- GIPPS, P. G. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, Elsevier, v. 15, n. 2, p. 105–111, 1981. Citado na página 19.
- GREENSHIELDS, B. et al. A study of traffic capacity. In: NATIONAL RESEARCH COUNCIL (USA), HIGHWAY RESEARCH BOARD. *Highway research board proceedings*. [S.l.], 1935. v. 1935. Citado na página 19.
- GREENSHIELDS, B. D. et al. The photographic method of studying traffic behavior. In: *Highway Research Board Proceedings*. [S.l.: s.n.], 1934. v. 13. Citado na página 19.
- HALL, R. E. et al. *The vision of a smart city*. [S.l.], 2000. Citado na página 14.
- HARRISON, C.; DONNELLY, I. A. A theory of smart cities. In: *Proceedings of the 55th Annual Meeting of the ISSS-2011, Hull, UK*. [S.l.: s.n.], 2011. v. 55, n. 1. Citado na página 14.
- IBM. *Smarter Cities*. 2017. Disponível em: <https://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/>. Citado na página 15.
- IRUNGBAM, R. S. The model of smart cities in theory and in practice. *Journal for Studies in Management and Planning*, v. 2, n. 4, p. 156–187, 2016. Citado na página 15.
- KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research*, v. 4, p. 237–285, 1996. Citado na página 25.
- KARNADI, F. K.; MO, Z. H.; LAN, K.-c. Rapid generation of realistic mobility models for vanet. In: IEEE. *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*. [S.l.], 2007. p. 2506–2511. Citado na página 21.
- KEREKES, J. P. et al. Sensor modeling and demonstration of a multi-object spectrometer for performance-driven sensing. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*. [S.l.], 2009. v. 7334, p. 73340J. Citado na página 22.
- KHATIB, A. E. et al. A cooperative detection model based on artificial neural network for vanet qos-olsr protocol. In: IEEE. *Ubiquitous Wireless Broadband (ICUWB), 2015 IEEE International Conference on*. [S.l.], 2015. p. 1–5. Citado na página 23.
- KOURTIT, K.; NIJKAMP, P.; PARTRIDGE, M. D. *The new urban world*. [S.l.]: Taylor & Francis, 2013. Citado na página 14.
- KRAJZEWICZ, D. et al. Recent development and applications of sumo-simulation of urban mobility. *International Journal On Advances in Systems and Measurements*, v. 5, n. 3&4, 2012. Citado 2 vezes nas páginas 21 e 33.
- KRAUSS, S. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. Tese (Doutorado), 1998. Citado na página 35.

- LIEBERMAN, E. B. Brief history of traffic simulation. *Traffic and Transportation Simulation*, p. 17, 2014. Citado na página 20.
- MEDINA, J. C.; HAJBABAIE, A.; BENEKOHAL, R. F. Arterial traffic control using reinforcement learning agents and information from adjacent intersections in the state and reward structure. In: IEEE. *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. [S.l.], 2010. p. 525–530. Citado na página 23.
- NEWELL, G. F. A simplified car-following theory: a lower order model. *Transportation Research Part B: Methodological*, Elsevier, v. 36, n. 3, p. 195–205, 2002. Citado na página 19.
- OH, S.-Y.; LEE, J.-H.; CHOI, D.-H. A new reinforcement learning vehicle control architecture for vision-based road following. *IEEE Transactions on Vehicular Technology*, IEEE, v. 49, n. 3, p. 997–1005, 2000. Citado na página 23.
- PACIONI, C. et al. Augmenting the conservation value of rehabilitated wildlife by integrating genetics and population modelling in the post-rehabilitation decision process. *Current Zoology*, 2017. Citado na página 25.
- PARAMICS. 2018. Disponível em: <<http://www.paramics-online.com/>>. Citado na página 21.
- PELLEGRINI, J.; WAINER, J. Processos de decisão de markov: um tutorial. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 133–179, 2007. Citado na página 26.
- PIORKOWSKI, M. et al. Trans: realistic joint traffic and network simulator for vanets. *ACM SIGMOBILE mobile computing and communications review*, ACM, v. 12, n. 1, p. 31–33, 2008. Citado na página 21.
- PIPES, L. A. An operational analysis of traffic dynamics. *Journal of applied physics*, AIP, v. 24, n. 3, p. 274–281, 1953. Citado na página 19.
- PTV-VISSIM. 2018. Disponível em: <<http://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/>>. Citado na página 21.
- REDE BRASIL ATUAL. *Após aumento da velocidade por Doria, marginais têm 60% mais acidentes*. 2017. Disponível em: <<http://www.redebrasilatual.com.br/cidadania/2017/03/apos-aumento-da-velocidade-por-doria-marginais-registram-60-mais-acidentes>>. Citado na página 17.
- SAHA, S.; ROY, U.; SINHA, D. Neural network based modified aodv routing protocol in vanet. *European Journal of Advances in Engineering and Technology*, v. 2, n. 10, p. 17–25, 2015. Citado na página 23.
- SILVA, F. L. S.; COSTA, G. M.; BASTOS, G. S. Semaforo inteligente-integracao de aprendizagem por reforco e a técnica cmac. 2013. Citado 2 vezes nas páginas 23 e 24.
- SINDIPECAS. *Relatório de Frota Circulante de 2018*. 2018. Disponível em: <<https://www.sindipecas.org.br/area-atuacao/?co=s&a=frota-circulante>>. Citado na página 16.
- SUMO. 2018. Disponível em: <<http://sumo.dlr.de/wiki/SUMO>>. Citado na página 21.

UOL, S. P. *Estudo prova que Waze e Google Maps, na verdade, atrapalham o trânsito*. 2018. Disponível em: <<https://tecnologia.uol.com.br/noticias/redacao/2018/03/16/estudo-prova-que-waze-e-google-maps-na-verdade-atrapalham-o-transito.htm>>. Citado na página 59.

USP, J. da. *Políticas de velocidade nas marginais carecem de dados*. 2017. Disponível em: <<https://jornal.usp.br/atualidades/politicas-de-velocidade-nas-marginais-carecem-de-dados/>>. Citado na página 17.

WAGENINGEN-KESSELS, F. van et al. Genealogy of traffic flow models. *EURO Journal on Transportation and Logistics*, Springer, v. 4, n. 4, p. 445–473, 2015. Citado 2 vezes nas páginas 19 e 20.

ZHENG, Y.; LIU, B. Fuzzy vehicle routing model with credibility measure and its hybrid intelligent algorithm. *Applied mathematics and computation*, Elsevier, v. 176, n. 2, p. 673–683, 2006. Citado na página 23.