Reinforcement Learning in Urban Network Traffic Signal Control: A Systematic Literature Review

Mohammad Noaeen[a,*], Atharva Naik[b], Liana Goodman[b], Jared Crebo[c,d], Taimoor Abrar[b], Behrouz Far[b], Zahra Shakeri Hossein Abad[e], and Ana L. C. Bazzan[f]

[a]Department of Civil and Mineral Engineering, University of Toronto, Toronto, Canada
[b]Department of Electrical and Software Engineering, University of Calgary, Calgary, Canada
[c]Department of Mechanical and Manufacturing Engineering, University of Calgary, Calgary, Canada
[d]Department of Physics and Astronomy, University of Calgary, Calgary, Canada
[e]Cumming School of Medicine, University of Calgary, Calgary, Canada
[f] Institute of Informatics, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil

Abstract

Improvement of traffic signal control (TSC) efficiency has been found to lead to improved urban transportation and enhanced quality of life. Recently, the use of reinforcement learning (RL) in various areas of TSC has gained significant traction; thus, we conducted a systematic literature review as a systematic, comprehensive, and reproducible review to dissect all the existing research that applied RL in the network-level TSC (NTSC) domain. The review only targeted the network-level articles that tested the proposed methods in networks with two or more intersections. We used natural language processing to define the search strings and searched Google Scholar, Web of Science, IEEE Xplore, ACM Digital Library, Springer Link, and Science Direct databases. This review covers 160 peer-reviewed articles from 30 countries published from 1994 to March 2020. The goal of this study is to provide the research community with statistical and conceptual knowledge, summarize existence evidence, characterize RL applications in NTSC domains, explore all applied methods and major first events in the defined scope, and identify areas for further research based on the explored research problems in current research.

Reinforcement Learning; traffic light control; urban network; multi-agent system; intelligent transportation system; artificial intelligence.

1. Introduction

With an explosion in urban and rural population rates, city transportation systems become less efficient at handling the ever-growing numbers of commuters. A lack of space and resources with which to improve infrastructure poses problems in accommodating the increasing urban population. The resulting congestion further leads to increased pollution caused by sitting idle in traffic jams, traffic delays and bottlenecks, and a rise in accidents. The secondary issues that arise are just as severe, including economic loss and an overall decrease in quality of life. This presents the problem of improving traffic flow and traffic signal control (TSC) within the already existing infrastructure.

Traffic signals are most often regularised through fixed time, actuated, or adaptive control methods, whether the state-of-the-art methods or the methods deployed in the real-world, such as SCATS (Sims et al., 1981), SCOOT (Hunt et al., 1981), and TUC (Diakaki et al., 2002). Fixed time signal control involves a repeating pattern that does not change with the live traffic situation and which continues through its cycles regardless of dynamic traffic changes

* *Corresponding author. E-mail: m.noaeen@utoronto.ca*

in that area. The actuated control method operates traffic signals based on real-time data of loop detectors. Despite being traffic responsive, the actuated control method is not designed to fully address fluctuating traffic demands, thereby rendering it less than optimal, specifically in highly saturated volumes. Conversely, an adaptive signal is a more efficient solution as it has the built-in capacity to adapt to traffic changes without the restrictions that plague the actuated method. Reinforcement Learning (RL) (Sutton, Barto, et al., 1998) in TSC is generally employed to advance the category of the adaptive methods.

Derived from the natural learning processes observed in animals, RL allows a TSC system to learn and adapt to their environment. In a traffic environment, several components, such as pedestrians, drivers, vehicles and traffic signals may interact with each other. In TSC, traffic signals are the most common agents. Some TSC systems have a single agent in the RL environment, however it is common to have multiple agents work either cooperatively or competitively, in what is called Multi-Agent Reinforcement Learning (MARL). The benefit here is that agents work across a large environment while still having the precision of a single agent or close to it. The agents interact in a simulated traffic environment in different situations to learn the optimal way of interacting with the environment in a real-world setup. RL works based on a reward system that promotes long-term goals in an environment. The learning process is a feedback cycle of state, action and reward, where RL learns how an agent should map the states to actions to maximize a reward (this is discussed in detail in Section 1.1.). See Figure 1. The action often involves setting the phase duration. However, there are other action types, including setting the phase order, cycle time, offset, etc. We define these TSC-related terms in the following. A phase is defined as a period of time during which a set of non-conflicting traffic movements receive a green signal. A cycle is composed of several phases and cycle time is the time required to complete a full sequence of the phases. The proportion of the cycle that is green is called split. Moreover, in a coordinated system, offset is defined as the time that the green phase at an intersection begins after the beginning of green time of the reference signal. The main goal of TSC is to improve the environment or network performance (e.g. delay time, travel time, queue length, and speed) by controlling the actions of the agents. The main focus of this paper is on controlling the timing of traffic signal agents, although this control can be integrated with the control of actions of other types of agents, like vehicles in a connected vehicle environment.

Due to the rising popularity of RL in TSC recently, specifically in NTSC, we aim to thoroughly characterize the existing research in the area of urban traffic networks where RL is applied and to provide a complete account of what has already been explored. To this end, we exclude the research that has only proposed or tested for single isolated intersection control. Thus, we only concentrate on the application of RL in the network-level TSC, called as RL in NTSC or RL-NTSC for brevity, that tested the proposed methods in networks with two or more intersections.

It is worth noting, however, that there are several surveys and review papers that do cover this area. For instance, (Wei et al., 2019c; Yau et al., 2017; Mannion et al., 2016; Bazzan and Klügl, 2014; Bazzan, 2009; Abdulhai and Kattan, 2003; Wei et al., 2021) all present general reviews or surveys on TSC methods, compiling a list of the most recent methods and algorithms related to RL in TSC. Additionally, two very recent papers, (Gregurić et al., 2020; Haydari and Yilmaz, 2020), discuss the applications and opportunities regarding Deep Reinforcement Learning (Deep RL) in TSC, while a number of relevant studies that do not exclusively focus on RL in TSC, e.g. (Yuan Wang et al., 2019; Nguyen et al., 2018; Tahilyani et al., 2013; Z. Liu, 2007; Jácome et al., 2018; Yizhe Wang et al., 2018a; D. Zhao et al., 2011; Eom and B.-I. Kim, 2020) were found to exist. Nonetheless, to the best of our knowledge, there is no such systematic literature review aimed at examining the existing research of RL-NTSC. This research aims to: (i) collect all the existing relevant papers in the defined area and present a systematic, explicit, comprehensive, and reproducible review for identifying, evaluating, and synthesizing the existing body of the literature (based on the definition of a systematic literature review given by Fink (Fink, 2019) and Okoli et al. (Okoli and Schabram, 2010)), (ii) provide statistical and conceptual knowledge based on the qualitative and descriptive data analysis on data extracted from the included articles to investigate what has been done in the area, which methods and techniques were used (alone or as a core or combined method in integration with other methods), and which patterns, trends, and information can be extracted using data analysis techniques from the data reviewed, (iii) show which methods and which NTSC application domains have still room to be more elaborated in further research, (iv) identify the major first events in RL-NTSC to get familiar with how the research novelties and contributions are temporally located in

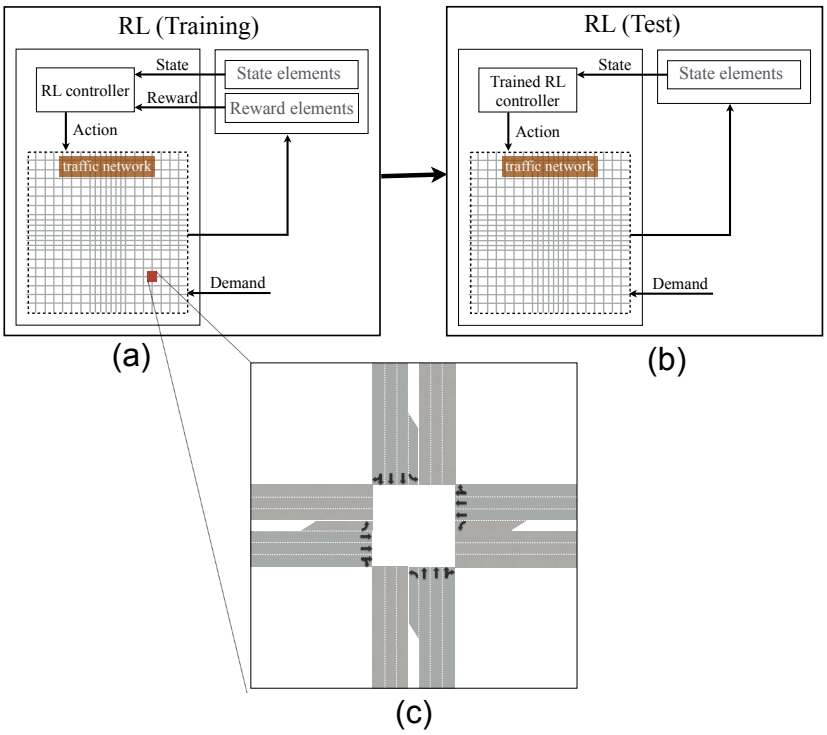


Figure 1: A general RL process: a) RL training, b) RL testing, and c) a sample intersection.

the course of research in the area, which is specifically well to find the very recent research problems, (v) explore the recent research problems and domains with their frequencies that helps identify potential future research, (vi) provide common future directions based on what the included papers recommended, and (vii) summarize existence evidence, (viii) identify and summarize areas for further research. For convenience, Tables 1 summarize the main acronyms in RL and NTSC domains.

1.1. Reinforcement Learning in Traffic Signal Control

A Markov Decision Process (MDP) is a mathematical framework well suited to optimize decision-making processes under uncertainty. An MDP is a four-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T} \rangle$, including, respectively, state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $\mathcal{R}$, and transition function $\mathcal{T}$. An MDP satisfies the Markov Property if the transition function, whether known or unknown, depends only upon the current state and the action taken, not on the sequence of events that preceded it.

If the reward and transition functions are known, the optimal policy can be found using dynamic programming (DP) methods via the recursive definitions of the value function. However, when the environmental dynamics are not known, i.e. reward and transition functions are unknown, the agent has to estimate the value of taking action in a state without using knowledge about the reward function and transition probabilities. In this situation, RL is suitable. RL can be model-based where the agent samples from the environment to estimate the reward and transition functions and find an optimal policy. Unlike model-based RL, in model-free RL algorithms the agent directly estimates the Q-function (Q-function is discussed later in this section) from experience while the reward and transition functions are unknown beforehand.

Table 1: Summary of abbreviations in RL and NTSC.

Acronym	Description
AC	Actor-Critic
AV	Autonomous Vehicle
ADP	Approximate Dynamic Programming
BPNN	Back Propagation Neural Networks
CAV	Connected and Autonomous Vehicles
CBR	Case-Based Reasoning
CMAC	Cerebellar Model Articulated Controller
CMOP	Correlated Multi-Objective Problem
CNN	Convolutional Neural Networks
CPT	Cumulative Prospect Theory
CQF	Connectionist Q-learning Framework
CRFs	Conditional Random Fields
CTM	Cell Transmission Model
DAE	Denoising Auto Encoder
DDQ	Double Deep Q-learning
Deep CNN	Deep Convolutional Neural Networks
Deep NN	deep Neural Networks
Deep RL	Deep Reinforcement Learning
DNN	Dueling Neural Networks
DP	Dynamic Programming
DQF	Decomposable Q-function
FFNN	Feed Forward Neural Networks
FNN	Fuzzy Neural Networks
GA	Genetic Algorithms
GAN	Graph Attentional Network
GCN	Graph Convolutional Network
GT	Game Theory
I2V	Infrastructure-to-Vehicle
IGMN	Incremental Gaussian Mixture Network
KNN	K-Nearest Neighbors
LA	Learning Automata
LC	Linear Combination
LCS	Learning Classifier System
LIME	Local Interpretable Model-Agnostic Explanations
Linear FA	Linear Function Approximation
LMT	Linear Model Tree
LR	Linear Regression
LSTM	Long Short-Term Memory
LTL	Linear Temporal Logic
MARL	Multi-Agent Reinforcement Learning
MDP	Markov Decision Process
ML	Machine Learning
MOPSO	Multi-Objective Particle Swarm Optimization
MP	Max Pressure
NFQI	Neural Fitted Q-iteration
NLP	Natural Language Processing
NN	Neural Network
Pro-CEP	Proactive Complex Event Processing
RBF	Radial Basis Functions
RL-CD	Reinforcement Learning with Context Detection
RM	Ramp Metering
RNN	Recurrent Neural Networks
SF	Smoothed Functional
SHAP	Shapley Additive exPlanation
SI	Signalized Intersection
SPSA	Simultaneous Perturbation Stochastic Approximation
SR	Sigmoid Regression
TD	Temporal Difference
TF	Triangular-shaped Functions
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VMS	Variable Message Signs
WNN	Wavelet Neural Networks

In model-free RL, an agent tries to learn the optimal way of interacting with an environment. RL learns how an agent should map the states to actions to maximize a numerical reward. At each time step (or decision point) k, based on a policy π that is intended to be optimized during the learning process towards reaching an optimal policy π^*, the agent takes action a_k from a set of possible actions $\mathcal{A}$ in response to the current state s_k from a set of possible states $\mathcal{S}$; i.e. $a_k = \pi(s_k)$. Simply put, a policy is a rule that the agent follows in selecting actions based on its current state. At the end of step k, the agent receives a reward r_k from the environment based on a reward function $\mathcal{R}$ where the elements of the reward can be collected through sensors. A sequence of state, action, and reward is a history of an agent saved in memory. At each time step, the RL agent tries to learn an optimal policy from its history of interactions with the environment that maximizes the discounted cumulative reward:

$$\mathcal{R}_k = \sum_{\phi=k}^{\infty} \gamma^{(\phi-k)} r_\phi = r_k + \gamma r_{k+1} + \gamma^2 r_{k+2} + ... = r_k + \gamma(r_{k+1} + \gamma r_{k+2} + ...) = r_k + \gamma \mathcal{R}_{k+1}, \tag{1}$$

where $\gamma \in [0,1]$ is a discount factor. The discount factor is associated with time horizons and is used to balance immediate and future rewards. It determines how much the RL agent cares about rewards in the distant future compared to those in the immediate future. If γ is 0, the agent only cares about the most immediate reward ($\mathcal{R}_k$=r_k). If γ is 1, the reward is not discounted and the distant future reward is considered ($\mathcal{R}_k$=r_k+r_{k+1}+r_{k+2}+...). As we set γ closer to 1, future rewards are given greater emphasis relative to the immediate reward. For more details, see (Sutton and Barto, 2018).

One of the most frequently used and successful RL methods in traffic signal control is Q-learning (Sutton and Barto, 2018), which was first investigated in 1989. Q-learning is a model-free RL. It is also an off-policy RL algorithm that uses a different policy for estimating Q-values than for action-selection. It updates the Q-values of the current state-action pair using the greedy policy to estimate the Q-value of the optimal policy of the next state-action pair. In other words, the optimal policy π^* is learned by estimating a second function, called Q-function, that specifies the value of an action (following a given policy π) given the current state. Q-function calculates the quality of a state-action combination. Assuming the agent continues to follow the optimal policy, the Q-value is defined as the expected discounted future reward of taking action a_k in state s_k:

$$Q^\pi(s_k, a_k) = \mathbb{E}_\pi[\mathcal{R}_k | s_k, a_k] = \mathbb{E}_\pi[\sum_{\phi=k}^{\infty} \gamma^{(\phi-k)} r_\phi | s_k, a_k]. \tag{2}$$

The Q-value is estimated by iterative Bellman updates:

$$Q^\pi(s_{k+1}, a_{k+1}) = Q^\pi(s_k, a_k) + \alpha(\Psi_k - Q^\pi(s_k, a_k)), \tag{3}$$

where $\alpha \in [0,1]$ is the learning rate that is set through experimentation, and $\Psi_k = r_k + \gamma \max_{a_{k+1}} Q^\pi(s_{k+1}, a_{k+1})$ is the target. Hence:

$$Q^\pi(s_{k+1}, a_{k+1}) = Q^\pi(s_k, a_k) + \alpha[r_k + \gamma \max_{a_{k+1}}[Q^\pi(s_{k+1}, a_{k+1})] - Q^\pi(s_k, a_k)], \tag{4}$$

Q-learning uses Q-table to decide which action to take. A simply put summary of Q-learning (Sutton and Barto, 1998) shows the algorithm to follow these steps:

1. s_k is the current state in which the agent resides.
2. The agent chooses action a_k from the available or acceptable actions.
3. In response, the agent receives a reward r_k for action a_k and the next state s_{k+1}. As already discussed, this state s_{k+1} is merely a projection or estimation of the next state, rather than a known value.
4. $Q(s_{k+1}, a_{k+1})$ is then updated using equation 4.
5. The entire process is repeated.

Table 2: Details of the search process

Journal/Conference	**#QGS**$_{(2017-2019)}$
IET Intelligent Transport Systems	2
Journal of Intelligent Transportation Systems	3
Engineering Applications of Artificial Intelligence	2
IEEE Transactions on Intelligent Transportation Systems	2
IEEE Transactions on Vehicular Technology	1
International Conference on Intelligent Transportation Systems (ITSC)	2
ACM International Conference on Information Knowledge Management (CIKM)	3
Total	15

Here, we should note that $\mathcal{R}_{k+1}$ in Equation 1 is the reward that is obtained at the next time step (after taking action), but it is not necessarily known upfront. That is why Q-learning is a model-free RL that does not know about the model of the environment, i.e. the reward and transition functions.

In the learning process, there is the exploration-exploitation dilemma. The agent tries to exploit based on what it already learned to achieve the reward, and at the same time, it also must explore possible actions for each state to find the one that has received the highest reward for exploiting it.

The rest of this paper is organized as follows: Section two will explore the methodology used to conduct this review, including the search strategy, selection criteria, and data extraction; Section three delves into the results and findings presented in section one; Section four discusses common future works and key findings; Section five covers threats to the validity of our work; and Section six wraps up with a look at future implications of the current paper.

2. Review Method

In this study, search strategy, inclusion and exclusion criteria, and data extraction are intertwined, and the selection criteria and data extraction are performed within the search and search evaluation steps; hence, we explore this as a whole rather than separately. It should be noted that to identify the relevant literature we conducted both manual and automated searches, and included all articles published up to the end of March 2020. We designed and implemented our review based on the guidelines provided by (Tricco et al., 2018).

2.1. Search strategy, selection criteria, and data extraction

To identify the most appropriate search terms and strings for the automated search, we used literature review, manual content analysis, and Natural Language Processing (NLP) in the steps that follow.

1: Based on knowledge in the area of the study, we identified seven venues, including five journals and two conferences as listed in Table 2, and manually searched for and reviewed the relevant published studies from 2017 to 2019 based on their title, abstract, and keywords. In very limited cases, we also examined the conclusion and searched for specific, relevant key terms to help with the inclusion decision. Fifteen pertinent articles were found during this first stage search. These papers were also used in the fourth step to form the Quasi-Gold Standard (QGS) (Abad et al., 2016) to evaluate the quality of the search strings.

2: We performed NLP, including language modelling and lexical association analysis (Abad et al., 2019), to extract the most frequently used terms in the 15 retrieved articles for the purpose of identifying search strings. Figure 2 shows the directed graph of common bi-grams formed from the QGS set, based on the frequency analysis on the pre-processed texts collected from the title, abstract, introduction, section/subsection headings, and conclusion of the 15 articles. Based on the results from the NLP, several inspections and investigations, and using various combinations of the most related search terms, the following single search string was chosen: $\langle$*reward AND action AND* (*traffic light OR traffic signal*) *AND reinforcement learning*$\rangle$.

3: Using the identified search string we queried Google Scholar, which served as our main search base, yielding 2,887 articles. To increase the reliability of our findings, the search was complemented by searching within Web of Science, IEEE Xplore, ACM Digital Library, Springer Link, and Science Direct databases after the data extraction.

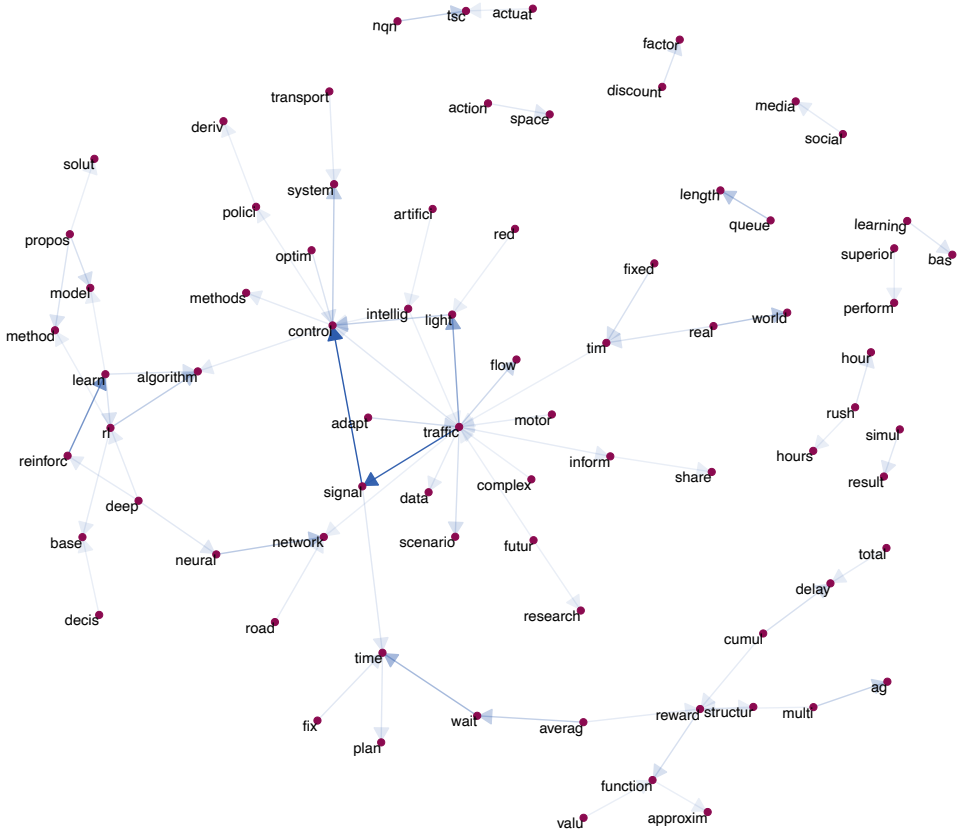


Figure 2: Directed graph of common bi-grams resulted from the NLP that is used to define the search strings.

This accelerated our search process by providing a view of those papers that were included and excluded, based on the defined criteria.

4: We formed the Quasi-Gold Standard. Our automated search retrieved all 15 articles, indicating the quasi-sensitivity of 100%.

5: We began the process of inclusion and exclusion by carefully defining the selection criteria. This iterative process continued even during data extraction, (i.e. the last step) to ensure the correctness of the included and excluded papers. Figure 3 depicts the flowchart of the article selection process in this paper. Publications that meet the following criteria were excluded: (1) those articles not related to RL-NTSC, (2) duplicate papers, (3) review and survey papers, (4) presentations, abstracts, extended abstracts, viewpoints, letters, reports, technical reports, projects, table of contents, any papers that have not been peer reviewed, theses and dissertations, books, and book chapters, (5) publications in any language other than English, (6) those papers whose better or updated version was found in another journal or publication and used, and (7) unavailability of the full-text of the paper on the internet.

Since the scope of our research is network-scale, it is henceforth referred to as NTSC (i.e. any scales involving two or more intersections), and another important exclusion criterion used was those papers that propose or test a proposed method in a single isolated intersection only.

We included those papers containing an RL method, whether it was the core or combined method applied in the context of NTSC. These were included whether or not they provided evaluation and simulation, and if they provided only a framework.

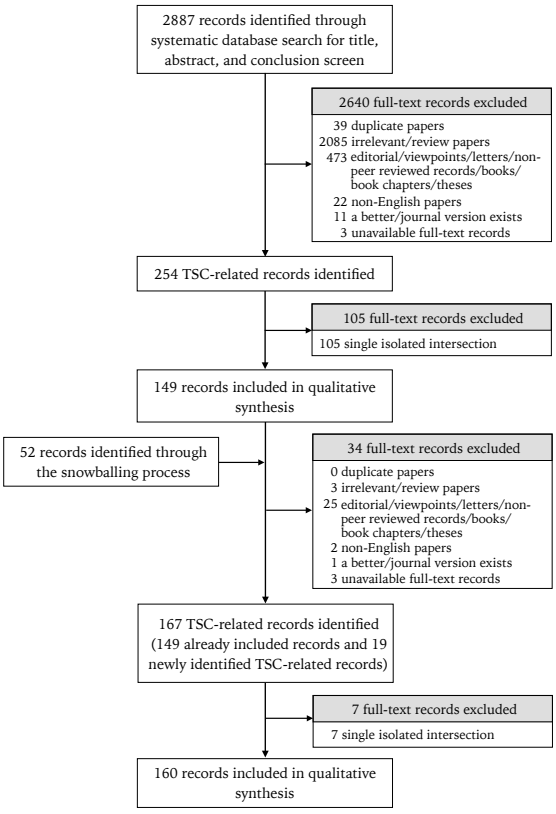


Figure 3: The flowchart of systematic literature review process.

This investigation is designed to cover the papers that apply RL in controlling traffic signals, and as such, if RL is used to control only vehicles, it is out of scope for our paper. In essence, as long as RL is being applied in NTSC, whether vehicle control is involved or not, the paper was included in the current study. Hence, the connected and autonomous vehicle (CAV) environment is covered in our paper as long as NTSC is involved. The CAV in the current study is inclusive of all environments of cases of CAV, including: vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-vehicle (I2V).

Despite the fact that below-mentioned research areas have traffic components (including: public transit, bikes, and pedestrians) in common with our research, they are out of scope since TSC is not their main focus as it is in the current study. The research areas include: ramp metering, freeway, traffic control (not TSC), public transit (where the focus is on bus control and TSC is excluded), route choice, routing systems, pedestrian routing, reactions of cyclists to speed advice, ride-sharing, best path selection, lane changing, autonomous intersection, traffic congestion detection, driver behaviour, traffic signal control simulation, simulator, simulation environment, online calibration, traffic assignment problems, couriers management in express systems, fleet management, toll plaza, traffic analytics, traffic control architecture based on fog computing paradigm where the focus is on fog paradigm (not TSC), image-based learning, image processing, and optimizing the sensor installation locations in a traffic network. We only found the topics, including ramp metering, public transit, emergency vehicles, and fog computing in NTSC, in which NTSC is included; thus we included these studies.

6: After applying the selection criteria and identifying the included papers, we used both, backward snowballing by screening the reference list of the included papers, and forward snowballing (Greenhalgh and Peacock, 2005) by

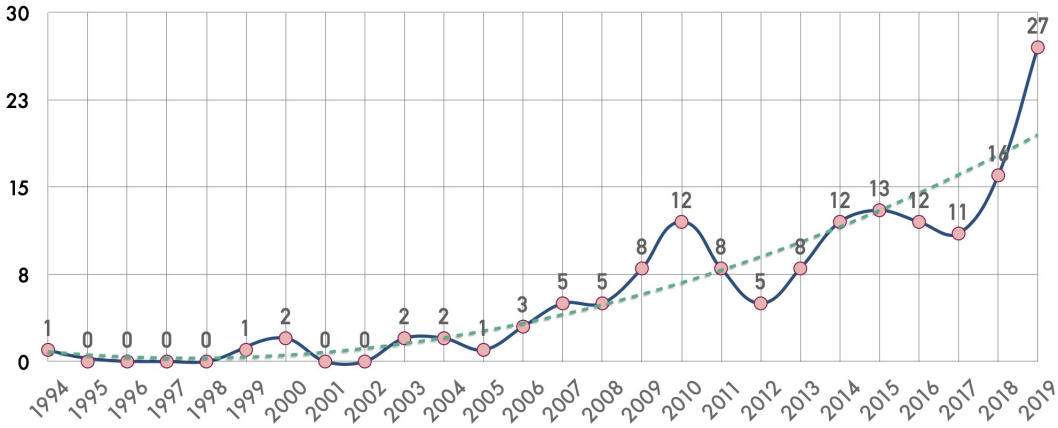


Figure 4: Distribution (solid line) and trend (dashed line) of number of studies over time. To make the comparison consistent, the 6 included papers published in 2020, until the end of March, are not included in this chart.

scanning the citation of the included papers to reduce the probability of missing some papers in our inclusion process; see Figure 3.

7: In addition to Google Scholar, five other electronic databases were searched, yielding no new articles. All articles found here had already been covered in the search using Google Scholar and the snowballing method.

8: From the included papers, data were extracted and analyzed.

In the following section, we provide an analysis of the data we extracted. In Table 3, we present an overall comparison along with details of some of the features extracted for all 160 papers included in our study. The papers that were reviewed are listed in the section entitled Included Articles. Other resources cited in this paper are listed in the reference section. It should also be noted that RL methods in Table 3 in the second column are classified into three main categories: model-free, model-based and RL. In this table, For model-free RL methods we present the specific model-free RL method that is used, such as Q-Learning, SARSA, etc. However, the model-based RL methods are shown as "Model-based RL". Also, in this table, by "RL" we mean to represent RL in general, where no specific RL method is mentioned in the paper. Moreover, there are articles that do not provide any specific RL method rather they mention the temporal difference (TD) update strategy. In this case, we report the method as TD as presented in these papers.

3. Results

As shown in Figure 4, the number of studies regarding RL-NTSC is increasing. This topic continues to gain momentum, and hence, importance as the world and specifically, urban populations increase and this is demonstrated in the large number of papers published recently. During the first 15 years, from 1994 to 2008, a total of 22 papers on the subject were published, compared to 27 papers in the year 2019 alone. These statistics further attest to the fact that an in-depth review like ours is important at this time as it provides future works a comprehensive view of the past 25 years.

The 160 included articles have been published in 104 different publication venues, including conferences (57%), journals (39%), and workshops and symposia (4%), see Figure 5. The first (Mikami and Kakazu, 1994) and second (Cao et al., 1999) papers in RL-NTSC were published in 1994 and 1999, respectively. These papers proposed model-free RL methods, while the first model-based RL method in RL-NTSC (M. A. Wiering, 2000) was proposed in 2000. Among the 160 included papers, 6 papers (M. Wiering et al., 2004; Kuyer et al., 2008; Houli et al., 2010; Khamis et al., 2012; Khamis and Gomaa, 2012, 2014) used or extended this model-based method, called as Wiering method. In addition to these Wiering method based research works, (Da Silva et al., 2006) proposed a model-based method that is based on a mechanism for creating, updating and selecting one among several partial models of the environment.

Table 3: Overall comparison of the included articles. [✓/✗]: yes for discrete methods and no for continuous methods.

Citation	RL Method	Combined Method	Approx. Method	Action Selection	Action Type	Discretization	Spillback or Sat.
Mikami and Kakazu, 1994	LA	Genetic Alg.	–	–	7	✗	Sat
Cao et al., 1999	LCS	Fuzzy	–	Credit Assignment	7	✗	Neither
Cao et al., 2000	LCS	–	–	Credit Assignment	7	✗	Sat
M. A. Wiering, 2000	Model-based RL	–	–	ϵ-greedy	7	✗	Sat
Camponogara and Kraus, 2003	Q-learning	–	–	ϵ-greedy	7	✗	Sat
Choy et al., 2003	RL	Fuzzy, Evol. Alg., NN	FNN	–	5	✓	Sat
M. Wiering et al., 2004	Model-based RL	–	–	–	5	✗	Neither
Dowling et al., 2004	RL	–	–	Greedy	7	–	No evaluation
Ling and Shalaby, 2005	Q-learning	–	IA	ϵ-greedy	6	✓	Neither
Da Silva et al., 2006	Model-based RL	–	–	–	3	✓	Sat
Srinivasan et al., 2006	RL	Fuzzy, Evol. Alg., NN	SPSA, FNN	–	7	✓	Sat
Oliveira et al., 2006	RL	–	–	–	3	✓	Sat
Richter et al., 2007	Actor-Critic	–	Linear and Non-linear FAs	Softmax	7	✓	Spillback
Su and Tham, 2007	Q-learning	SensorGrid	CMAC	Softmax	6	✗	Neither
Zhang et al., 2007	Policy-Gradient	CRF	–	–	7	✗	Sat
Bazzan et al., 2007	LA	–	–	–	Multiple	✗	Sat
Ritcher, 2007	Policy-Gradient	–	Linear FA	–	7	✓	Neither
Kuyer et al., 2008	Model-based RL	Max-plus	–	ϵ-greedy	7	✓	Sat
Salkham et al., 2008	Q-learning	–	–	Softmax	5	✓	Neither
T. Li et al., 2008b	ADP	–	FFNN	–	5	✗	Neither
S. Lu et al., 2008	Q-learning	Multiband	–	ϵ-greedy	6	✗	Sat
T. Li et al., 2008a	ADP	Neuro-fuzzy	–	–	3	✗	Sat
Xinhai and Lunhui, 2009	Q-learning	–	–	–	7	–	No evaluation
Cai et al., 2009	ADP	–	Linear FA	–	7	✗	Sat
Dusparic and Cahill, 2009a	W-learning	–	–	DWL	7	✗	Sat
X. Zhao et al., 2009	Q-learning	GT	BPNN	–	7	✗	Sat
Dusparic and Cahill, 2009c	W-learning	–	–	DWL	7	✓	Sat
W. Wu et al., 2009	Q-learning	–	–	–	Not clear	✗	Neither
Dusparic and Cahill, 2009b	W-learning	–	–	–	7	–	No evaluation
Prothmann et al., 2009	LCS	Evolutionary Alg.	–	–	5	✗	Sat
Medina et al., 2010	Q-learning	–	–	Multiple	7	✓	Spillback
Prashanth and Bhatnagar, 2010	Q-learning	–	Proposed Method	–	7	✓	Sat
Bazzan et al., 2010	Q-learning	–	–	Softmax	3	✓	Sat
Arel et al., 2010	Q-learning	–	FFNN		6	✗	Sat
Davarynejad et al., 2010	Q-learning	Fuzzy Granulation	–	–	7	–	No evaluation
Dai et al., 2010	ADP	–	NN	ϵ-greedy	7	✓	Sat
Balaji et al., 2010	Q-learning	–	–	Greedy	5	✓	Spillback
Natarajan et al., 2010	Inverse RL	–	–	Softmax	7	✓	Neither
Waskow and Bazzan, 2010	Q-learning	–	Tile Coding	ϵ-greedy	3	✓	Sat
El-Tantawy and Abdulhai, 2010	Q-learning	GT (Coalition)	–	–	7	–	No evaluation
Salkham and Cahill, 2010	Q-learning	–	–	ϵ-greedy	5	✓	Sat
Houli et al., 2010	Model-based RL	–	–	–	7	✗	Spillback
Prashanth and Bhatnagar, 2011	Multiple RL	–	Average Cost Alg.	–	6	✓	Sat
Song et al., 2011	Q-learning	–	–	–	7	✗	Neither

Citation	RL Method	Combined Method	Approx. Method	Act. Selection	Action Type	Quantization	Spill or Sat.
C.-g. Li et al., 2011	Q-learning	–	–	–	Not clear	✗	Sat
W. Lu et al., 2011	Actor-Critic	Swarm Optimization	–	–	7	✗	Neither
Natarajan et al., 2011	Inverse RL	–	–	–	7	✓	Neither
Heinen et al., 2011	Q-learning	–	IGMN	ϵ-greedy	3	✗	Sat
Abdoos et al., 2011	Q-learning	–	–	ϵ-greedy	4	✓	Sat
Dai et al., 2011	RL	–	FFNN	–	7	✗	Sat
Dusparic and Cahill, 2012	W-learning	–	–	Softmax	7	✗	Sat
Medina and Benekohal, 2012	Q-learning	Max-plus	–	–	7	✗	Spillback
Khamis et al., 2012	Model-based RL	–	–	–	7	✗	Sat
Khamis and Gomaa, 2012	Model-based RL	–	–	Softmax	7	✓	Sat
M. N. S. Jadhao and M. P. A. Kulkarni, 2012	Q-learning	–	–	–	7	✓	Spillback
L.-H. Xu et al., 2013	Q-learning	–	–	Softmax	3	✗	Sat
Moghadam and Mozayani, 2013	Q-learning	–	–	Random	3	✗	Neither
Y. K. Chin et al., n.d.	Q-learning	–	–	ϵ-greedy	5	✓	Sat
Nuli and Mathew, 2013	Actor-Critic	–	–	ϵ-greedy	5	✗	Sat
El-Tantawy et al., 2013	Q-learning	–	–	–	7	✗	Sat
Abdoos et al., 2013	Q-learning	–	–	ϵ-greedy	4	✓	Sat
Box and Waterson, 2013	Q-learning	–	NN	ϵ-greedy	7	✗	Sat
Pham et al., 2013	Q-learning	–	Tile Coding	ϵ-greedy	7	✗	Sat
Brys et al., 2014	Q-learning	–	Tile Coding	ϵ-greedy	7	✓	Neither
Prabuchandran et al., 2014	Q-learning	–	–	ϵ-greedy	4	✓	Sat
Chanloha et al., 2014	Q-learning	CTM	–	ϵ-greedy	7	✓	Sat
Fagan and Meier, 2014	Actor-Critic	–	–	ϵ-greedy	7	✗	Neither
Marsetič et al., 2014	Q-learning	–	–	ϵ-greedy	7	✓	Sat
Khamis and Gomaa, 2014	Model-based RL	–	–	Multiple	7	✗	Sat
El-Tantawy et al., 2014	Multiple RL	–	–	Comparison	Multiple	✓	Sat
Sadigh et al., 2014	TD	–	–	–	7	✓	Neither
Abdoos et al., 2014	Q-learning	–	Tile Coding	–	3	✓	Sat
W. Liu et al., 2014	Q-learning	–	Gradient-descent	–	Not clear	✗	Sat
N. S. Jadhao and A. S. Jadhao, 2014	Q-learning	Edge Detection Alg.	–	Random	7	✗	Sat
Teo et al., 2014	Q-learning	–	–	–	5	✓	Sat
Xiang and Z. Chen, 2015	Q-learning	Grey Theory	BPNN	Greedy	7	✗	Spillback
W. Xu et al., 2015	RL	–	–	–	Not clear	–	No evaluation
Yin et al., 2015	ADP	–	Gradient-descent	–	7	✗	Sat
Araghi et al., 2015	Q-learning	–	–	ϵ-greedy	7	✓	Sat
Ajorlou et al., 2015	Q-learning	CTM	–	ϵ-greedy	7	✗	Neither
Ozan et al., 2015	Q-learning	–	–	Random	6	✗	Sat
Zhu et al., 2015	RL	Junction Tree Alg.	–	ϵ-greedy	7	✓	Spillback
El-Tantawy et al., 2015	Q-learning	–	Linear Model Tree	–	7	✗	Sat
Prabuchandran et al., 2015	Q-learning	–	–	ϵ-greedy	2	✓	Sat
Abdelgawad et al., 2015	Q-learning	–	–	ϵ-softmax	7	✗	Spillback
Mashayekhi and List, 2015	Q-learning	Auction Theory	–	ϵ-greedy	7	✗	Sat
Abdoos et al., 2015	Q-learning	–	–	–	5	✗	Neither
Tahifa et al., 2015	Q-learning	Swarm Optimization	–	ϵ-greedy	3	✗	Neither
Van der Pol and Oliehoek, 2016	Q-learning	–	Deep NN	Greedy	7	✗	Neither

Citation	RL Method	Combined Method	Approx. Method	Act. Selection	Action Type	Quantization	Spill or Sat.
Dusparic et al., 2016	W-learning	–	–	Softmax	7	✗	Sat
Gaikwad et al., 2016	Q-learning	–	–	ϵ-greedy	3	✓	Sat
Prashanth and Ghavamzadeh, 2016	Actor-Critic	–	SPSA, SF	Softmax	7	✓	Neither
Iyer et al., 2016	Q-learning	Fuzzy	–	Greedy	7	✓	Neither
Yongheng Wang et al., 2016	Q-learning	Pro-CEP Model	DQF	–	Not clear	✓	Sat
Prashanth et al., 2016	RS RL	CPT Paradigm	SPSA	–	7	✗	Neither
Chu et al., 2016a	Q-learning	–	CNN	UCB	7	✗	Neither
Shen, 2016	Q-learning	Cellular Automata	WNN	–	7	–	Spillback
Chu et al., 2016b	Q-learning	–	Linear Q-function	ϵ-greedy	7	✓	Neither
Rosyadi et al., 2016	Q-learning	–	–	–	7	✓	Sat
Yin et al., 2016	Q-learning	–	Linear FA	–	7	✗	Sat
Y. Chen et al., 2017	Q-learning	–	–	–	7	✗	Sat
Kristensen and Ezeora, 2017	Q-learning	JADE	–	Greedy	7	✓	Neither
Darmoul et al., 2017	Q-learning	Immune Network	–	–	6	✓	Spillback
Jin and Ma, 2017	SARSA	–	–	–	7	✗	Neither
W. Liu et al., 2017	Q-learning	Dynamic Clustering	Gradient-descent, Tile Coding	–	7	✓	Neither
El Hatri and Boumhidi, 2017	Q-learning	–	–	–	6	✗	Incident
C. Lu et al., 2017	Q-learning	–	–	Softmax	7	✓	Sat
Chu and J. Wang, 2017	Q-learning	Max-sum	LR, SR	ϵ-greedy	7	✗	Sat
Y. Liu et al., 2017	Q-learning	–	–	ϵ-greedy	7	✗	Sat
Vidhate and P. Kulkarni, 2017	Q-learning	–	–	–	7	✓	Neither
Aslani et al., 2017	Actor-Critic	–	Tile Coding, RBF	ϵ-greedy	5	✓/✗	Sat
Al Islam et al., 2018	Q-learning	–	–	ϵ-greedy	7	✗	Sat
Daeichian and Haghani, 2018	Q-learning	Fuzzy, GT	–	–	3	✗	Sat
Wei et al., 2018	Q-learning	–	Phase Gate	ϵ-greedy	7	✗	Sat
Aslani et al., 2018a	Multiple RL	–	Tile Coding	ϵ-greedy	5	✓/✗	Spillback
Aziz et al., 2018	TD	–	–	ϵ-greedy	7	✓	Spillback
Aslani et al., 2018b	CR RL	–	Tile Coding, RBF, TF	–	5	✓/✗	Spillback
C. Li et al., 2018	Q-learning	–	DAE	ϵ-greedy	7	✗	Sat
Kao and C.-W. Wu, 2018	Q-learning	Self-Organizing Map	–	–	5	✗	Neither
X.-Y. Liu et al., 2018	Q-learning	–	NN	ϵ-greedy	5	✗	Neither
Torabi et al., 2018a	RL	Self-adaptive NTSC	–	–	Act 1	✓	Sat
Yizhe Wang et al., 2018b	Q-learning	Bayesian Optimization	–	–	6	✗	Sat
Shi and F. Chen, 2018	Q-learning	Transfer Learning	RNN	ϵ-greedy	7	✗	Sat
Jin and Ma, 2018	SARSA	–	KNN	ϵ-greedy	7	✗	Sat
Vinitsky et al., 2018	RL	–	NN	–	Multiple	✗	Sat
Lemos et al., 2018	Q-learning	–	–	ϵ-greedy	7	✓	Sat
Nishi et al., 2018	Q-learning	–	GCN	–	7	✓	Sat
Jin and Ma, 2019	SARSA	–	LC,KNN,CMAC,FFNN,CQF	ϵ-greedy	3	✗	Sat
Wei et al., 2019b	Q-learning	–	NN, GAN	Softmax	7	✗	Sat
Kitagawa et al., 2019	Q-learning	–	Deep NN	–	6	✗	Sat
Reda et al., 2019	Q-learning	–	–	–	7	✗	No evaluation
Shabestray and Abdulhai, 2019	Q-learning	–	CNN	ϵ-greedy	7	✓	Spillback
Q. Wu et al., 2019	RL	Fog Computing	–	–	4	✗	Neither
P. Chen et al., 2019	Q-learning	–	CNN	Greedy	7	✗	Sat

Citation	RL Method	Combined Method	Approx. Method	Act. Selection	Action Type	Quantization	Spill or Sat.
Rizzo et al., 2019a	Policy Gradient	–	Deep NN	Softmax	7	✗	Sat
Aslani et al., 2019	Actor-Critic	–	–	Comparison	7	✓	Neither
Huang et al., 2019	Q-learning	–	Deep NN	–	5	✓	Sat
R. Gao et al., 2019	Q-learning	Fuzzy, Edge Computing	–	ϵ-greedy	6	✓	Sat
T. Tan et al., 2019	Q-learning	–	–	ϵ-greedy	7	✗	Sat
Horsuwan and Aswakul, 2019	Q-learning	–	Deep NN	–	7	✗	Spillback
Chu et al., 2019	Actor-Critic	–	Deep NN	ϵ-greedy	7	✗	Sat
Wei et al., 2019a	Q-learning	–	Deep NN	–	7	✗	Spillback
N. Xu et al., 2019	Q-learning	Transfer Learning	LSTM	–	7	✗	Sat
Rizzo et al., 2019b	Policy-Gradient	–	Deep NN	Softmax	7	✗	Sat
Zheng et al., 2019a	Q-learning	FRAP Model Design	Deep NN	–	7	✗	Sat
Gong et al., 2019	Q-learning	–	Deep CNN	–	7	✗	Sat
Zhou et al., 2019	Q-learning	Edge Computing	Deep NN	ϵ-greedy	Act 1	✗	Sat
Bouderba and Moussa, 2019	Q-learning	Cellular Automata	–	–	Not clear	✓	Sat
Higuera et al., 2019	Q-learning	–	–	ϵ-greedy	7	✗	Neither
Ni and Cassidy, 2019	Actor-Critic	–	GCN	ϵ-greedy	Act 2	✗	Sat
Ge et al., 2019	Q-learning	–	Deep CNN	ϵ-greedy	7	✓	Sat
Shu et al., 2019	RL	–	–	–	7	✗	Sat
Gan et al., 2019	Q-learning	–	–	Softmax	7	✗	Sat
Yang et al., 2019	Actor-Critic	–	Deep NN,CNN	Softmax	7	✗	Sat
J. Lee et al., 2020	Q-learning	–	Deep CNN	–	2	✗	Sat
Genders and Razavi, 2020	Multiple RL	–	FFNN	Random	7	✗	Sat
M. Xu et al., 2020	Q-learning	–	RNN	ϵ-greedy	7	✗	Neither
Qu et al., 2020	RL	RMS-NE, CTM	–	Softmax	5	✓	Sat
Y. Zhao et al., 2020	RL	Junction Tree Alg.	–	–	7	✓	Sat
D. Kim and Jeong, 2020	Q-learning	–	DNN	ϵ-greedy	7	✗	Sat

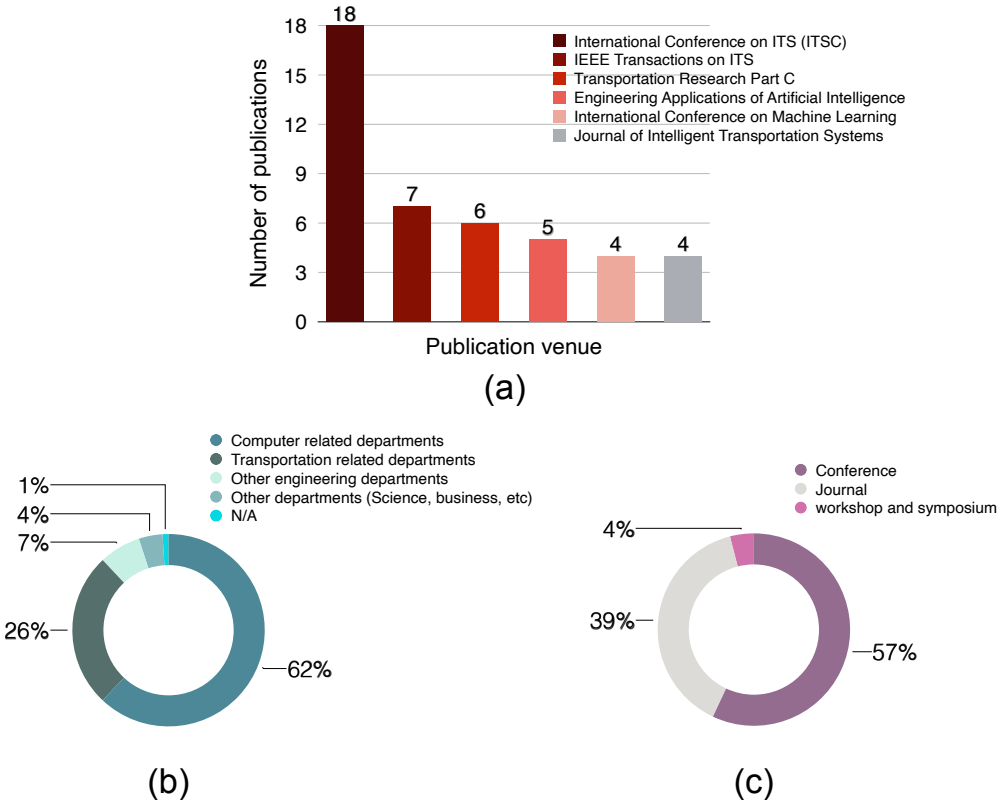


Figure 5: a) The frequency of publications in RL-NTSC in different publication venues, b) distribution of university department of authors in RL-NTSC, and c) distribution of publication venue types in RL-NTSC.

3.1. Countries, Departments and Affiliations

Figure 6 exhibits our data of the distribution of research papers among each country. 160 papers from a total of 30 different countries were included in this review, with most of the research papers coming from 7 countries: China, USA, India, Iran, Ireland, Canada, and Brazil. These papers represent 67% of our pool of papers, strongly suggesting that as of 2020, these countries are the global leaders in the research of RL-NTSC. An interesting fact to note is that all seven of the countries mentioned have a traffic index[1] above 140.45 according to (Index, 2014), with Iran having the highest traffic index of the group, at 216.09. The motor data company, INRIX (Inrix, 2020), finds Ireland's capital city of Dublin to be the 7th worst city in the world in terms of hours lost due to traffic congestion based on 2019 data (154 hours). And, according to the Central Intelligence Agency (CIA) World Factbook (Central Intelligence Agency, 2020), the USA, China, India, and Brazil rank in the top four countries with the longest road network, respectively, with Canada at eighth.

This study also took into consideration the department and affiliations of authors, and identified four groups of departments, as follows: (1) computer related departments, such as computer engineering/science, information technology, and electrical engineering, (2) transportation related departments, including civil and transportation

[1]Metric that is a composite index of time consumed in traffic due to job commute, estimation of time consumption dissatisfaction, CO2 consumption estimation in traffic and overall inefficiencies in the traffic system.

Figure 6: Left) the distribution of studies based on country of the authors, and right) the number of authors. The number of authors of the papers published on RL in NTSC ranges from 1 to 10.

engineering departments, (3) other engineering departments, such as industrial, mechanical, material, and geomatics, and (4) other departments, like science, business, management, astronautics, and English. It was found that 62% of the authors were from computer related departments (group 1), and 26% were from transportation related departments (group 2), while researchers from other departments (groups 3 and 4) accounted for 12% of the total.

Notably, authors from these four groups of departments had very low research collaboration with each other, and had conducted research in only 20 out of 160 articles, generally. The computer related and transportation related departments collaborated in only 11 articles, while they contributed independently in 98 and 36 papers, respectively.

This investigation also uncovered that academia and research institutes have low research collaborations with industry and government, with 11 instances of collaboration between academia and industry and only one appearance of government in research papers that collaborated with academia in a paper in 2018. Suffice it to say that research exploring the potential benefits of the collaborations between these three parts, i.e. academia, industry, and government is needed, as the findings in the literature (Anderson and Odei, 2018) suggest that increased collaboration in these domains may boost the efficiency of the proposed methods in real-time, real-world applications.

3.2. Method identification and analysis

In this section, we elaborate on the proposed methods from different angles. We start with the categorization of the proposed methods in terms of the method and environment attributes, respectively, in Figures 7 and 8.

3.2.1. Methods' contribution and combination

The proposed methods in the included articles consist of (M1) the applied RL methods in NTSC (154 papers) or (M2) theoretical developments in RL with feasibility or applicability assessment of the proposed methods in different contexts, one of which is the NTSC environment (6 papers).

To get familiar with the variations of application of RL in solving different research problems in NTSC and the way that RL is applied, we identified two main groups: (M1.1) RL is used alone or in combination with methods

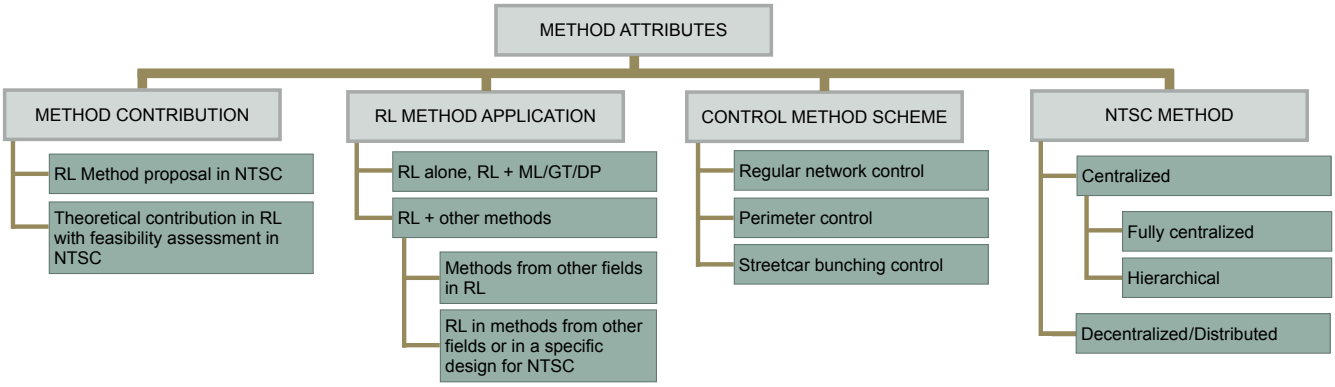


Figure 7: Method attributes in RL-NTSC.

from the same field (i.e. machine learning (ML), game theory (GT), and DP methods) in NTSC (134 papers), and (M1.2) RL is used in combination with other methods from other fields in NTSC (20 papers).

⟨M1.1⟩: The vast majority of the studies (134) are classified in this category where RL is the only applied method, or is used in combination with other ML, GT, and DP methods. Different innovative designs were proposed to tackle the problems of NTSC. The NTSC problem has a continuous state space, infinite horizon, and is only partially observable and difficult to model. In this context, most of the papers try to improve the performance (e.g. (Moghadam and Mozayani, 2013)), dimensionality (e.g. by means of function approximation (Waskow and Bazzan, 2010; Prashanth and Bhatnagar, 2010)), complexity (e.g. by organizing agents in groups of limited size (Bazzan et al., 2010), and using holonic RL methods (Abdoos et al., 2013)), scalability (e.g. (Nuli and Mathew, 2013)), stability (e.g. (Aslani et al., 2018b)), speed of optimization (e.g. in Transfer Learning models (N. Xu et al., 2019)), state and/or action space manageability or generalizability (e.g. (Araghi et al., 2015; Gaikwad et al., 2016)), and centrality problem in applying RL in NTSC. Some studies invest in solving convergence and oscillation problems that commonly appear in the multi-agent context. For instance, (Reda et al., 2019) proposed a model based on Double Deep Q-learning (DDQ), with Experience Replay and cooperation between agents. They used NN to reduce the correlation between agents and improve performance. Improving the accuracy, optimum functioning, and efficiency of the results is also of high interest to the researchers, through hierarchical methods (Yizhe Wang et al., 2018b; T. Tan et al., 2019), for example. There are several papers where the main focus of research is on studying the coordination between agents (e.g. (Higuera et al., 2019)), and the integrated network, specifically signalized intersections and ramp metering (e.g. (El-Tantawy and Abdulhai, 2010; El-Tantawy et al., 2013, 2015)). Famous methods proposed in traffic theory context, such as CTM (Chanloha et al., 2014; Ajorlou et al., 2015; Qu et al., 2020), Max-plus (Kuyer et al., 2008; Medina and Benekohal, 2012), and Max Pressure (MP) or back pressure (brought in NTSC from the communications networks theory) (Wei et al., 2019a), are also applied in some research, and in other studies multiple traffic optimization goals are simultaneously optimized (multi-policy RL), e.g. (Dusparic et al., 2016).

Considering both vehicular and pedestrian traffic in the network-scale is a recent application of RL, where a distributed multi-agent RL method is proposed by (Y. Liu et al., 2017) for the first time in 2017. By the increasing use of deep RL methods, the number of papers that focused on improving deep RL models in NTSC has increased since 2018 (e.g. (Wei et al., 2018; C. Li et al., 2018; X.-Y. Liu et al., 2018). The co-learning problem of both classes of learning agents, traffic signals and drivers, with different goals (minimizing individual travel times vs minimizing the queues locally), different nature (driver agents learn in episodes that are asynchronous, while traffic light agents learn continuously (non-episodic)), and the nontrivial task of microscopic modelling and simulation (whose actions are highly coupled) is another area of research that was addressed (Lemos et al., 2018). In addition to these lines of research in this category, analyzing what specifically RL does differently (i.e. analysis of the learned policies) than

other TSC methods is another research effort that is motivated and conducted by (Genders and Razavi, 2020).

⟨M1.2⟩: In category M1.2, the articles applied methods from other fields (rather than ML, GT, and DP) in RL or RL framework (M1.2.1), or used RL in the methods from other fields or in a specific design/model/framework for NTSC context (M1.2.2). Out of 20 articles in category M1.2, there are 8 articles that either applied RL in optimization problems (e.g. Swarm optimization) or an optimization algorithm is used in RL methods/frameworks in the context of NTSC.

⟨M1.2.1⟩: To cope with non-stationary environments, (Da Silva et al., 2006) proposed, formalized and showed the efficiency of a method called the RL-CD, or Reinforcement Learning with Context Detection, which performs well in non-stationary environments, and better than classic RL algorithms (Q-learning and Prioritized Sweeping). In a similar work, (Oliveira et al., 2006) assessed the feasibility of applying RL-CD approach in a more realistic scenario, implemented by means of a microscopic traffic simulator. (Zhang et al., 2007) showed how to use Conditional Random Fields (CRFs) to model control processes, where CRFs model joint actions in a decentralized Markov decision process and define how agents can communicate with each other to choose the best joint action. The CRF model clearly outperformed the independent agents approach. (Khamis et al., 2012) enhanced the single-objective controller by developing a multi-agent NTSC system based on a multi-objective sequential decision-making framework using Bayesian interpretation and some innovative reward design. (Zhu et al., 2015) proposed a Junction Tree Algorithm (JTA) based approach to obtain an exact inference of the best joint actions in traffic signal coordination that outperformed independent learning (Q-learning), real-time adaptive learning, and fixed timing plans. To predict future system states and avoid unwanted states, (Yongheng Wang et al., 2016) applied Proactive Complex Event Processing (Pro-CEP) method (in processing proactive traffic congestion control) that uses RL to find the optimal joint policy. This method works well when used to control congestion. To develop learning and adaptation mechanisms to deal with disturbances, (Darmoul et al., 2017) proposed a distributed TSC system based on hybridization between Case-Based Reasoning (CBR) and an adaptation of the reinforcement principle within artificial immune networks (a mechanism inspired by biological immunity). The method controls interrupted flow at signalized intersections. (Wei et al., 2019b) studied how the attention mechanism helps cooperation (to minimize the average queue length) via using graph attentional networks to facilitate communication, incorporating the temporal and spatial influences of neighbouring intersections to the target intersection, and building up index-free modelling of neighbouring intersections.

⟨M1.2.2⟩: (Su and Tham, 2007) integrated sensor networks and grid computing and the usage of web services to implement this integration. They used Q-learning algorithms in distributed Stargates to NTSC. Stargates is a computer with sensor signal processing capabilities. (Abdoos et al., 2015) modelled a holonic multi-agent system and proposed a holonic RL multi-agent system method that improves the performance of individual Q-learning in NTSC. (Mashayekhi and List, 2015) showed the applicability and efficacy of using auction theory combined with an RL in a multi-agent system. In low traffic volume the proposed method outperforms actuated and pre-timed control strategies, but in heavy volume the pre-timed control strategy outperforms the proposed method. (Iyer et al., 2016) designed a distributed multi-agent method with coordination between agents through the communication of decision data. In this paper, the effectiveness of integrating fuzzy logic controller (to deal with continuous states and actions) and Q-learning (for learning during the process) is studied.

With regards to the optimization-related articles in category M1.2, all of the proposed methods in this category are designed to handle the growing complexity and curse of dimensionality and to increase the speed of TSC by using RL in the optimization problems or applying the optimization algorithms in an RL method or framework. To achieve cooperation in the long term, as stated earlier, (Mikami and Kakazu, 1994) combined RL of a local agent with global optimization through Genetic Algorithms (GA) by which the RL parameters are modified. (Cao et al., 1999) used a classifier system with a fuzzy rule representation, with both evolutionary and RL methods to provide a faster method than hierarchical control. (Cao et al., 2000) incorporated Learning Classifier System (LCS) and TCP/IP (Transmission Control Protocol/Internet Protocol) based communication server into a distributed learning control strategy to increase the speed of control. (Prothmann et al., 2009) managed the complexity by using an organic approach to NTSC and proved the feasibility of the proposed approach. (W. Lu et al., 2011) proposed

a multi-agent NTSC by using Swarm Intelligence and Neuro-Fuzzy RL to combine the better attributes of both with improving the learning speed and performance. (W. Liu et al., 2014) optimized the TSC for V2I networks by proposing a cooperative distributed Q-learning algorithm with a fast gradient-descent function approximation. (Ozan et al., 2015) applied an RL method in the optimization problem to reach good solutions for NTSC. The results were better than the genetic algorithm and hill-climbing methods in low demand but could not outperform them in medium and high demands. (Tahifa et al., 2015) showed that Swarm Q-learning performs better than standard Q-learning in increasing the speed of TSC. To alleviate traffic congestion and limit the effects of incidents on traffic flow, (El Hatri and Boumhidi, 2017) proposed a Q-learning based traffic management model, which simultaneously optimizes vehicle re-routing and TSC based on the Multi-Objective Particle Swarm Optimization (MOPSO) method.

⟨M2⟩: To provide a simple and efficient method to implement, (Natarajan et al., 2011) put into operation a functional gradient boosting approach to imitation learning in relational domains. The proposed approach outperforms both learning a single relational regression tree and performing a propositional functional gradient to represent the policy in all domains. To provide a solution to multi-objective problems with correlated objectives rather than typical multi-objective problems, (Brys et al., 2014) proposed an RL-based method combining multiple correlated rewards and shaping signals by measuring confidence (i.e. combining the feedback from all objectives, instead of only looking at a single one), called adaptive objective selection. They formally defined a new class of multi-objective problems called correlated multi-objective problems (CMOP), whose set of solutions being optimal for at least one objective is so restricted that the decision-maker is least concerned about which of these is found, and more so about how fast one is found, or how well one is approximated. (Sadigh et al., 2014) proposed a method for synthesizing a control policy for an MDP such that traces of the MDP satisfy a control objective expressed as a linear temporal logic (LTL) formula through using an RL algorithm that finds the policy optimizing the expected utility of every state in the Rabin-weighted product MDP. They prove that the method is guaranteed to find a controller that satisfies the LTL property with probability one if such a policy exists, and they suggest empirically with a case study in traffic control that their method produces reasonable control strategies even when the LTL property cannot be satisfied with probability one. (Prashanth and Ghavamzadeh, 2016) optimized variance-related risk measures in rewards and demonstrated its usefulness in an NTSC application. The risk-sensitive algorithms result in lower variance but higher long-term cost compared to their risk-neutral counterparts. (Prashanth et al., 2016) illustrated the usefulness of modeling human decisions by Cumulative Prospect Theory (CPT) paradigm in RL and suggested that CPT-based criteria is useful in a NTSC application. (Gan et al., 2019) proposed a dynamic correlation matrix based MARL approach where the meta-parameters are evolved using an evolutionary algorithm in a distributed manner. This was done to provide meaningful theoretical verification by using both agent-level implementation and system-level convergence verification. Agents using the proposed learning algorithm reach optimal behaviours faster than other canonical learning techniques.

In the following, we introduce the types of RL methods used in the articles we included in this paper. Furthermore, we extracted data about other methods that an RL method has been integrated with to provide a solution, whether as a core or combined method.

3.2.2. RL methods

Q-learning (Watkins and Dayan, 1992) ranks first on the list of used RL methods in 96 (60%) studies. 13 papers do not provide any specific RL method, especially when the goal of the paper is the proposal of a framework or when an RL concept is used. In this case, we used "RL" in Table 3. The remaining papers include methods that are listed (along with their frequency) in the first column of Table 4.

Furthermore, RL is the core method in 149 (93%) of the studies. In 11 studies, other methods were employed as the core method in which RL is used as a combined method, see the second column of Table 4. There are also other non-RL methods that were used in combination with the RL methods or frameworks, see the third column of Table 4. Below is a synopsis of the RL methods.

- *Q-learning and SARSA*: Both Q-learning and SARSA (State, Action, Reward, State, Action) methods are critic-only where they use Q-tables to decide which action to take. The biggest difference between the two is

Table 4: RL, core non-RL, and combined non-RL methods.

RL method	Core non-RL method	Combined non-RL method
Q-learning (96)	Auction theory	JTA
No specific RL (13)	Fuzzy	CPT paradigm
Actor-critic (10)	Fog Computing	Self-Organizing Map
Model-based RL (8)	Pro-CEP	Model-based and Bayesian optimization algorithms
ADP (6)	Self-adaptive NTSC system	Transfer Learning
W-learning (5)	Cellular Automaton	FRAP model design
Multiple RL (4)	Immune Network algorithm	CTM
Policy-Gradient (4)	NN	RMS-NE
LCS (3)	Multiband	Max-plus
SARSA (3)	SensorGrid	CRF
TD (2)		GA
LA (2)		Grey theory model
Inverse RL (2)		Qualitative method
Continuous Residual RL (1)		Evolutionary Alg.
Risk-sensitive RL (1)		Dynamic Clustering Alg.
		Neuro-Fuzzy
		JADE
		Coalition
		Fuzzy granulation
		Edge computing framework
		Max-sum message passing Alg.
		Swarm optimization

that Q-learning is an off-policy RL method, while SARSA is an on-policy method. Q-learning and SARSA Q-table update equations are respectively presented below, where the policy is TD.

- *Critic-only, Actor-only, and Actor-Critic*: The critic uses the calculated Q-values (or function values) to choose its action while the actor uses the policy to decide. Actor-only methods work to improve the policy. Actor-critic being a combination of both, allows both calculations of the Q-values and the policy to choose appropriate action. The papers that used an actor-critic method are identified in Table 5.

- *W-learning:* W-learning is a multi-policy self-organising action-selection technique proposed in (Humphrys, 1996) that builds on Q-learning. In W-learning, there is a competition among selfish Q-learners where agents learn Q-values for state-action pairs for each policy and W-values for each of the states of each of their policies to explore what happens if the nominated action is not followed.

- *Approximate Dynamic Programming (ADP):* The computational complexity of DP algorithms due to excessive system state and their need for an exact algorithm and true value function makes the algorithm impractical in solving large-scale TSC problems. ADP aims not to fall into the predicament of computational complexity by replacing the true value function of the DP with an approximation function. In other words, it is similar to the model-based RL with function approximation. The research papers that used ADP method are identified in Table 5.

- *Policy-Gradient:* The policy gradient method does not need to estimate the state or action value functions. It learns parameterized policy functions directly by searching policy space to maximize a measure based on the accumulated reward. In this way, it averts the convergence problems of estimating value functions.

- *Inverse RL*: In Inverse RL (Ng, Russell, et al., 2000), the reward function of an agent (that the agent tries to optimize) is learned and determined by observing the agent's behaviour over time, the environment model, and the environment measurements. This approach is akin to learning from an expert and is helpful in the domains where the reward may not be easily accessible, like TSC (Natarajan et al., 2010). This method has its

origins in Imitation learning (also called as apprenticeship learning, learning by observation, or learning from demonstrations). It is comparable to supervised learning, with the key difference being that the examples are not i.i.d, but instead, follow a meaningful trajectory (Bagnell et al., 2006).

- *Learning Classifier Systems:* A learning classifier system (LCS) (Butz et al., 2005) is a rule-based RL system in which each rule (or classifier) is composed of a condition, an action, and a reward (or evaluation). LCS combines an evolutionary process (e.g. a genetic algorithm), with a learning process (e.g. RL), wherein a rule is constructed as {IF 'condition' THEN 'action'}. A genetic algorithm tries to improve condition-action rule space by generating new classifiers from current strong classifiers and removing the weak ones. RL is responsible for selecting the action with the best-rewarded response or evaluation to be executed.

- *Learning Automata:* The action selection in learning automata is performed based on the last selected action and the received reward. A learning automata method forms from a vector of probabilities over the set of actions, which are updated (i.e. increase or remain the same) based on the reward.

- *Model-based and Model-Free*: Model-based methods provide the agent with part or all of a model in which the agent must work. In model-free methods the agent develops its own model in which to work and has fewer restrictions. Essentially, in model-based methods the transition and reward functions are assumed to be available to construct a model, unlike in model-free methods where the agents do not need to have access to information regarding how the environment works. The papers that used model-based methods are identified in Table 5.

3.2.3. Control method scheme

By analyzing the proposed methods in the included articles, we identified three levels of control, including (i) regular network control, which covers the general idea of controlling all traffic signals in a regular network, (ii) perimeter control, which improves the performance of the entire network by controlling a part of traffic signals that located on the boundary of a region in the network, and (iii) streetcar [2] bunching control, which mitigates the effect of streetcar bunching along transit routes by controlling successive signalized intersections.

The regular network control has been the focus of 158 out of 160 included articles. On the other hand, although there are numerous research works in the domain of perimeter or cordon control, the application of RL for perimeter control is studied, for the first time, in 2019. (Ni and Cassidy, 2019) explored how RL can be used to re-time traffic signals on the perimeter by developing an RL based controller with NN architectures that controls perimeter with spatially-varying metering rates. With regard to the third level of control, (Ling and Shalaby, 2005) conducted research to mitigate the effects of streetcar bunching along transit routes through automating streetcar bunching control by means of multiple RL agents that act on a series of successive signalized intersections. The proposed method was able to effectively split up a streetcar bunch and prevent it from forming again.

3.2.4. NTSC method: Centralization (centralized, hierarchical, and decentralized/distributed methods)

When tackling multi-agent problems, there is a spectrum from centralized to decentralized decision making. At large scale implementation, multiple agents tackle the task given while communicating with each other. This usually results in quicker optimization as each agent learns from its neighbours as well as from itself (Tahifa et al., 2015). With multiple agents, the collected data and actions can be stored centrally in a location that all agents can access to function as one agent. In this setup, the central agent often makes all decisions for the system, and this can slow down the learning process while coordinating the unit (OroojlooyJadid and Hajinezhad, 2019). Although DNNs helps enhance the scalability of RL, training a centralized RL agent is still infeasible for large-scale NTSC. Conversely, a

[2] street railway or tram

Table 5: Identification of the included papers in terms of method identification, code, simulation, and evaluation.

Category	citation
Method Identification	
Actor-critic	Prashanth and Ghavamzadeh, 2016; Aslani et al., 2018a, 2017; Genders and Razavi, 2020; Chu et al., 2019; Yang et al., 2019; Ni and Cassidy, 2019; Fagan and Meier, 2014; Nuli and Mathew, 2013; W. Lu et al., 2011; Zhang et al., 2007; Richter et al., 2007; Aslani et al., 2019; Dai et al., 2010
Deep Learning	Wei et al., 2018; C. Li et al., 2018; X.-Y. Liu et al., 2018; Zhou et al., 2019; M. Xu et al., 2020; Shi and F. Chen, 2018; J. Lee et al., 2020; T. Tan et al., 2019; Horsuwan and Aswakul, 2019; Chu et al., 2019; Wei et al., 2019a; N. Xu et al., 2019; Rizzo et al., 2019b; Zheng et al., 2019a; Gong et al., 2019; D. Kim and Jeong, 2020; Yang et al., 2019; Ni and Cassidy, 2019; Ge et al., 2019; Shabestray and Abdulhai, 2019; Kitagawa et al., 2019; Van der Pol and Oliehoek, 2016; Rizzo et al., 2019a; P. Chen et al., 2019; Chu et al., 2016a; Vinitsky et al., 2018; Reda et al., 2019; Huang et al., 2019
Model-based methods	Khamis and Gomaa, 2014; Kuyer et al., 2008; M. Wiering et al., 2004; Houli et al., 2010; Khamis et al., 2012; M. A. Wiering, 2000; Khamis and Gomaa, 2012; Da Silva et al., 2006
RL and GT	Qu et al., 2020; L.-H. Xu et al., 2013; X. Zhao et al., 2009; Daeichian and Haghani, 2018; El-Tantawy et al., 2013; El-Tantawy and Abdulhai, 2010; Xinhai and Lunhui, 2009
ADP	Fagan and Meier, 2014; T. Li et al., 2008b; Yin et al., 2015; Dai et al., 2010; Reda et al., 2019
Code, Simulation, and Evaluation	
Code is available	Wei et al., 2018; Prashanth et al., 2016; Zhou et al., 2019; Genders and Razavi, 2020; Chu et al., 2019; Wei et al., 2019a; Zheng et al., 2019a; Wei et al., 2019b; Brys et al., 2014; Vinitsky et al., 2018
No Simulation	W. Xu et al., 2015; Reda et al., 2019; Dowling et al., 2004; El-Tantawy and Abdulhai, 2010; Davarynejad et al., 2010; Xinhai and Lunhui, 2009
No Evaluation/Self-comparison	W. Xu et al., 2015; Reda et al., 2019; Dowling et al., 2004; El-Tantawy and Abdulhai, 2010; Davarynejad et al., 2010; Xinhai and Lunhui, 2009; Dusparic and Cahill, 2009b; Prashanth et al., 2016; X.-Y. Liu et al., 2018; Prothmann et al., 2009

distributed approach can be used to store the local information around multiple agents, allowing each one to make its own decision while still communicating with its neighbours (Hüttenrauch et al., 2019). In this approach, all agents would be considered "equal" (Baldazo et al., 2019). If the local agent does not communicate with other neighbor agents the system is called a decentralized system. Another setup that uses a combination of both centralized and decentralized/distributed methods is a hierarchical system, which can be categorized as a centralized method since it involves centralization. It forms a hierarchy of sorts, where the lower agents may have limited to no ability to enact upon the environment without permission from the "leader". A hierarchical control allows agents to perform micro-actions between tasks to improve the finesse of the agents.

Most (65%) of the proposed methods in NTSC are designed in a decentralized way. There are 7 papers proposing holonic or hierarchical methods while the rest are centralized methods that might be rarely applicable in real-world scenarios in a real-time process.

3.2.5. RL methods' components and types

State, action, and reward. There are normally three main components in RL: state, action, and reward. There are several elements that can define state and action in various papers. The elements of the state can be similar or different from those of the rewards. Based on our collected data of 160 studies, we identified 35 distinct elements of state and 30 of reward. The top 5 frequently used elements in state are queue size with 73 occurrences (38%), phase

Table 6: Categorization of types of traffic signal based actions in the RL methods. The numbers in parentheses indicate the number of publications in which an action type is used.

class	cycle length	phase duration	phases order	frequency	cycle-/phase-based (#)
2	fixed	fixed	variable	(2)	cycle-based (48)
3	fixed	variable	fixed	(13)	
4	fixed	variable	variable	(4)	
5	variable	variable	fixed	(18)	
6	variable	variable	variable	(11)	
7	-	variable	-	(100)	phase-based (100)

other action types		
class	**description**	**frequency**
Act1	set the value of a threshold metric for each traffic signal	(2)
Act2	set a link specific metering rate in the perimeter control	(1)
Multiple Actions	including: (**Act3**) set the value of a threshold metric for each traffic signal, (**Act4**) set a link specific metering rate in the perimeter control, and (**Act5**) select a route as a driver's action	(3)

state (11%), number of vehicles (10%), the position of the vehicles (6%), and speed (6%). In reward, the top 5 are queue size with 71 occurrences (30%), delay (13%), waiting time (9%), the number of vehicles (6%), and number of vehicles passed the intersection (or generally throughput) (4%). The elements of the state in 16 (5%) research papers and those of reward in 18 (7%) papers are not available. The list of all the elements of the state (38 unique elements) and reward (39 unique elements) found in the papers are available in Appendix A and Appendix B. This might help the new researchers get some idea in defining these components. These two appendices also depict the most frequently used elements in states and reward, respectively.

In addition to state and reward, action needs to be defined. In the majority of the research papers action is defined as a traffic signal or phase switch. However, there are some that will use a different definition than the signal switch. These actions include (Act 1) set the value of a threshold metric for each traffic signal, (Act 2) a link specific metering rate in the perimeter (or cordon) control, (Act 3) select a route as a driver's action, (Act 4) set the acceleration of the vehicles, and (Act 5) set the maximum speed of the vehicles. The last three options are used in a mixed environment where the vehicles are also considered as well as traffic signals. When two or three types of actions were used, it was reported as "Multiple" in Table 3.

The actions directly related to control traffic signals are categorized into two main groups (phase-based and cycle-based), and seven classes where each class is defined based on cycle length, phase duration, and phase order. Each of these three elements can be fixed or variable. See Table 6. The decision point in cycle-based methods is the end of the cycle where cycle length, phase duration, or phase order are determined. In phase-based methods, the decision is made at the end of a phase, which includes phase duration determination and phase selection. In this case, phase duration can be set as fixed for the entire phase or can be allowed to be extended at the end of the phase. Note that in phase-based methods, the cycle and phase order are not applicable. In Table 6, we removed class 1 in which all the three elements (cycle length, phase duration, and phase order) are fixed. This is not applicable in RL design and is counter-intuitive to optimize.

62% of the papers proposed phase-based methods (i.e. class 7), while 32% constructed their methods on cycle-based methods. Only 2 papers focused on using RL, assuming that the phase duration is fixed. Moreover, only 20 (12%) papers worked on applying RL in a fixed cycle-length setup. There are also 3 papers with multiple actions or sets of actions, and 6 papers in which the action is not clearly or completely defined. The details related to each paper are given in Table 3.

Action selection methods and parameters. To select the actions, various action selection methods can be used. Based on the data we retrieved, 71 (44%) papers did not state which action selection they used, which is a significant number. ϵ-greedy has the highest usage and was observed in 51 (32%) papers. Softmax (or Boltzmann) and greedy methods are used in 17 (11%) and 7 (5%) papers, respectively. Other action selection methods with a frequency of 4 or less are Random, Distributed W-learning (DWL), credit assignment algorithms, ϵ-softmax, and Upper Confidence Bound (UCB). Two papers employed multiple or combined action selection methods, while two others compared various action selection methods.

- *ϵ-greedy strategy and SoftMax*: ϵ-greedy uses the epsilon term to balance exploration and exploitation of the environment, encouraging the former early on and switching to the later as the algorithm learns. It randomly selects actions for the next round based on the values of the exploration rate (ϵ), discount factor (γ) and learning rate (α). SoftMax behaves similarly but with weight parameters either assigned or learned to each action. They can be quite sensitive to changes and encourage an outcome where important parameters have more value. The sensitive nature of the weights makes them tricky to learn/find and can affect the performance of the algorithm significantly.

We collected data about the RL parameters, including: exploration rate, discount factor and learning rate. We found that the information was not displayed in a number of cases. 121, 60, and 86 out of 160 papers did not reveal the information about these parameters, respectively. In papers where the data are presented, the most common option was for the authors to acknowledge that these parameters exist within the range of (0,1). While it is a piece of information, it is not especially useful seeing as this information is well known among common practitioners of reinforcement/machine learning.

State space and action space discretization. Reducing state space (for the states such as queue size, flow rate, and density), action space, and even reward space is one of the ways that can be used to reduce computational cost to make the methods more applicable in real-time process in the real world. To this end, the continuous states are grouped, for example, in 3 levels: low, medium, and high. Another way is using the comparison between the states with the previous step. This way the space is divided into two groups: better or worse. Reducing the space using grouping the data may come at the expense of the lower accuracy of the results, thus lowering the efficiency or optimality. Nonetheless, 59 (37%) of the papers used discretization while 91 (57%) did not discretize the state or action spaces. 3 (2%) papers used discretization only for the discrete methods, but not for the continuous methods that were evaluated.

Tabular vs approximation-based methods. Another point that impacts the efficiency of the performance of the methods in real-world is using tabular RL methods where a look-up table is used to map the spaces. Approximation methods are used to provide a good approximation of the states that were not experienced in training. Interestingly, the number of the papers that used either of these two methods are eerily close: 74 (47%) papers used a tabular method while 72 (45%) used approximation methods.

The approximation methods that are used in our pool of research studies are shown in Table 7. The applied neural networks, as one of the approximation methods, in the papers are also presented in the table. 38 papers used different neural network approximation methods, and the second-most used group is the Tile Coding, observed in 7 papers. One of these applied its own proposed method (Prashanth and Bhatnagar, 2010).

- *Neural Networks*: NNs are loosely designed after the brain and are constructed for tasks such as pattern recognition, labelling, and processing of data. They are commonly used for clustering, classification, and predictions. In the case of traffic control, prediction is the prominent use, though other use does occur. NNs consist of 3 main sections: input section, hidden section, and output section. Usually, there is only one of each input and output layers, but the hidden section may contain more. Each layer contains nodes. In the output layer, the number of nodes often corresponds with the number of possible outputs. In the input layer,

Table 7: First column) approximation methods used in RL-NTS, and second column) the types of NNs used in RL-NTSC.

Approximation method	Neural Networks types
NN	denoising auto encoder (DAE)
tile coding	Deep NN
linear model tree	feed forward NN (FFNN)
simultaneous perturbation stochastic approximation (SPSA)	recurrent NN (RNN)
smoothed functional (SF)	convolutional NN (CNN)
phase gate	deep convolutional network (Deep CNN)
decomposable Q-function (DQF)	perceptron
radial basis functions (RBF)	graph attentional network (GAN)
triangular-shaped functions (TF)	DNN
linear function approximation (Linear FA)	graph convolutional network (GCN)
average cost algorithm	incremental Gaussian mixture network (IGMN)
iterative approximation	back propagation NN (BPNN)
KNN	fuzzy NN (FNN)
linear combination (LC)	neural fitted Q-iteration (NFQI)
cerebellar model articulated controller (CMAC)	wavelet NN (WNN)
connectionist Q-learning framework (CQF)	
gradient-descent linear function approximation	
double deep Q-learning	
fuzzy granulation	
linear regression (LR)	
sigmoid regression (SR)	
long short-term memory (LSTM)	
linear Q-function	

the number of nodes often corresponds with the different types/sources of input data. The number of nodes within the hidden layers depends on what each layer is designed to do, as well as changes with the purpose of the network. There are many types of NNs in deep RL.

  - Artificial Neural Networks (ANN): Known as a feed forward network, all incoming information is only processed and pushed in the forward direction.
  - Recurrent Neural Networks (RNN): Unlike ANN, this neural network pushed processed data back to previous layers and nodes. It shares parameters across different time steps and results in fewer overall parameters. The fewer parameters allows for a smaller network.
  - Convolutional Neural Networks (CNN): These networks use an extra step called convolution (for which it was named), which involves applying different "filters" to reduce the complexity of the input data. As the information filters through the network, these filters can be applied to highlight specific features of the data. This is one of the most common neural networks and is used in many disciplines.

  With neural networks, deep RL becomes highly suited for complex environments presented by intersections and the dynamic changes that occur within a day of traffic.

- *Tile Coding*: Tile coding is another well-known function approximator. Unlike the continuous methods such as radial basis functions (RBF), tile coding is a discretization method which is used in RL. It is a piece-wise constant approximation method that approximates the action-value functions by partitioning the state space into small regions with a constant reward value. Tilings design considers three main components: width of tiles, the resolution, and the number of required tilings based on the hyper-volume of the whole state space. For more information about Tile coding method, the reader can refer to (Abdoos et al., 2014).

Deep reinforcement learning based methods. Deep RL takes a different approach when dealing with the complex influx of data associated with traffic. It incorporates neural networks into RL algorithms and combines the advances

in training layered neural networks into abstract high-level representations of the raw input data, giving non-linear methods (El-Tantawy et al., 2015), (C. Li et al., 2018). These "layers" allow the agent to look at smaller, more reduced versions of data to extract information without an overload. The neural networks are what allow an algorithm to go "deep" and work with large or complex data input more efficiently.

Here, we compare Q-learning and Deep Q-learning. Q-learning, is the process of iteratively updating Q-Values for each state-action pair using the Bellman Equation until the Q-function eventually converges to $Q*$. Instead of estimating the Q-value of each state-action pair separately in Q-learning, deep reinforcement learning algorithms (Mnih et al., 2013) use deep neural networks as function approximators to map from states to Q-values. This makes possible the use of a larger and/or continuous state space through parameterization (Lillicrap et al., 2015). The integration of artificial Neural Nets (NNs) into the Q-learning process is referred to as Deep Q-learning, and a network that uses NNs to approximate Q-functions is called a Deep Q-Network (or DQN). In other words, DQN is a Q-learning, which is parameterized with deep NN with parameters θ, i.e. $Q(s, a; \theta)$. The neural network input is the state, the number of output neurons is the number of the possible actions, and the targets are the Q-values of each of the actions.

Unlike Q-learning, whose convergence in the limit (infinity) is guaranteed, we do not have such guarantees for DQN. This is because (i) the data set is not i.i.d. and (ii) as the agent learns, the targets move (Van der Pol and Oliehoek, 2016). To ameliorate this issue, different methods can be used, such as dueling architecture (Ziyu Wang et al., 2016) to improve stability and target network(Mnih et al., 2015) to solve the overoptimistic problem. Dropout (Srivastava et al., 2014) can also be used to make the controller more robust and prevent the neural network from overfitting.

Learning rate needs to be optimized in training deep neural networks. To this purpose, different optimization methods can be used, such as stochastic gradient descent, Adagrad, and Rmsprop. In this line, Adam optimizer (Kingma and Ba, 2014) is an adaptive learning rate optimization algorithm, which is computationally efficient, has little memory requirements, and is generally fairly robust to the choice of hyper parameters (Goodfellow et al., 2016). In addition, experience replay is also used to help with stability and convergence behaviour of the algorithm when using a non-linear function approximator.

Deep learning was used in 27 (17%) papers. This is certainly a reasonable portion of the study, and the point to be noted is that using deep learning in RL in the network-scale began in 2016 with 2 papers. After a year with no cases, deep learning was used in 5 studies in 2018, followed by a sharp increase in 2019 with 18 papers. It is expected that this trend will continue into the future, as in the first season of 2020, 3 out of 6 research papers used deep learning. The papers that used deep RL methods are identified in Table 5. SUMO has the frequency of 16 (out of 28) in conducting simulations for the deep learning related methods. The traffic simulation software tools used for deep learning are depicted in Figure 9.

3.3. Environment attributes and traffic simulation

In this section, we discuss the environment attributes, including network type, vehicle class, data source, and data communication processing. This classification presented in Figure 8.

3.3.1. Simulation and simulated networks

16 traffic simulator software or platforms are identified in these studies to simulate the traffic. These include SUMO[3], VISSIM[4], PARAMICS[5], GLD[6], AIMSUN[7], ITSUMO (Silva et al., 2004), CityFlow[8], TRANSYT[9]/TRANSYT-

[3] https://www.eclipse.org/sumo/
[4] https://www.ptvgroup.com/en/solutions/products/ptv-vissim/
[5] https://www.paramics.co.uk/en/
[6] http://www.sf.net/projects/stoplicht
[7] https://www.aimsun.com/
[8] https://cityflow-project.github.io/
[9] https://trlsoftware.com/products/junction-signal-design/transyt/

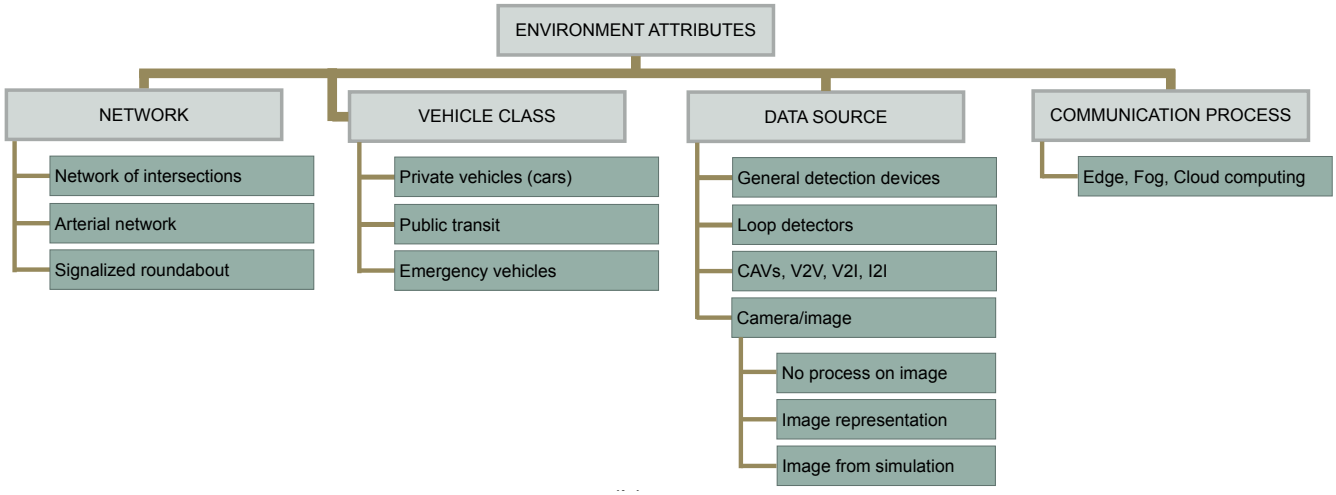


Figure 8: Environment attributes in RL-NTSC.

7F[10], MATLAB[11], AIM (Dresner and Stone, 2008), TSIS[12], MATISSE (Torabi et al., 2018b), USTCMTS (used in (Shi and F. Chen, 2018)), SMPL (used in (Su and Tham, 2007)), and SeSAm (used in (Bazzan et al., 2007)). See the distribution of frequency of the traffic simulators in included papers in Figure 9. Since the source of the last three simulation software tools is not found, we referred the reader to the papers in which they are used, for further information.

Of interest is that SUMO was used for the first time in 2015 and has since become the most frequently used software in this area, with 17 out of 27 occasions in 2019 alone. The second most frequently used software, (i.e. VISSIM) started being used in 2010, and PARAMICS, the first well-known traffic simulator has been in use since 2003. Prior to this the tools were either custom-built or not clearly outlined in the research. In 18 studies, the authors designed or used a custom-built environment for simulations. What is surprising is that 27 studies (16.7%) did not state the simulation tools that they used for the test, not including the those that did not use any simulations.

With regard to the type of maps used to test the proposed RL methods by the included studies, of those that used maps, 100 (62%) papers used a synthetic map, 45 (28%) papers used a real-world map, and 9 (6%) papers used both types, see Figure 10. Moreover, 49 real maps are used from 14 countries for simulation purposes, with China topping the list at 10 maps, followed by the USA, Ireland and Canada, see Figure 10. These maps are mostly large-scale networks of intersections.

Figure 11 shows the distribution of the number of intersections studied. Though many studies are still currently being held with fewer than a dozen intersections, there is also work on 25+ intersections. And while 44% of the papers run simulations in small-scale networks with 8 intersections or lower, we observed that the recent trend is to test the proposed methods in medium and large-scale networks. The size of the network is also important as it may impact the exponential growth of state-space and action-space and generally the complexity and computational effort required to reach a solution through an RL method, whether for training or testing stages.

We categorized the type of the testbed network into three main groups, including: (1) the network of intersections, (2) arterial network, and (3) the signalized roundabout networks. As already mentioned, the isolated intersection case is excluded from our study. The arterial network is an open network, as compared to a closed network. Arterial

[10] https://mctrans.ce.ufl.edu/mct/index.php/hcs/transyt-7f/
[11] https://www.mathworks.com/products/matlab.html
[12] https://mctrans.ce.ufl.edu/featured/tsis/Version6/index.htm

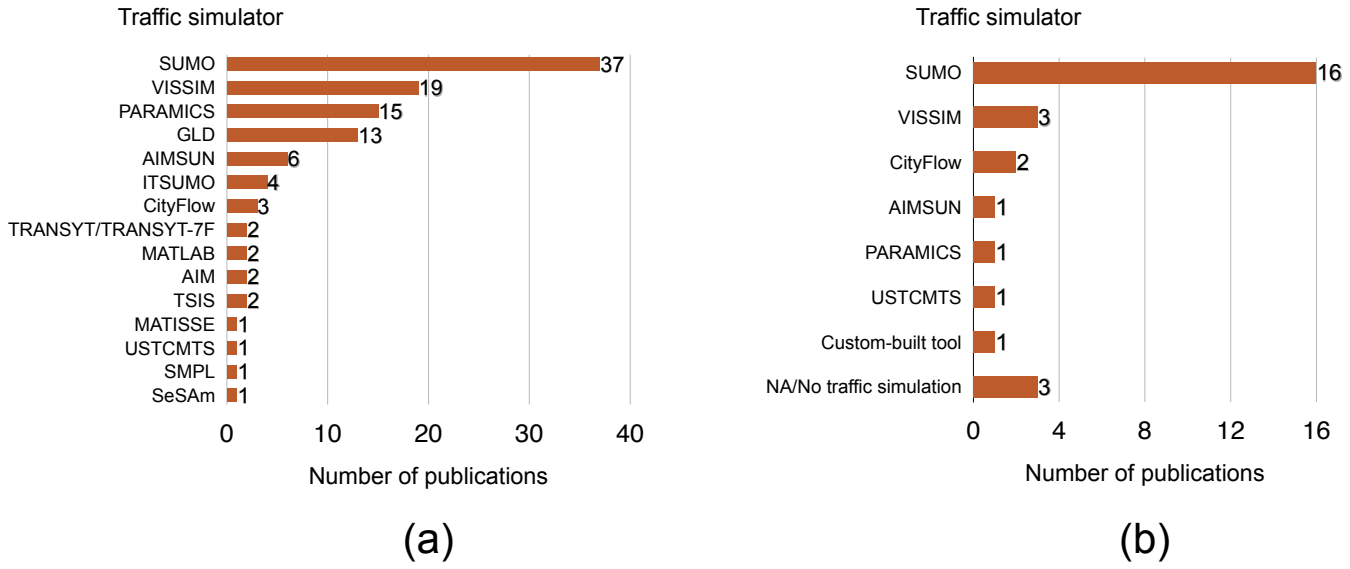


Figure 9: Traffic simulators: a) distribution of frequency of the traffic simulators in RL-NTSC, and b) distribution of frequency of the traffic simulators in deep RL-based methods in RL-NTSC.

network control is applied on a sequence of intersections to provide preference to progressive traffic flow along the arterial. Unlike the isolated intersections, the intersections in the arterial network operate as a system and the system coordinates timing of adjacent intersections. On the other hand, the network of intersections is considered as a closed loop. Being a closed loop, the network demands at least four intersections. The networks of two and three intersections may have the characteristics and behaviour of both groups (the network of intersections and arterial network); therefore, we considered them separately as two sub-groups under the first group, i.e. the network of intersections. Finally, the signalized roundabout networks deals with both approaching and circulatory lanes, which are fundamentally different from the first two groups. The signalized roundabout networks involve more than one set of traffic signals in a node. (Rizzo et al., 2019b,a), both published in 2019 are only two papers in the literature that studied the signalized roundabouts, which indicates a potentially open research problem on the RL area.

The dynamics of signalized roundabouts is complex because it deals with both approaching and circulatory lanes. The conflicts between approaching and circulatory flows cannot be solved by metering only the approaching lanes and reacting consequently because the circulatory lanes may be occupied. (Rizzo et al., 2019b) proposed a deep RL method for signalized roundabouts in congested networks to maximize traffic flow while being able to avoid traffic jams in connected junctions. In another research conducted in this domain, (Rizzo et al., 2019a) studied the possibility of deriving explanations from a neural network agent (trained using Policy Gradient) for TSC in a signalized roundabout. How the agent learns to react differently based on each specific lane's traffic by implicitly predicting the route of the traffic and thus its future circulatory occupancy is explored. This is done by analyzing the relation between the agent phase preferences and the actual traffic, assessing the agent capability of reacting to the current detectors state, and estimating the effect of the road detectors state on the agent selected phases through the SHAP model-agnostic technique. The results of this research reveal that it is possible to extract meaningful explanations on the decisions taken by the policy. Further research involves the study of the trade-off of accuracy in comparison with a complex deep learning controller.

And, only 17 papers worked on arterial networks (out of 189 network configurations in 160 papers), all between 4-8 intersections except one with 16 intersections, which is the maximum number in an arterial network. The highest numbers of intersections in a network are 225 ((Yongheng Wang et al., 2016; Ni and Cassidy, 2019)), 196 ((Wei et al.,

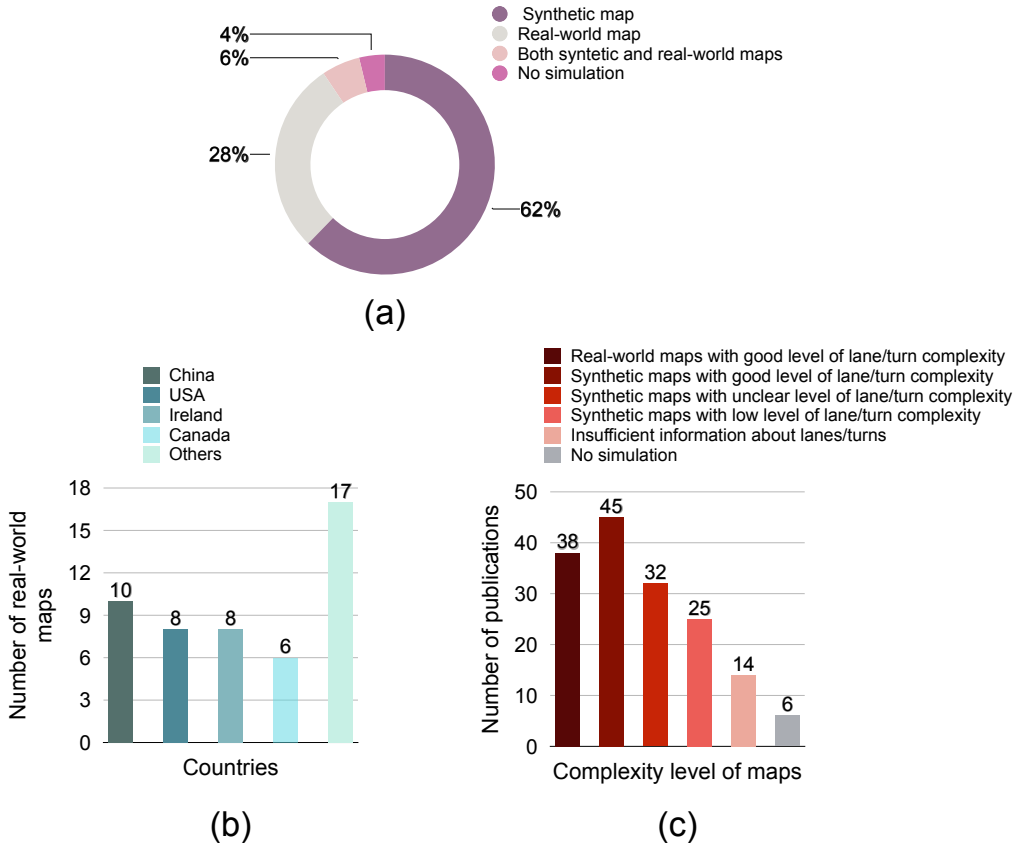


Figure 10: Maps: a) distribution of type of maps used in RL-NTSC, b) the frequency of real-world maps in terms of countries in RL-NTSC, and c) distribution of type of maps in terms of level of lane/turn complexities used in RL-NTSC.

2019b)), and 127 ((Zhou et al., 2019)), respectively. The paper (Yongheng Wang et al., 2016) is the first paper in the literature that tested the proposed method in a network with the highest number of intersections: 225 intersections in 2016. In 5 of the papers that we came across, the number of intersections of the network or arterial network is not released, and in 6 papers there is no simulation used. See Figure 11.

We identified three vehicle classes, including private vehicles (cars), public transit (buses), and emergency vehicles (ambulances). Among 160 articles, only 3 of them addressed the public transit (2 papers) and emergency systems (1 paper) in RL-NTSC.

In the public transit domain, there are two types of vehices, including buses and streetcars. We address streetcars in one of the three control schemes, which is streetcars bunching control. Excluding streetcars, there are 2 research works regarding buses. One considers transit priority, while another one does not. In 2014, (Chanloha et al., 2014) developed a distributed CTM-Based MARL for network-scale signal control with transit priority that outperforms pre-emptive and differential priority control methods because of the improved awareness of the signal switching cost. To eliminate the need for feature extraction in the state space and to directly use available information received from the high-detailed traffic sensors (Shabestray and Abdulhai, 2019) proposed a multimodal Deep RL based traffic signal controller that combines both regular traffic and public transit and minimizes the overall travellers' delay through the intersection.

With regard to emergency vehicles, in 2017, to detect and give priority to emergency vehicles, (Kristensen and

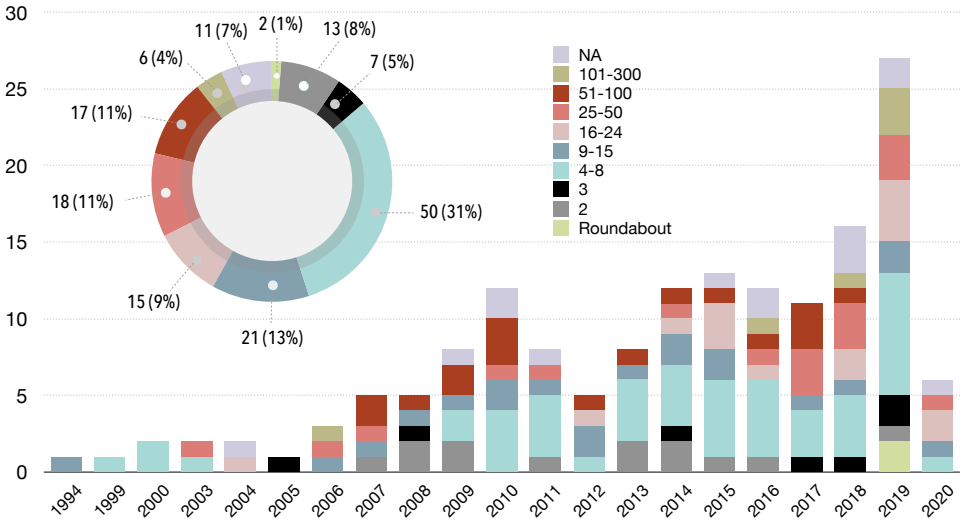


Figure 11: Distribution of number of intersections in simulations: in all studies (pie chart), and over time (bar chart).
Notes: 1) The data of arterial and network have been merged and the highest network size in each paper is reported, 2) NA represents the papers in which the number of intersections in the arterial or network is not explicitly revealed, 3) The years with no publication, i.e. 1995-1998 and 2001-2002, are not shown.

Ezeora, 2017) proposed a reinforced traffic control policy that reduces the waiting time of emergency vehicles at intersections as well travel time of other vehicles using a multi-agent system development framework (JADE). Also, (Y. Liu et al., 2017) is among very few articles that addressed pedestrian element along with private vehicle class in the network.

It is important that the considered map as a testbed would be consistent with a real-world set-up. For example, one-lane or one-way crossing links cannot replicate the common cases in real-world scenarios. This may dramatically impact the usability and performance evaluation of the methods, specifically in terms of computation efficiency. Based on the collected data, most of the papers used a good level of complexity in the number of lanes and turns. 24% of the papers used real-world maps with multiple lanes, 28% used two or more lanes (including 13% using three or more lanes) with a good level of complexity, and 20% used the synthetic maps with multiple lanes but without significant information on the number and types of turns. See the number of publications in each category in Figure 10. By a good level of complexity, we mean that in a regular driving style the through and left lanes are involved regardless of the right turn. The through and left lanes in the opposite approaches have conflicting right of ways and add to the state and action spaces, which increases the complexity of the problem. In the regular driving style, right turns can usually be accommodated simultaneously with either through or left lanes and do not impact the state and action dimensions. Also, in a right-side driving style, the through and right lanes are considered regardless of a left turn (similar logic to the regular driving applies here). In 25 (15%) of the papers, a low level of lane/turn complexity, including one-lane links, one-way crossing links, or multiple-lane links with only through lanes is represented. 9% of the papers did not reveal any or enough information about the lanes and turns.

3.3.2. Traffic data collection and traffic demand

We identified four categories of data source for the proposed methods, including general detection devices (66%), loop detectors (17%), vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) and infrastructure-to-infrastructure (I2I) (11%), and camera/image (6%). By general detection devices, we mean that the authors did not specify any specific data-source. The papers that use loop detectors provide either no specific design or specific designs like (1) one detector at the stop-line, (2) one detector at a distance from the stop-line, (3) one detector at the upstream

(end of the lane), (4) two detectors at the stop-line and at a distance from the stop-line, or (5) two detectors at the stop-line and at the end of the lane.

The third category covers the methods in V2V, V2I, and I2I environment. Connected vehicles is a specific class in this category where vehicles can exchange data with other vehicles and the infrastructure. The fourth category presents the methods that just use traffic cameras as a data source without any process on images or are inherently image-based. In image-based methods, there are two directions: (i) an image-like representation, i.e. image representation of vehicles' positions, where an image is a matrix of pixel values of a view of intersections, and (ii) images retrieved from traffic simulation software are used as the data source.

In line with the data and the data sources, computing paradigms in the data communication process in RL-NTSC is a new research area and the research on application of computing paradigms, specifically edge and fog computing, in RL-NTSC is topical and has started since 2019 with three papers. (R. Gao et al., 2019) designed a framework of edge computing under the NTSC scene and proposed a cooperative NTSC algorithm based on MARL to avoid the curse of dimensionality, provide minimal response time, and reduce network load. (Zhou et al., 2019) proposed a large-scale Edge-based RL (ERL) solution to better alleviate congestion in complex traffic scenarios based on Edge Computing nodes for traffic data collection. They concluded that ERL in distributed edge servers has much better scalability and faster Deep NN training than the cloud service. (Q. Wu et al., 2019) proposed a traffic control architecture based on fog computing paradigm and a distributed RL algorithm (that connects traffic signals, vehicles, Fog nodes and traffic cloud) to overcome communication bandwidth limitation and reduce communication delay, make real-time traffic condition information available to vehicles, and lower the probability of traffic congestion in the city through generating traffic signal control flow and communication flow for each intersection. This is not only suitable for current vehicles but also more useful for driverless vehicles anticipated in the future, as it will be able to plan its route much more intelligently with information from the Fog node.

Although there are numerous methods listed in the literature, for traffic signal control, they are mostly effective in low traffic loads. The main challenge in traffic control is managing the high load of traffic representing rush-hour in the morning or afternoon in a network that may lead to a spillback condition in different links. Our research found that 74% of the studies simulated the traffic condition in high demands, close to saturation, saturated, or oversaturated conditions to test the efficiency of the proposed methods in traffic congestion (labelled as *Sat* in Table 3). Nevertheless, among these, only 5% explicitly applied and addressed a spillback prevention strategy, and 6% analyzed or mentioned that the proposed method is able to prevent spillback (labelled as *Spillback* in Table 3). 6% of the papers explained spillback but never considered or applied it, whereas in 56% of the studies with high demand, spillback is not even mentioned. 22% neither addressed high demand and spillback, nor explicitly mentioned them (labelled as *Neither* in Table 3), and 4% provided no evaluation (labelled as *No Evaluation* in Table 3).

3.4. RL-NTSC application domains

Figure 12 depicts the NTSC application domains (or scenarios) and the frequency of publications in each domain based on the control method scheme and environment attributes defined in Figures 7 and 8. The references to the publications, except for the first row to save space, are also provided. This figure helps to locate which areas have already researched, what are the publications in the areas, and in which areas there is still room for improvement and development. This can be found out based on the frequency of the publications in the areas.

3.5. Major first events in RL-NTSC and authors' key statements

Figure 13 is an infographic timeline that aids in identifying past and current trends, specifically highlighting the research areas and challenges that have come to the fore most recently. It essentially indicates major first events in RL in NTSC. This helps to find out since when a research line has started. In addition, **??** includes key statements made by the authors of the studies included in this paper.

Control method scheme	Vehicle class	Network	Communication process	Data source	Image data appliacation	CAVs application	Number of publications [reference]
Regular network control	Private vehicles	Network/ Arterial	-	General/Loop detectors	-	-	**126**
			-	V2V/V2I/I2I	-	CAVs	**6** (Houli et al. 2010, W. Liu et al. 2014, W. Liu et al. 2017, Al Islam et al. 2018, Aziz et al. 2018, Vinitsky et al. 2018)
					-	-	**9** (Wiering 2000, Kuyer et al. 2008, Khamis et al. 2012, L. Xu et al. 2013, Box and Waterson 2013, Khamis and Gomaa 2014, Gaikwad et al. 2016, Kao and Wu 2018, Lemos et al. 2018)
			-	Camera/ image	No process on image	-	**3** (Abdelgawad et al. 2015, Y. Liu et al. 2017, Gong et al. 2019)
					Image representatio	-	**4** (Chu et al. 2016a, Wei et al. 2018, Reda et al. 2019, Yang et al. 2019)
					Image from simulation	-	**2** (N. S. Jadhao and A. S. Jadhao 2014, J. Lee et al. 2020)
			Edge, Fog, Cloud computing	General/Loop detectors	-	-	**1** (Gao et al. 2019)
				V2V/V2I/I2I	-	CAVs	**1** (Q. Wu et al. 2019)
				Camera/ image	No process on image	-	**1** (Zhou et al. 2019)
		Signalized roundabout	-	General/Loop detectors	-	-	**2** (Rizzo et al. 2019a,b)
	Public transit	Network/ Arterial	-	General/Loop detectors	-	-	**2** (Chanloha et al. 2014, Shabestray and Abdulhai 2019)
	Emergency vehicles	Network/ Arterial	-	V2V/V2I/I2I	-	CAVs	**1** (Kristensen and Ezeora 2017)
Perimeter control	Private vehicles	Network/ Arterial	-	General/Loop detectors	-	-	**1** (Ni and Cassidy 2019)
Streetcar bunching control	Public transit	Network/ Arterial	-	General/Loop detectors	-	-	**1** (Ling and Shalaby 2005)

Figure 12: NTSC application domain and frequency of publications in each domain.

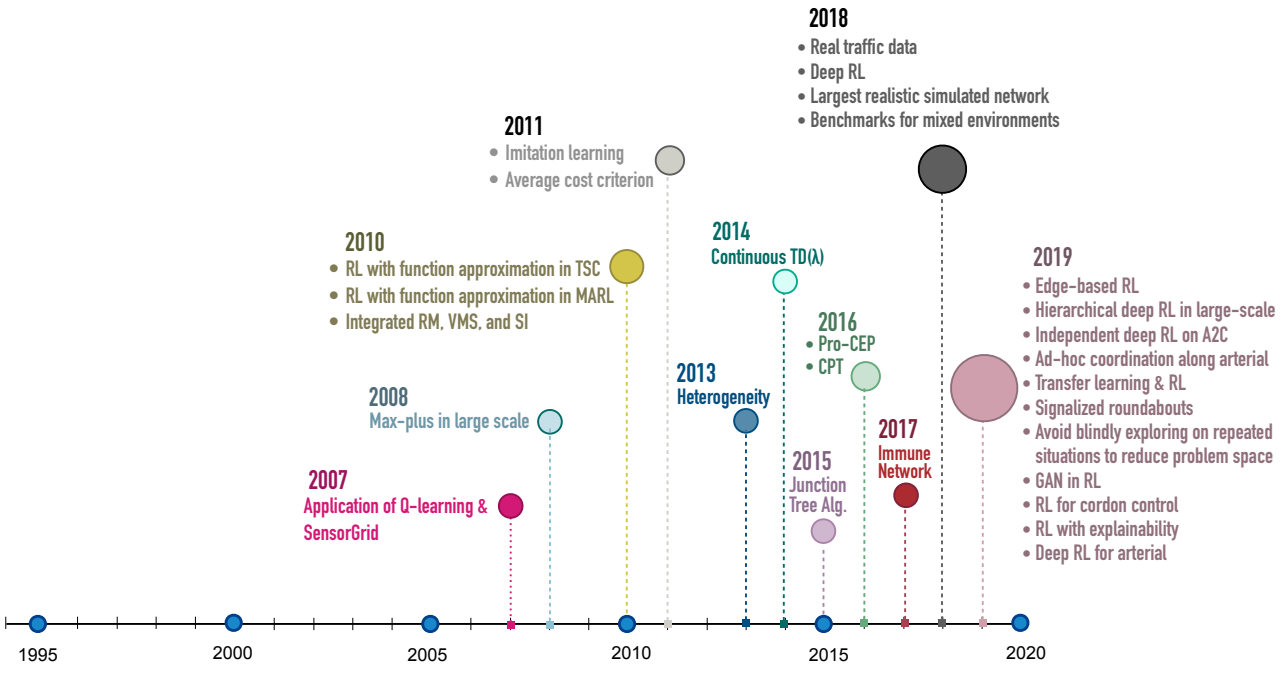


Figure 13: Major first events in RL in NTSC collected from the included papers.

Table 8: The details of major first events in RL in NTSC collected from the included papers.

Citation	Statement
Su and Tham, 2007	This **application of Q-learning and SensorGrid** can be seen as the first step towards expanding the usage of SensorGrid.
Kuyer et al., 2008	The first application of **max-plus** to a **large-scale** problem (not in small applications) and thus verifies its efficacy in realistic settings.
El-Tantawy and Abdulhai, 2010	The first study that has tackled the **Integrated traffic control problem** (Ramp Metering (RM), Variable Message Signs (VMS), and Signalized Intersections (SI)) to find a **closed-loop optimal control solution** using a **coordination** mechanism that minimizes the communication requirements.
Prashanth and Bhatnagar, 2010	The first application of **RL with function approximation** for NTSC.
Waskow and Bazzan, 2010	The first attempt to tackle the dimensionality problem in MARL by means of **function approximation**.
Natarajan et al., 2011	The first adaptation of a Statistical Relational technique for the problem of learning relational policies from an expert (**imitation learning**).
Prashanth and Bhatnagar, 2011	The first to design RL-based NTSC algorithms that minimize a long-run **average cost** criterion.
Nuli and Mathew, 2013	No traffic adaptive control model exists to account for **traffic heterogeneity** and limited lane discipline.
El-Tantawy et al., 2014	The first study to investigate the effect of **TD(λ)** methods for NTSC as a **continuing task** (i.e., not a finite episode) with a **discounted reward** in which looking ahead to future steps is less important compared to a finite episodic task with undiscounted reward.
Zhu et al., 2015	**Junction Tree Algorithm** has not been applied to address the coordinated signal control problem.
Yongheng Wang et al., 2016	Other works with the same functionality (predicting future system states by **Proactive Complex Event Processing**) have not been found.
Prashanth et al., 2016	The first work to combine **cumulative prospect theory (CPT)** with RL, and to investigate (and define) **human-centered RL**.
Darmoul et al., 2017	The first to integrate **Case-based reasoning** and RL for NTSC and integrate **immune features** within MARL to achieve disturbance management.
Wei et al., 2018	None of existing studies have used the **real traffic data** to test their methods.
C. Li et al., 2018	The study of the **applicability of deep RL** on the road network has not yet been carried out.
Torabi et al., 2018a	Validated on the **largest realistic simulated traffic network** published to date for collaborative multi-agent based NTSC.
Vinitsky et al., 2018	The first to propose **a standard set of benchmarks for traffic control in a micro-simulator** and a framework for simultaneously learning control for a **mixture of AVs interacting with human drivers and infrastructure** in which **deep RL** can be applied to the control task.
Zhou et al., 2019	**Edge based RL** is the first RL proposal to optimize traffic signals on neighborhood scale.
T. Tan et al., 2019	The first attempt to use **hierarchical deep RL** models in large-scale NTSC.
Chu et al., 2019	The first paper to present a fully scalable and decentralized MARL algorithm for the state-of-the-art deep RL agent: **independent advantage actor critic (IA2C)**, within the context of NTSC by extending the idea of Independent Q-learning on A2C.
Wei et al., 2019a	The first time that the individual RL model **automatically achieves coordination along arterial without any prior knowledge**.
N. Xu et al., 2019	The first work to consider the impact of slow learning in RL on real-world applications by the effective **transfer of RL algorithms** trained on simulated traffic to the real-world traffic to reduce the mistakes to be made in the real world.
Rizzo et al., 2019b	The first to address **signalized roundabouts** in congested network, as a complex TSC scenario using a **deep RL** method.
Zheng et al., 2019a	The first work to **reduce the problem space** and explore different scenarios more efficiently, so that the RL algorithm can find the optimal solution within a minimal number of trials, **instead of blindly exploring on repeated situations**.
Wei et al., 2019b	The first work to use **GAN** in RL for NTSC and to conduct experiments on the large-scale road network with **hundreds of traffic signals**.
Ni and Cassidy, 2019	The first to extend RL to the **cordon-control** problem.
Rizzo et al., 2019a	The first to consider **model explanation methods** such as LIME and SHAP for the explanation and interpretation of RL agents decisions that can be verified by domain experts (**RL with Explainability**).
P. Chen et al., 2019	This study is among the earliest to apply **deep RL** for **arterial** adaptive signal control.

3.6. Code availability

Still another feature that we investigated is the availability of the code, which we considered a good resource for those new to the field, and useful for reproducing the research. We found that in 10 papers in NTSC the authors made their code available, and these can be found in Table 5. (Brys et al., 2014) is the first paper in the area of RL in NTSC that made the code available in 2014. 7 of these 10 papers are deep methods published in 2018 and 2019. 2 papers provide actor-critic methods, while the rest are based on Q-learning. The codes are written for SUMO (5 papers), CityFlow (3 papers), GLD (1 paper), and AIM (1 paper). The approximation methods include NN, SPSA, Phase gate, and Tile coding.

3.7. Evaluation

150 (94%) of the studies provided an evaluation, while 7 (4%) did not. Three (2%) papers provided self-comparison, meaning that they compared the variations of the proposed methods with each other, but not with other NTSC methods. The papers that did not provide an evaluation are identified in Table 5.

The authors used different TSC methods and performance measures to compare and validate their proposed method. Fixed time methods alone are inefficient in evaluating other methods because they are unable to adapt to the traffic flow changes, however, they are sufficient to use in exploring the feasibility of a proposed method or the proof of concept. We found that 27 (17%) of the studies used only fixed time or random methods/policies for comparisons. In 39 (24%) cases, the TSC methods are used where no RL method such as actuated and adaptive methods are included, as they provide for better evaluation. In 76 (47%) studies, RL has been used either alone or with other types of methods. Involving RL methods for evaluation, however, cannot always guarantee a perfect evaluation. Generally, if an RL method is used along with actuated and adaptive methods, it can provide a great evaluation, specifically when the comparison is made with state-of-the-art RL methods proposed by the other authors in the field. We collected these RL methods as a reference for the readers in Table 9. The table also provides the citation of referenced papers that used evolutionary and meta-heuristics algorithms, real-world, fixed time, and adaptive methods. It might be of interest to the reader to know the number of methods that are used in these papers for evaluation/comparison purposes: 1 (33%), 2 (31%), 3 (14%), 4 (5%), and 5 to 8 (5%), demonstrating that comparison of a method with only one other is the most common.

Among the performance measures, delay is the most frequent performance measure in the papers with 71 occurrences (20%), followed by travel time and waiting time and queue size (each 12%), number of stops and speed and throughput (e.g. the number of the vehicles passed the intersection) (each 6%), and environmental measures (5%), which accounts for 80% of the papers. We found 33 unique performance measures that are listed in Appendix C. Each unique measure delegates several similar measures. The list shows a variety of measures that authors may consider or use in their following research works. Appendix C also depicts the top ten performance measures.

4. Discussion

4.1. Common future works and research opportunities

During the course of our investigation, a few recurring steps the authors took in order to advance their research into the future, were noticed. One of the most recommended areas for future investigation involves testing the traffic signal controllers in the real world. Given that the final hurdle from theory to implementation is to see if the concept can successfully direct traffic at busy and unpredictable intersections and not just in simulations, this comes as no surprise. Authors also look to expand their work to a bigger network and to increase the number of phases that controllers could select, in addition to making traffic signal control a multi agent system and adapting to bigger intersections. In the same vein, the diversification of the proposed traffic signal formulations so that they could potentially be of greater use to more people, is also important. Adapting plans to different modes of transport, including motorized traffic such as public and mass transit, taxis, and freight vehicles, and non-motorized traffic like pedestrians and bikes are challenging at best. Incorporating better communication methods between

Table 9: Methods used for comparison

Compared Methods	citation
RL methods	
Distributed and multi-agent RL	Arel et al., 2010; El-Tantawy et al., 2013; L.-H. Xu et al., 2013; Dusparic and Cahill, 2009a; Kuyer et al., 2008; Prabuchandran et al., 2014; Araghi et al., 2015; Camponogara and Kraus, 2003; Jin and Ma, 2018,Watkins and Dayan, 1992; Abdulhai et al., 2003; Riedmiller, 2005; Claus and Boutilier, 1998; M. Tan, 1993; Choy et al., 2006; Vu et al., 2018
Actor-critic RL	Konda and Borkar, 1999; Sutton and Barto, 1998; Baird III, 1999
Deep RL	Van der Pol and Oliehoek, 2016; Nishi et al., 2018; Chu et al., 2019; Wei et al., 2018,Schutera et al., 2018; J. Gao et al., 2017; C. Wu et al., 2017; Zheng et al., 2019b
Distributed Intersection Management Protocol	Liang et al., 2018
Junction Tree Algorithm	Zhu et al., 2015
TILDE	Blockeel and De Raedt, 1998
Propositional Function-Gradient Boosting	Dietterich et al., 2004
Wiering Controller (TC-1)	M. A. Wiering, 2000
Traffic Controller with State Bit for Congestion	Bakker et al., 2005
SPSA-based multi-agent	Spall and D. C. Chin, 1997
Evolutionary and meta-heuristics methods	
Genetic Algorithm	M. Wiering et al., 2004
Cooperative Ensemble	Srinivasan and Choy, 2007
Real-world methods	
GLIDE (SCATS-like method in Singapore)	Keong, 1993
Fixed time methods	
Optimized fixed time (TRANSYT)	Robertson, 1969
Fixed Time	Miller, 1963
Fixed time with random offsets	Koonce and Rodegerdts, 2008
Webster method	Webster, 1958
Adaptive methods	
Saturation balancing method	Richter et al., 2006
Self-Organizing Traffic Light Control (SOTL)	Cools et al., 2013; Gershenson, 2004
Max Pressure	Varaiya, 2013
Longest Queue First - Maximal Weight Matching	Wunderlich et al., 2007
Anticipated All Clearing policy	Lämmer and Helbing, 2008
Cluster based adaptive methods	Zhigang Wang et al., 2008; Daeinabi et al., 2011

their controllers, accounting for delays in communication and addressing noise (unwanted data) that their sensors might pick up are other points of focus for the future. Still another popular theme that arose out of our study is improving the performance of controllers. Specifically, a common direction is to change the definition of the reward function and obtain an improved state space, thereby allowing the controller to render a decision. The final future implication from this study is online learning, which is a popular choice since its strength lies in the controller's ability to continuously adapt to traffic signal conditions. Although some efforts are already underway to achieving this, such as focusing on reducing the time required for learning during this continuous process, the area is still in its infancy.

4.2. Key findings

This paper allows us to have a comprehensive view of the past 25 years of research on applying RL to NTSC. This view allows us to see that the community has employed classical approaches (e.g., Q-learning) in the vast majority of the investigations. Thus, we see a large avenue for extensions, especially given that Q-learning is a tabular method

and, as such, it is not fully equipped to deal with continuous spaces and/or with centralized approaches, in which the state space tends to be vast. Related to this issue, deep learning is advancing to fill the gap left by methods that do not fully deal with huge state (and possibly, action) spaces. The number of papers employing deep learning is increasing, as demonstrated in our literature review.

The use of non-commercial microscopic traffic simulators is on the rise, with SUMO being used more and more, especially within the computer science community. Associated with this trend, is that there has been an increase in the exchange of code and experiences (e.g., SUMO has an active mailing list), which is certainly a positive trend.

Furthermore, we have noted that there is a lack of interaction between traffic and transportation engineering practitioners and researchers investigating the use of RL on NTSC. One of the consequences of this is that a high number of papers does not include or deal with real data, thus challenging the proper validation of the experimental results. Also, no testbeds are being proposed. In fact, real-world scenarios are lacking, for which one could find at the very least a detailed map (including geometry), actual demand, fixed-time signal timings and target measurements to be used for comparison purposes. Moreover, the creation of testbeds would likely bring different communities together around common goals.

Based on the revealed data by the authors of the included papers and our analysis, the literature of RL in NTSC motivates the following areas to expand as open research problems and research opportunities for future work:

- Using different RL methods for different research topics in NTSC: There are a few studies that compared the efficiency of different methods. For instance, in (Aslani et al., 2018a), several methods are evaluated, including discrete state Q-learning(λ), discrete state SARSA(λ), discrete state actor-critic(λ), continuous state Q-learning(λ), continuous state SARSA(λ), continuous state actor-critic(λ), continuous state residual actor-critic(λ), which is a combination of the residual algorithm with actor-critic(λ). In this study, continuous state actor-critic(λ) showed the best robustness and performance. In another research, (Chu et al., 2019) used independent advantage actor critic (IA2C) where deep neural networks are employed for both policy and value approximations. Using different RL methods may provide an insight about how to improve the performance, robustness, speed, and efficiency of the proposed methods.

- Using various state, action, and reward elements in RL methods: defining and designing effective states and actions and reward functions are very important in the RL process to reach efficient results. Appendix A, Appendix B, and Table 6 help to get familiar with the options of state, action, and reward elements that exist and already used and to guide reaching new definitions of these components.

- Using and extending the idea of independent RL: In independent RL, the local agents learn their own policy independently by modeling other agents in the environment. This approach is scalable, however its convergence issue needs to be addressed, like by using Experience Replay.

- Using deep RL and hierarchical deep RL methods in large-scale networks: The efficiency of the deep RL methods and hierarchical methods has already been discussed. The first hierarchical deep RL in large-scale network was published in 2019.

- Using deep RL for arterial networks: This topic was only researched in 2019.

- Automatically achieving coordination along arterial without any prior knowledge using RL: Coordination in arterial can be achieved by using (i) the conventional coordination systems where a fixed offset among all intersections is used (Urbanik et al., 2015), (ii) the optimization-based methods (Little et al., 1981), and (iii) centralized RL-based optimization methods, which consider jointly modeling the action between learning agents (Van der Pol and Oliehoek, 2016). The methods of the last option are computationally expensive as they need to negotiate between the agents in entire network. An alternative is to use decentralized RL agents to achieve coordination. Research was conducted into this interesting area once in 2019.

- Using different function approximation (e.g. GAN) in RL methods: It is worth noting that the application of RL with function approximation started since 2010. Each function approximation has its own advantage and strength in different contexts, environment, and scale.
- Establishing an appropriate tradeoff between optimality and scalability: This is a long-run research topic that is still very important, specifically in real-world large-scale networks.
- Defining manageable state-space, action-space, and reward function: Considering the high volume of computation efforts in RL and the need for faster solutions, specifically in online learning, the necessity of defining efficient state-space, action-space, and reward function is of importance, while keeping the accuracy of the results.
- Reducing the problem space: For instance, by not blindly exploring on repeated situations, one can reduce the problem space. This was researched for the first time in 2019.
- Defining multiple reward functions for different traffic situations: Generally, a single reward function is defined for all traffic conditions; however, rewards can be defined dynamically as a response to the traffic states and multiple reward structure can be used (Ngai and Yung, 2011). This is done whether as pre-defined rewards for different time of analysis where the congestion level is known in advance (Houli et al., 2010), or the reward function can dynamically be adjusted with varying congestion states at the intersection (Aziz et al., 2018).
- Using real traffic data, realistic networks, and large-scale networks, and considering traffic heterogeneity: Real traffic are highly dynamic over time and realistic traffic network environment demands more challenges and concerns in dealing with applying RL rather than simple hypothetical setups using traffic simulations. Although there are a few research papers which recently focused on this challenge (Wei et al., 2018; Torabi et al., 2018a), there are still several challenges in dealing with the application of RL in the real-world setup. Considering traffic heterogeneity in traffic control models can also be of interest to research (Nuli and Mathew, 2013).
- Using RL and SensorGrid: The application of SensorGrid, i.e. the integration of sensor networks and grid computing, in RL started in 2007 but its usage in NTSC has never expanded since then.
- Integrating the concepts, methods, and frameworks from other fields with RL, such as JTA (for coordinated TSC problems), Pro-CEP (for predicting future system states), CPT (as a human-centered RL), Immune Network (for disturbance management): There are only 4 papers that applied RL in the methods/models/frameworks from other fields (i.e. group V3). This trend has ceased since 2016. Nevertheless, the application of other methods in RL (i.e. group V2) is still in progress. Except for the general NTSC (AD1), in other NTSC application domains we do not observe any applications of V2 to V5. This essentially motivates to propose methods based on incorporating various RL methods with the methods from other fields and optimization methods, and developing theoretical aspects of RL methods.
- Using RL in optimization problems and optimization methods in RL: Optimization algorithms, such as swarm optimization, can be integrated with RL for improvement. For instance, swarm optimization can be applied to find the optimal parameters in the reward function as a sub-problem in RL (W. Lu et al., 2011). It can also be applied to rapidly find the global optimal solution for functions with wide solution space (Tahifa et al., 2015). On the other hand, RL method can be applied in the optimization problem to reach good solutions for signal timing optimization (Ozan et al., 2015). The application of integration of optimization algorithms and RL is not very recent and not very frequently used (appeared in only 8 articles) but shows the efficiency of the integration compared to using RL or the optimization algorithm alone.
- Proposing or extending theoretical RL methods/models with feasibility assessment in NTSC: There are only 6 articles focusing on theoretical aspects of RL which has been tested in NTSC. These aspects include: (i)

learning relational policies from an expert (i.e. imitating expert) (Natarajan et al., 2011), (ii) defining a new class of multi-objective problems called CMOP (Brys et al., 2014), (iii) synthesizing a control policy for an MDP such that traces of the MDP satisfy a LTL control objective (Sadigh et al., 2014), (iv) optimizing variance-related risk measures in rewards (Prashanth and Ghavamzadeh, 2016), (v) modeling human decisions (Prashanth et al., 2016) in RL, and (vi) proposing a dynamic correlation matrix based MARL approach to reach optimal behaviours faster than other canonical learning techniques. More theoretical methods can still be proposed and examined to address the requirements of NTSC problem.

- Using RL in combination with traffic theories (e.g. CTM, MP, shock wave theory, etc): Although the usefulness of this type of combination has been shown in a few research works (Wei et al., 2019a; Chanloha et al., 2014; Ajorlou et al., 2015; Qu et al., 2020), this trend can be developed more, for instance, for shock wave theory and RL.
- Using RL as a core or combined method in integration with other methods: A part of research works employed the integration of the methods and frameworks from other fields with RL, which are listed in Table 4.
- Using RL in the integrated traffic systems/networks, including ramp metering, variable message signs, network of signalized intersections, arterials, and freeways: The integration of different traffic systems is a topic that started in 2010 and contains a few research works, yet there is room for more exploration and improvement.
- Using statistical relational techniques, such as imitation learning: This topic started in 2011, but the number of publications in this area have been very limited since then.
- Using Transfer learning in RL: It is the transfer of RL algorithms trained on simulated traffic to the real-world traffic to reduce the mistakes made in the real world. This topic is currently demanding in the area of NTSC to make RL methods well suited and deployable for a real world setup.
- Using RL in Signalized roundabouts, specifically in congested networks: The idea of applying RL for signalized roundabouts was first proposed in 2019 in 2 articles.
- Using RL in perimeter or cordon control: Although there are several publications in the area of perimeter control, RL was applied to this area only once in 2019, which shows a great opportunity to investigate new solutions in this area of research.
- Focusing on explainability when using RL: This is a very recent topic, which is explored in 2019 for the first time, where the RL agents decisions can be explained, interpretted, and verified by domain experts. There is only 1 article published on this topic.
- cloud/edge/fog-based RL: High latency communication between vehicles (or driver-less vehicles) in a connected vehicles network and limited communication bandwidth to apply the real traffic infrastructure are among the problems that NTSC still suffers from. Integrating computing paradigm such as edge and fog and RL algorithms has been shown to be a good solution for these research problems (Q. Wu et al., 2019). This topic was researched in 2019 in 3 articles, which shows a great potential and tendency towards improving the communication efficiency when using RL.
- Using RL for public transit, whether with or without public transit priority: Modeling and analysing bi-modal traffic environment, including cars and buses, are very important in urban traffic management and it is a long-lasting research area. However, there are only 2 publications in the area of RL-NTSC published in 2014 and 2019.
- Using RL for emergency vehicles: We found only one study that uses RL in the domain of emergency vehicles. The study models and simulates a TSC for autonomous vehicles at intersections, that also gives priority to

emergency vehicles. By advancement of communication between vehicles and infrastructure, controlling traffic signals based on emergency vehicles in both traditional and connected vehicles environment using RL should become a hot topic in the area of RL-NTSC.

- Using RL in image-based NTSC methods: Based on our systematic review in the area of RL-NTSC we did not find any articles that uses real images from video detection devices as a data source for the proposed RL methods. Instead, image representation and images from simulation have been used for this purpose. Image-based RL methods in NTSC will be a great avenue for research to adopt to real-world situations.

- Using RL in automating streetcar bunching control in transit routes: This interesting idea is not new (back to 2005) but it was never extended in further research works.

- Considering the use of RL in mixed environments, including regular, connected and autonomous vehicles: With the development of wireless communication, connected vehicles environments (called as vehicular ad hoc network or VANET), provides the capabilities to collect real-time traffic information for adaptive TSC. The connected vehicles technology facilitates two communication modes, including V2V and V2I, where the vehicles send vehicles identification number (ID), position, current time, speed data, and a timestamp to the intersection agents. And the intersection agents process the information, and can share it with the other neighboring intersection agents. Using RL in this type of network in NTSC is common and research in this area still continues to address several challenges, such as the difficulty of transportation modelling and optimization.

- Establishing standard benchmarks for traffic control in traffic simulators: Benchmarks helps researchers to concentrate on algorithmic improvements and control techniques rather than system and environment design. Using the benchmarks, the researchers can evaluate their results against other compared methods effectively. In different contexts, several benchmarks have been prepared to evaluate and compare RL methods, such as the Arcade Learning Environment (ALE) (Bellemare et al., 2013) for evaluating algorithms designed for tasks with high-dimensional state inputs and discrete actions, rllab (Duan et al., 2016) for tasks with partial observations and hierarchically structured tasks, and NGSIM dataset (Transportation., 2008) for microscopic data on human driving behavior. (Vinitsky et al., 2018) is the first and only article that provides new benchmarks in the use of deep RL in a micro-simulator to create controllers for mixed-autonomy traffic, where CAVs interact with human drivers and infrastructure. They characterize a set of RL algorithms by their effectiveness in training deep NN policies.

  As indicated in this study, the unexplored open questions include fairness, decentralization, generalization, and sample-efficiency. Future works can provide similar standard benchmarks in other traffic simulators and to address the abovementioned open questions in addition to considering other factors and cases (such as considering buses or integration of different methods with RL) and for other contexts in NTSC.

- online learning: This line of research in RL-NTSC started with (Choy et al., 2003) in 2003 and continued in 5 other articles (Srinivasan et al., 2006; Cai et al., 2009; Dai et al., 2011; Dusparic and Cahill, 2012; Yin et al., 2015), where the TSC agents need to continuously learn in the traffic network and update the weights and connections in the NN in real-time. The regular RL methods like Q-learning and W-learning require exploration periods while learning optimal actions and weights and make online learning almost impossible. Moreover, in the online learning process when dealing with an infinite horizon control problem, the issue of lack of stochastic exploration and the possibility of getting stuck in local minima need to be addressed (Kohonen, 2012; Yen et al., 2002). Multistage online learning processes that involve reinforcement learning, weight adjustment, and adjustment of fuzzy relations are proposed in (Choy et al., 2003; Srinivasan et al., 2006). (Cai et al., 2009) investigated two online learning techniques, including reinforcement learning and monotonicity approximation. A reinforcement training based online learning NTSC is designed in (Dai et al., 2011), which employs a feed-forward neural network. (Dusparic and Cahill, 2012) used DWL and (Yin et al.,

2015) used ADP for online learning. The online learning process helps continue learning in real-time, where the adaptation of approximation is processed online.

It is also worth noting that the classification provided in Figure 12 can help with finding the open research problems. For instance, based on this figure, there is only one article that presents the application of RL in the perimeter control. We also see that the focus of the paper is on general detection and network of intersections. Thus, different open research questions can be explored, such as extending the perimeter control for a bi-modal environment, considering bus priority, image-based traffic data detection, or perimeter control in a connected vehicle environment. This logic can also be applied for the other application domains.

Furthermore, more open research questions can be found based on a range of excluded papers. By considering NTSC within the context of the excluded papers, this becomes possible. For instance, application of RL-NTSC considering or combining with following components: bikes, pedestrians, route choice, routing systems, pedestrian routing, reactions of cyclists to speed advice, ride-sharing, best path selection, lane changing, autonomous intersection, traffic congestion detection, driver behaviour, NTSC simulation, simulators, simulation environment, online calibration, traffic assignment problems, couriers management in express systems, fleet management, toll plaza, traffic analytics, image processing, and the sensor installation locations in a traffic network.

5. Threats to Validity

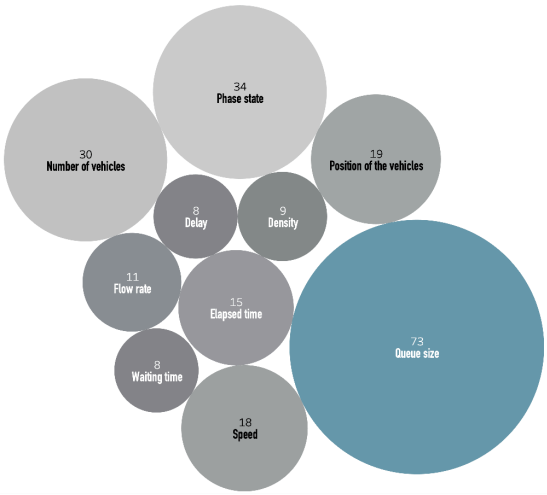
There are threats to the validity of the results and findings of our review, which will now be discussed. Although attempts were made to select our search systematically and so that they capture the existing articles in the area under investigation, part of the included articles were retrieved during the forward and backward snowballing process. This suggests that the possibility of losing part of the existing evidence is real. Moreover, research papers that may be relevant based on our criteria might have been excluded. The authors of this paper strove to collect as much relevant data as possible, and to cross reference the information for accuracy, inaccuracy remains a possibility due to the large number of research papers and features we were dealing with. Differences in our understanding, as well as the intersection of concepts with RL, such as GT, ADP, DP, and LA may have also lead to omitting some relevant research papers that the key term "reinforcement learning" in our search string could not capture.

6. Conclusion

This paper presented a comprehensive, systematic literature review on the application of RL in NTSC. The main goal of this research is to identify all eligible articles in the defined area, analyze the data of the included articles, provide statistical and conceptual knowledge based on the qualitative and descriptive data analysis, provide the highlights, variety of the applied methods, patterns, trends, frequency of existing research works in various application domains, major first events in RL-NTSC, common future directions based on what the included papers recommended, and other useful information for further research in the area of RL-NTSC. In addition to the detailed material throughout the paper, the key findings are summarized and discussed. Considering all the published review papers we uncovered in this area in our literature review, this paper covers the highest number of articles.

The review of the literature for the application of RL in single isolated intersections can be considered as an implication for future practice, which is complementary to this review paper. The integration of the results and findings of both scales can provide useful insights.

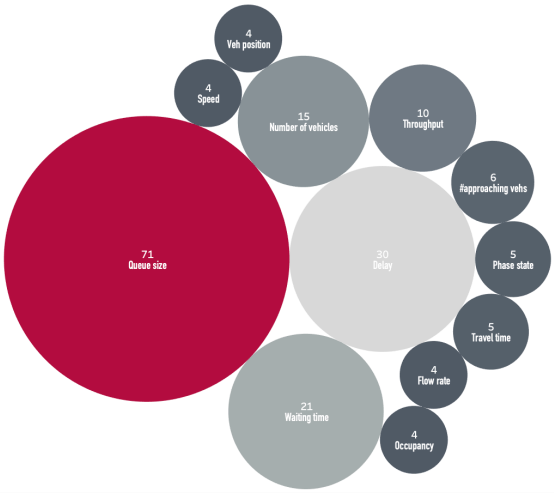
Appendix A. Elements in State definitions



ELEMENTS IN STATE DEFINITION	#	ELEMENTS IN STATE DEFINITION	#
Queue size or average queue size (in phase, lane, downstream lane, upstream lane, link, intersection, network), Congestion level or queue level (in phases, link, intersection, neighboring intersection), Pedestrian queue size (in direction), Number of waiting vehicles (in lane, link, departing link, intersection, network), Number of stopped vehicles (in cycle, link), Maximum queue size (phase, associated with each phase, associated with each direction), Saturation level in the current cycle, Congestion based on comparison between consecutive phases - no threshold is set as in congestion level (in links)	73	Distance of position of a particle with previous best position and position of its leader in Swarm	1
Current phase, Current signal state, Next phase, Phase state (in intersection, neighboring intersections), Signal control state, Index of the current green phase, Phase scenario (whether the signal group agent has to wait for other signal group agents), Index of the phase with maximum queue length for the intersection, Index of the adjacent intersection with maximum queue length, Index of the phase that has to be set green, Phase information	34	Cycle duration	1
Number of vehicles (in phase, lane, lane in neighboring links, downstream lane, link, intersection, network), Number of waiting and approaching vehicles (in link), Vehicle accumulation, Average number of vehicles per lane associated with the lanes that are/are not governed by the phase agent	30	Number of vehicles enter the network	1
Position of the vehicles (in intersection, adjacent intersections), Position of the vehicle within certain distance from the stop line, Distance to the intersection of the n nearest vehicles, Position of the vehicles in the queue, Position (in terms of segments) for the first vehicle approaching the intersection	19	The code of combination phase	1
Speed or average speed (in green signal phase lane, lane in neighboring links), Average vehicle speed (in link), Speed of the vehicles (in intersection, adjacent intersections), Velocities of the n nearest vehicles	18	Traffic light (id)	1
N/A	16	Current time of the day	1
Elapsed time (in phase, lane), Elapsed green time, The time that current phase has been green, The time since the most recent light change, The time since the second-to-last light change, The time duration since the last detection sent by the short/long detector, The time that the current phase will last	15	Presence of public transport vehicles (in link)	1
Flow rate or average flow rate (in phase, lane, link, upstream neighboring intersections), Relative flow, Flow within the previous five minutes (in lane)	11	Number of priority vehicles (in network)	1
Density (in lane, link, based on camera image of the road), Density level	9	Travel time (in link)	1
Delay (in phase, lane, link, intersection, network), Path-wise delay, Cumulative delay (of the corresponding phase, of the first vehicle), Delay of stopped vehicles	8	Vehicles' current headway	1
Waiting time or average waiting time (in phase, lane, link, intersection, network), Waiting time level (in lane), Accumulated waiting time, Waiting time of the lead vehicle	8	Number of neighbor intersections	1
Green times associated with each phase, Green time after minimum green time is passed (for current signal group, for candidate signal group), Maximum value of green times among the other signal groups in the current phase, Elapsed times since the signal turned red, Phase duration, Current phase duration, Green time duration	8	Inter-arrival times	1
Occupancy or average occupancy (in lane, link, downstream link, upstream link, intersection, network, current signal group, candidate signal group), Cell occupancies (in CTM), Relative time occupancy	7	Total stop time	1
Image-like representation (of vehicles' position, of queues), Video images	5	Acceleration of the vehicles	1
Destination of vehicles	5	Time	1
Static attributes of links, Capacity, Link indicator, Jam density	4	Vehicle information	1
Number of arriving/approaching vehicles (in link, intersection), Maximum arrivals in the green phase	4	Direction of the vehicle	1
Node where each vehicle stands at	3	The cloud control from traffic could center	1
Value of an objective function	1	Detector activation state	1
Number of incoming lanes (in intersection)	1		

Figure A.1: Elements in state definitions: (Top) The most frequently used elements in state definitions. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All elements used in state definitions (with their frequency).

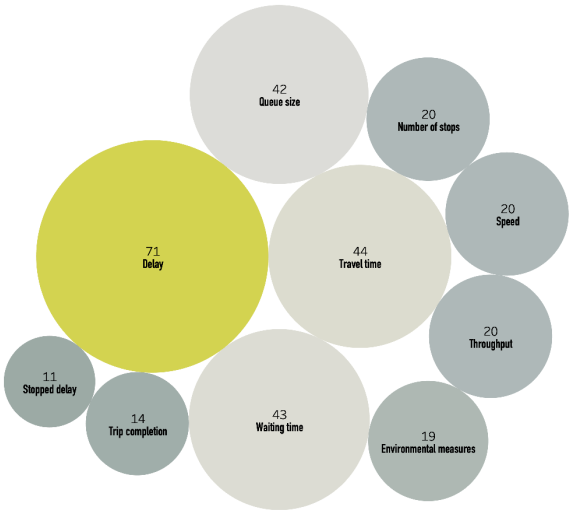
Appendix B. Elements in Reward Functions



ELEMENTS IN REWARD DEFINITION	#	ELEMENTS IN REWARD DEFINITION	#
Queue size or average queue size (in phase, lane, link, intersection, neighboring intersection), Number of waiting vehicles (in lane, link, intersection), Congestion level or average congestion level (in intersection, neighboring intersections), Pedestrian queue size (in intersection), Maximum queue size (in phase, lane, link), Number of stopped vehicles that are passing the intersection, Congestion cost, Number of stopping vehicles (in network), Number of vehicles being stopped when signal is switched from green to red, Number of stopped vehicles (in link), Congestion based on comparison between consecutive phases - no threshold is set as in congestion level (in links)	71	Total number of intersections, Number of neighboring intersections, Number of intersections that are involved in the decision about the new plan	1
delay or average delay (in link, intersection), total cumulative delay (in intersection, network), red light delay, delay of each vehicle, average delay per vehicle, average travel delay of vehicles, delay of leaving vehicles (in intersection), delay of stopped vehicles, delay incurred during the yellow-red transition	30	Presence of public transport vehicles (in link)	1
waiting time or average waiting time (in link, intersection), average waiting time level (in intersection), cumulative waiting time (in intersection), waiting time for the vehicles in cells on the links	21	The time required to travel between the upstream and downstream detectors	1
N/A	18	Number of emergency vehicles (in network)	1
Number of vehicles (in cycle, neighboring cycle, link, intersection, network), Number of vehicles in approaches receiving green, Number of vehicles in approaches receiving red	15	Capacity (link)	1
Number of vehicles passed the intersection, Throughput (in lane, link, intersection, network), Total discharge, Number of vehicles exit the network at each time step, Total volume of passing vehicles	10	Total stop time	1
Number of approaching/arriving/moving vehicles	6	The distance traveled by the vehicle in the current time step	1
Current phase, Indicator of light switches (0 or 1), Signal state, Phase state	5	Pressure (based on density)	1
Travel time, Average travel time, Total travel time of vehicles, Total travel time of vehicles passed the intersection	5	Number of of official vehicles, e.g. bus, taxi (in network)	1
Position of the vehicles	4	Vehicle information	1
Speed or average speed (in cycle, neighboring cycle, link, intersection), Speed of the vehicles	4	Number of stops	1
Flow rate (in lane, on congested link, passes through the cordon)	4	A penalty for giving green to approaches with congested receiving (downstream) links	1
Number of vehicles enter (the intersection, the network)	3	The cloud control from traffic could center	1
Occupancy or average occupancy (in lane, link, intersection, network), Occupancy ratio of waiting vehicles (in network),	4	An incentive to give green to approaches expected to receive several vehicles from upstream links	1
Number of departed vehicles (from the roundabout)	2	The importance of the vehicle information and cloud control flow	1
Whether crossing the intersection or not (vehicle level)	2	Number of vehicles within a certain distance from the intersection	1
Density or average density (of the phase in intersection), Jam density (in link)	2	Distance travelled	1
Cycle length	2	Whether the lane is on the main road or not	1
Fuel consumption, Energy consumption (in intersection)	2	Value of an objective function	1

Figure B.2: Elements in reward definitions: (Top) The most frequently used elements in reward definitions. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All elements used in reward definitions (with their frequency).

Appendix C. Performance measures



PERFORMANCE MEASURES	#	PERFORMANCE MEASURES	#
Delay, Average delay, System delay, Average delay per vehicle, Depart delay, Bus delay, Total passenger delay	71	Green time	2
Travel time, Average travel time, Pedestrian travel time, Time Through system, Person Travel Time, Time per vehicle	44	Congestion level, Congestion factor	1
Waiting time, Average waiting time, Maximum waiting time in a queue, Pedestrian waiting time, Bus Waiting time	43	Preasure	1
Queue size, Average queue size, Maximum queue size, Number of stopped vehicles, Number of waiting vehicles, Pedestrian queue size, Percent of stopped vehicles, Cross-blocking, Standard deviation of queue lengths at intersection, Stopped vehicle volume rate, Number of refused cars to enter to network in virtual queue, Maximum number of waiting vehicles in a queue	42	Distance to optimal value	1
Speed, Average speed, Harmonic speed, Distribution of speed	20	Precision-recall	1
Throughput, Number of vehicles crossed the intersection during green time, Number of passed vehicles, Passenger throughput	20	Gap between Cars, Gap time	1
Number of stops, Number of stops of buses, Trip Stops, Emercency Stops around intersection	20	Mean squared error	1
Fuel consumption,energy consumption, Emission, Air Pollution, Noise pollution	19	The time for queue dissipation after incident clearance	1
Vehicles leaving the system, Completed passenger trips, Trip completion flow, Complete Trip, Number of arrived vehicles, Total arrived road users, Completion	14	Decision time	1
Stopped delay, Stop time	11	Time loss	1
Reward, Average reward, Distribution of the discounted cumulative cost, Cumulative returns	9	Objective function value	1
N/A	9	Number of moving vehicles	1
Flow rate, Volume, Volume rate, Emergency vehicle volume rate, Maximum discharging flow	7	Arrival flow rate, Number of arriving vehicles	1
Number of vehicles (in network), Vehicles left in the system, Cumulative number of vehicles, Number of Road Users, Total number of vehicles in the coordinated lanes	5	Average lifetime of the cluster	1
Density	4	Packet delivery ratio	1
Occupancy, Occupancy Level	2	Convergence of a parameter	1
Operating cost, Communication costs	2		

Figure C.3: Performance measures and the number of their respective occurrences: (Top) The most frequently used performance measures. The numbers indicate the frequency of occurrences in the included papers, and (Bottom) All utilized performance measures (with their frequency).

Included Articles

Abdelgawad, Hossam, Baher Abdulhai, Samah El-Tantawy, Alireza Hadayeghi, and Brue Zvaniga (2015). "Assessment of self-learning adaptive traffic signal control on congested urban areas: independent versus coordinated perspectives". In: *Canadian Journal of Civil Engineering* 42.6, pp. 353–366.

Abdoos, Monireh, Nasser Mozayani, and Ana Bazzan (2011). "Traffic light control in non-stationary environments based on multi agent Q-learning". In: *2011 14th International IEEE conference on intelligent transportation systems (ITSC)*. IEEE, pp. 1580–1585.

Abdoos, Monireh, Nasser Mozayani, and Ana Bazzan (2013). "Holonic multi-agent system for traffic signals control". In: *Engineering Applications of Artificial Intelligence* 26.5-6, pp. 1575–1587.

Abdoos, Monireh, Nasser Mozayani, and Ana Bazzan (2014). "Hierarchical control of traffic signals using Q-learning with tile coding". In: *Applied intelligence* 40.2, pp. 201–213.

Abdoos, Monireh, Nasser Mozayani, and Ana Bazzan (2015). "Towards reinforcement learning for holonic multi-agent systems". In: *Intelligent Data Analysis* 19.2, pp. 211–232.

Ajorlou, Amir, Anjali Awasthi, and Amir G Aghdam (2015). "Distributed urban traffic control based on locally observable cell occupancies". In: *2015 American Control Conference (ACC)*. IEEE, pp. 1035–1040.

Al Islam, SMA Bin, HM Abdul Aziz, Hong Wang, and Stanley E Young (2018). "Minimizing energy consumption from connected signalized intersections by reinforcement learning". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1870–1875.

Araghi, Sahar, Abbas Khosravi, and Douglas Creighton (2015). "Distributed Q-learning controller for a multi-intersection traffic network". In: *International Conference on Neural Information Processing*. Springer, pp. 337–344.

Arel, Itamar, Cong Liu, Tom Urbanik, and Airton G Kohls (2010). "Reinforcement learning-based multi-agent system for network traffic signal control". In: *IET Intelligent Transport Systems* 4.2, pp. 128–135.

Aslani, Mohammad, Mohammad Saadi Mesgari, Stefan Seipel, and Marco Wiering (2019). "Developing adaptive traffic signal control by actor–critic and direct exploration methods". In: *Proceedings of the Institution of Civil Engineers-Transport*. Vol. 172. 5. Thomas Telford Ltd, pp. 289–298.

Aslani, Mohammad, Mohammad Saadi Mesgari, and Marco Wiering (2017). "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events". In: *Transportation Research Part C: Emerging Technologies* 85, pp. 732–752.

Aslani, Mohammad, Stefan Seipel, Mohammad Saadi Mesgari, and Marco Wiering (2018a). "Traffic signal optimization through discrete and continuous reinforcement learning with robustness analysis in downtown Tehran". In: *Advanced Engineering Informatics* 38, pp. 639–655.

Aslani, Mohammad, Stefan Seipel, and Marco Wiering (2018b). "Continuous residual reinforcement learning for traffic signal control optimization". In: *Canadian Journal of Civil Engineering* 45.8, pp. 690–702.

Aziz, HM Abdul, Feng Zhu, and Satish V Ukkusuri (2018). "Learning-based traffic signal control algorithms with neighborhood information sharing: An application for sustainable mobility". In: *Journal of Intelligent Transportation Systems* 22.1, pp. 40–52.

Balaji, PG, X German, and Dipti Srinivasan (2010). "Urban traffic signal control using reinforcement learning agents". In: *IET Intelligent Transport Systems* 4.3, pp. 177–188.

Bazzan, Ana, Denise De Oliveira, and Bruno C da Silva (2010). "Learning in groups of traffic signals". In: *Engineering Applications of Artificial Intelligence* 23.4, pp. 560–568.

Bazzan, Ana, Franziska Klügl, and Kai Nagel (2007). "Adaptation in games with many co-evolving agents". In: *Portuguese Conference on Artificial Intelligence*. Springer, pp. 195–206.

Bouderba, Saif Islam and Najem Moussa (2019). "Reinforcement learning (Q-LEARNING) traffic light controller within intersection traffic system". In: *Proceedings of the 4th International Conference on Big Data and Internet of Things*, pp. 1–6.

Box, Simon and Ben Waterson (2013). "An automated signalized junction controller that learns strategies by temporal difference reinforcement learning". In: *Engineering applications of artificial intelligence* 26.1, pp. 652–659.

Brys, Tim, Ann Nowé, Daniel Kudenko, and Matthew Taylor (2014). "Combining multiple correlated reward and shaping signals by measuring confidence". In: *The 28th Conference of the American Association on Artificial Intelligence: AAAI 2014*. York.

Cai, Chen, Chi Kwong Wong, and Benjamin G Heydecker (2009). "Adaptive traffic signal control using approximate dynamic programming". In: *Transportation Research Part C: Emerging Technologies* 17.5, pp. 456–474.

Camponogara, Eduardo and Werner Kraus (2003). "Distributed learning agents in urban traffic control". In: *Portuguese Conference on Artificial Intelligence*. Springer, pp. 324–335.

Cao, YJ, N Ireson, Larry Bull, and R Miles (1999). "Design of a traffic junction controller using classifier system and fuzzy logic". In: *International Conference on Computational Intelligence*. Springer, pp. 342–353.

Cao, YJ, N Ireson, Larry Bull, and R Miles (2000). "Distributed learning control of traffic signals". In: *Workshops on Real-World Applications of Evolutionary Computation*. Springer, pp. 117–126.

Chanloha, Pitipong, Jatuporn Chinrungrueng, Wipawee Usaha, and Chaodit Aswakul (2014). "Cell transmission model-based multiagent q-learning for network-scale signal control with transit priority". In: *The Computer Journal* 57.3, pp. 451–468.

Chen, Peng, Zemao Zhu, and Guangquan Lu (2019). "An Adaptive Control Method for Arterial Signal Coordination Based on Deep Reinforcement Learning". In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 3553–3558.

Chen, Yong, Juncheng Yao, Chunjiang He, Hanhua Chen, and Hai Jin (2017). "Adaptive Traffic Signal Control with Network-Wide Coordination". In: *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, pp. 180–194.

Chin, Yit Kwong, Heng Jin Tham, NSV Kameswara Rao, Nurmin Bolong, and Kenneth Tze Kin Teo (n.d.). "OPTIMIZATION OF URBAN MULTI-INTERSECTION TRAFFIC FLOW VIA Q-LEARNING". In: ().

Choy, Min Chee, Dipti Srinivasan, and Ruey Long Cheu (2003). "Cooperative, hybrid agent architecture for real-time traffic signal control". In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: systems and humans* 33.5, pp. 597–607.

Chu, Tianshu, Shuhui Qu, and Jie Wang (2016a). "Large-scale multi-agent reinforcement learning using image-based state representation". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, pp. 7592–7597.

Chu, Tianshu, Shuhui Qu, and Jie Wang (2016b). "Large-scale traffic grid signal control with regional reinforcement learning". In: *2016 American Control Conference (ACC)*. IEEE, pp. 815–820.

Chu, Tianshu and Jie Wang (2017). "Traffic signal control by distributed Reinforcement Learning with min-sum communication". In: *2017 American Control Conference (ACC)*. IEEE, pp. 5095–5100.

Chu, Tianshu, Jie Wang, Lara Codecà, and Zhaojian Li (2019). "Multi-agent deep reinforcement learning for large-scale traffic signal control". In: *IEEE Transactions on Intelligent Transportation Systems* 21.3, pp. 1086–1095.

Da Silva, Bruno C, Eduardo W Basso, Ana Bazzan, and Paulo M Engel (2006). "Dealing with non-stationary environments using context detection". In: *Proceedings of the 23rd international conference on Machine learning*, pp. 217–224.

Daeichian, Abolghasem and Amir Haghani (2018). "Fuzzy q-learning-based multi-agent system for intelligent traffic control by a game theory approach". In: *Arabian Journal for Science and Engineering* 43.6, pp. 3241–3247.

Dai, Yujie, Jinzong Hu, Dongbin Zhao, and Fenghua Zhu (2011). "Neural network based online traffic signal controller design with reinforcement training". In: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1045–1050.

Dai, Yujie, Dongbin Zhao, and Jianqiang Yi (2010). "A comparative study of urban traffic signal control with reinforcement learning and adaptive dynamic programming". In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1–7.

Darmoul, Saber, Sabeur Elkosantini, Ali Louati, and Lamjed Ben Said (2017). "Multi-agent immune networks to control interrupted flow at signalized intersections". In: *Transportation Research Part C: Emerging Technologies* 82, pp. 290–313.

Davarynejad, Mohsen, Sobhan Davarynejad, Jos Vrancken, and Jan van den Berg (2010). "Granular value-function approximation for road network traffic control". In: *2010 International Conference on Networking, Sensing and Control (ICNSC)*. IEEE, pp. 14–19.

Dowling, Jim, Raymond Cunningham, Anthony Harrington, Eoin Curran, and Vinny Cahill (2004). "Emergent consensus in decentralised systems using collaborative reinforcement learning". In: *Self-star Workshop*. Springer, pp. 63–80.

Dusparic, Ivana and Vinny Cahill (2009a). "Distributed w-learning: Multi-policy optimization in self-organizing systems". In: *2009 Third IEEE international conference on self-adaptive and self-organizing systems*. IEEE, pp. 20–29.

Dusparic, Ivana and Vinny Cahill (2009b). "Using distributed w-learning for multi-policy optimization in decentralized autonomic systems". In: *Proceedings of the 6th international conference on Autonomic computing*, pp. 63–64.

Dusparic, Ivana and Vinny Cahill (2009c). "Using Reinforcement Learning for Multi-policy Optimization in Decentralized Autonomic Systems–An Experimental Evaluation". In: *International Conference on Autonomic and Trusted Computing*. Springer, pp. 105–119.

Dusparic, Ivana and Vinny Cahill (2012). "Autonomic multi-policy optimization in pervasive systems: Overview and evaluation". In: *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 7.1, pp. 1–25.

Dusparic, Ivana, Julien Monteil, and Vinny Cahill (2016). "Towards autonomic urban traffic control with collaborative multi-policy reinforcement learning". In: *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*. IEEE, pp. 2065–2070.

El Hatri, Chaimae and Jaouad Boumhidi (2017). "Traffic management model for vehicle re-routing and traffic light control based on Multi-Objective Particle Swarm Optimization". In: *Intelligent Decision Technologies* 11.2, pp. 199–208.

Fagan, Derek and René Meier (2014). "Dynamic multi-agent reinforcement learning for control optimization". In: *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, pp. 99–104.

Gaikwad, Vinayak V, Sanket S Kadarkar, and Gaurav S Kasbekar (2016). "Intelligent traffic signal duration adaptation using q-learning with an evolving state space". In: *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE, pp. 1–6.

Gan, Xingli, Hongliang Guo, and Zhan Li (2019). "A new multi-agent reinforcement learning method based on evolving dynamic correlation matrix". In: *IEEE Access* 7, pp. 162127–162138.

Gao, Ruowen, Zhihan Liu, Jinglin Li, and Quan Yuan (2019). "Cooperative Traffic Signal Control Based on Multi-agent Reinforcement Learning". In: *International Conference on Blockchain and Trustworthy Systems*. Springer, pp. 787–793.

Ge, Hongwei, Yumei Song, Chunguo Wu, Jiankang Ren, and Guozhen Tan (2019). "Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control". In: *IEEE Access* 7, pp. 40797–40809.

Genders, Wade and Saiedeh Razavi (2020). "Policy Analysis of Adaptive Traffic Signal Control Using Reinforcement Learning". In: *Journal of Computing in Civil Engineering* 34.1, p. 04019046.

Gong, Yaobang, Mohamed Abdel-Aty, Qing Cai, and Md Sharikur Rahman (2019). "Decentralized network level adaptive signal control by multi-agent deep reinforcement learning". In: *Transportation Research Interdisciplinary Perspectives* 1, p. 100020.

Heinen, Milton R, Ana Bazzan, and Paulo M Engel (2011). "Dealing with continuous-state reinforcement learning for intelligent control of traffic signals". In: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 890–895.

Higuera, Carolina, Fernando Lozano, Edgar Camilo Camacho, and Carlos Hernando Higuera (2019). "Multiagent Reinforcement Learning Applied to Traffic Light Signal Control". In: *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, pp. 115–126.

Horsuwan, Thanapapas and Chaodit Aswakul (2019). "Reinforcement Learning Agent under Partial Observability for Traffic Light Control in Presence of Gridlocks." In: *SUMO*, pp. 29–47.

Houli, Duan, Li Zhiheng, and Zhang Yi (2010). "Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network". In: *EURASIP journal on advances in signal processing* 2010.1, p. 724035.

Huang, Rui, Jianming Hu, Yusen Huo, and Xin Pei (2019). "Cooperative Multi-Intersection Traffic Signal Control Based on Deep Reinforcement Learning". In: *CICTP 2019*, pp. 2959–2970.

Iyer, Vignesh, Rashmi Jadhav, Unnati Mavchi, and Jibi Abraham (2016). "Intelligent traffic signal synchronization using fuzzy logic and Q-learning". In: *2016 International Conference on Computing, Analytics and Security Trends (CAST)*. IEEE, pp. 156–161.

Jadhao, Ms Namrata S and Mr Parag A Kulkarni (2012). "Reinforcement Learning Based for Traffic Signal Monitoring and Management". In: *International Journal of Engineering Research and Technology* 1.

Jadhao, Namrata S and Ashish S Jadhao (2014). "Traffic Signal Control Using Reinforcement Learning". In: *2014 Fourth International Conference on Communication Systems and Network Technologies*. IEEE, pp. 1130–1135.

Jin, Junchen and Xiaoliang Ma (2017). "A multi-objective multi-agent framework for traffic light control". In: *2017 11th Asian Control Conference (ASCC)*. IEEE, pp. 1199–1204.

Jin, Junchen and Xiaoliang Ma (2018). "Hierarchical multi-agent control of traffic lights based on collective learning". In: *Engineering applications of artificial intelligence* 68, pp. 236–248.

Jin, Junchen and Xiaoliang Ma (2019). "A multi-objective agent-based control approach with application in intelligent traffic signal system". In: *IEEE Transactions on Intelligent Transportation Systems* 20.10, pp. 3900–3912.

Kao, Ying-Cih and Cheng-Wen Wu (2018). "A Self-Organizing Map-Based Adaptive Traffic Light Control System with Reinforcement Learning". In: *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, pp. 2060–2064.

Khamis, Mohamed A and Walid Gomaa (2012). "Enhanced multiagent multi-objective reinforcement learning for urban traffic light control". In: *2012 11th International Conference on Machine Learning and Applications*. Vol. 1. IEEE, pp. 586–591.

Khamis, Mohamed A and Walid Gomaa (2014). "Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework". In: *Engineering Applications of Artificial Intelligence* 29, pp. 134–151.

Khamis, Mohamed A, Walid Gomaa, and Hisham El-Shishiny (2012). "Multi-objective traffic light control system based on Bayesian probability interpretation". In: *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 995–1000.

Kim, Daeho and Okran Jeong (2020). "Cooperative traffic signal control with traffic flow prediction in multi-intersection". In: *Sensors* 20.1, p. 137.

Kitagawa, Shunya, Ahmed Moustafa, and Takayuki Ito (2019). "Urban Traffic Control Using Distributed Multi-agent Deep Reinforcement Learning". In: *Pacific Rim International Conference on Artificial Intelligence*. Springer, pp. 337–349.

Kristensen, Terje and Nnamdi Johnson Ezeora (2017). "Simulation of intelligent traffic control for autonomous vehicles". In: *2017 IEEE International Conference on Information and Automation (ICIA)*. IEEE, pp. 459–465.

Kuyer, Lior, Shimon Whiteson, Bram Bakker, and Nikos Vlassis (2008). "Multiagent reinforcement learning for urban traffic control using coordination graphs". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 656–671.

Lee, Jincheol, Jiyong Chung, and Keemin Sohn (2020). "Reinforcement learning for joint control of traffic signals in a transportation network". In: *IEEE Transactions on Vehicular Technology* 69.2, pp. 1375–1387.

Lemos, Liza L, Ana Bazzan, and Márcia Pasin (2018). "Co-adaptive reinforcement learning in microscopic traffic systems". In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, pp. 1–8.

Li, Chun-gui, Xiang-lei Yan, Fei-Ying Lin, and Hong-lei Zhang (2011). "Multi-intersections traffic signal intelligent control using collaborative q-learning algorithm". In: *2011 Seventh International Conference on Natural Computation*. Vol. 1. IEEE, pp. 185–188.

Li, Congcong, Fei Yan, Yiduo Zhou, Jia Wu, and Xiaomin Wang (2018). "A Regional Traffic Signal Control Strategy with Deep Reinforcement Learning". In: *2018 37th Chinese Control Conference (CCC)*. IEEE, pp. 7690–7695.

Li, Tao, Dongbin Zhao, and Jianqiang Yi (2008a). "Adaptive dynamic neuro-fuzzy system for traffic signal control". In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, pp. 1840–1846.

Li, Tao, Dongbin Zhao, and Jianqiang Yi (2008b). "Adaptive dynamic programming for multi-intersections traffic signal intelligent control". In: *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 286–291.

Ling, Kenny and Amer S Shalaby (2005). "A reinforcement learning approach to streetcar bunching control". In: *Journal of Intelligent Transportation Systems* 9.2, pp. 59–68.

Liu, Weirong, Jing Liu, Jun Peng, and Zhengfa Zhu (2014). "Cooperative multi-agent traffic signal control system using fast gradient-descent function approximation for V2I networks". In: *2014 IEEE International Conference on Communications (ICC)*. IEEE, pp. 2562–2567.

Liu, Weirong, Gaorong Qin, Yun He, and Fei Jiang (2017). "Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering". In: *IEEE transactions on vehicular technology* 66.10, pp. 8667–8681.

Liu, Xiao-Yang, Zihan Ding, Sem Borst, and Anwar Walid (2018). "Deep reinforcement learning for intelligent transportation systems". In: *arXiv preprint arXiv:1812.00979*.

Liu, Ying, Lei Liu, and Wei-Peng Chen (2017). "Intelligent traffic light control using distributed multi-agent Q learning". In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1–8.

Lu, Chenqing, Feng Wen, and Mitsuo Gen (2017). "Traffic Lights Dynamic Timing Algorithm Based on Reinforcement Learning". In: *International Conference on Management Science and Engineering Management*. Springer, pp. 1752–1761.

Lu, Shoufeng, Ximin Liu, and Shiqiang Dai (2008). "Adaptive and coordinated traffic signal control based on Q-learning and multiband model". In: *2008 IEEE Conference on Cybernetics and Intelligent Systems*. IEEE, pp. 765–770.

Lu, Wei, Yunlong Zhang, and Yuanchang Xie (2011). "A multi-agent adaptive traffic signal control system using swarm intelligence and neuro-fuzzy reinforcement learning". In: *2011 IEEE Forum on Integrated and Sustainable Transportation Systems*. IEEE, pp. 233–238.

Marsetič, Rok, Darja Šemrov, and Marijan Žura (2014). "Road artery traffic light optimization with use of the reinforcement learning". In: *Promet-Traffic&Transportation* 26.2, pp. 101–108.

Mashayekhi, Mehdi and George List (2015). "A multiagent auction-based approach for modeling of signalized intersections". In: *IJCAI Workshops on Synergies Between Multiagent Systems, Machine Learning and Complex Systems*, pp. 13–24.

Medina, Juan C and Rahim F Benekohal (2012). "Traffic signal control using reinforcement learning and the max-plus algorithm as a coordinating strategy". In: *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 596–601.

Medina, Juan C, Ali Hajbabaie, and Rahim F Benekohal (2010). "Arterial traffic control using reinforcement learning agents and information from adjacent intersections in the state and reward structure". In: *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 525–530.

Mikami, Sadayoshi and Yukinori Kakazu (1994). "Genetic reinforcement learning for cooperative traffic signal control". In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. IEEE, pp. 223–228.

Moghadam, Mahshid Helali and Nasser Mozayani (2013). "Urban Traffic Control Using Adjusted Reinforcement Learning in a Multi-agent System". In: *Research Journal of Applied Sciences, Engineering and Technology* 6.16, pp. 2943–2950.

Natarajan, Sriraam, Saket Joshi, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik (2011). "Imitation learning in relational domains: A functional-gradient boosting approach". In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. Vol. 22. 1. Citeseer, p. 1414.

Natarajan, Sriraam, Gautam Kunapuli, Kshitij Judah, Prasad Tadepalli, Kristian Kersting, and Jude Shavlik (2010). "Multi-agent inverse reinforcement learning". In: *2010 Ninth International Conference on Machine Learning and Applications*. IEEE, pp. 395–400.

Ni, Wei and Michael J Cassidy (2019). "Cordon control with spatially-varying metering rates: A Reinforcement Learning approach". In: *Transportation Research Part C: Emerging Technologies* 98, pp. 358–369.

Nishi, Tomoki, Keisuke Otaki, Keiichiro Hayakawa, and Takayoshi Yoshimura (2018). "Traffic signal control based on reinforcement learning with graph convolutional neural nets". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 877–883.

Nuli, Sadguna and Tom V Mathew (2013). "Online coordination of signals for heterogeneous traffic using stop line detection". In: *Procedia-Social and Behavioral Sciences* 104, pp. 765–774.

Oliveira, Denise de, Ana Bazzan, Bruno Castro da Silva, Eduardo W Basso, Luis Nunes, Rosaldo Rossetti, Eugénio de Oliveira, Roberto da Silva, and Luis Lamb (2006). "Reinforcement Learning based Control of Traffic Lights in Non-stationary Environments: A Case Study in a Microscopic Simulator." In: *EUMAS*.

Ozan, Cenk, Ozgur Baskan, Soner Haldenbilen, and Halim Ceylan (2015). "A modified reinforcement learning algorithm for solving coordinated signalized networks". In: *Transportation Research Part C: Emerging Technologies* 54, pp. 40–55.

Pham, Tong Thanh, Tim Brys, Matthew E Taylor, Tim Brys, Madalina M Drugan, PA Bosman, Martine-De Cock, Cosmin Lazar, L Demarchi, David Steenhoff, et al. (2013). "Learning coordinated traffic light control". In: *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS-13)*. Vol. 10. IEEE, pp. 1196–1201.

Prabuchandran, KJ, Hemanth Kumar AN, and Shalabh Bhatnagar (2014). "Multi-agent reinforcement learning for traffic signal control". In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 2529–2534.

Prabuchandran, KJ, Hemanth Kumar AN, and Shalabh Bhatnagar (2015). "Decentralized learning for traffic signal control". In: *2015 7th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, pp. 1–6.

Prashanth, LA and Shalabh Bhatnagar (2010). "Reinforcement learning with function approximation for traffic signal control". In: *IEEE Transactions on Intelligent Transportation Systems* 12.2, pp. 412–421.

Prashanth, LA and Shalabh Bhatnagar (2011). "Reinforcement learning with average cost for adaptive control of traffic lights at intersections". In: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1640–1645.

Prashanth, LA and Mohammad Ghavamzadeh (2016). "Variance-constrained actor-critic algorithms for discounted and average reward MDPs". In: *Machine Learning* 105.3, pp. 367–417.

Prashanth, LA, Cheng Jie, Michael Fu, Steve Marcus, and Csaba Szepesvári (2016). "Cumulative prospect theory meets reinforcement learning: Prediction and control". In: *International Conference on Machine Learning*, pp. 1406–1415.

Prothmann, Holger, Jurgen Branke, Hartmut Schmeck, Sven Tomforde, Fabian Rochner, Jorg Hahner, and Christian Muller-Schloer (2009). "Organic traffic light control for urban road networks". In: *International Journal of Autonomous and Adaptive Communications Systems* 2.3, pp. 203–225.

Qu, Zhaowei, Zhaotian Pan, Yongheng Chen, Xin Wang, and Haitao Li (2020). "A Distributed Control Method for Urban Networks Using Multi-Agent Reinforcement Learning Based on Regional Mixed Strategy Nash-Equilibrium". In: *IEEE Access* 8, pp. 19750–19766.

Reda, Mali, Fouad Mountassir, and Bousmah Mohamed (2019). "Introduction to Coordinated Deep Agents for Traffic Signal". In: *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*. IEEE, pp. 1–6.

Richter, Silvia, Douglas Aberdeen, and Jin Yu (2007). "Natural actor-critic for road traffic optimisation". In: *Advances in neural information processing systems*, pp. 1169–1176.

Ritcher, S (2007). “Traffic light scheduling using policy-gradient reinforcement learning”. In: *The International Conference on Automated Planning and Scheduling., ICAPS*.

Rizzo, Stefano Giovanni, Giovanna Vantini, and Sanjay Chawla (2019a). “Reinforcement Learning with Explainability for Traffic Signal Control”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 3567–3572.

Rizzo, Stefano Giovanni, Giovanna Vantini, and Sanjay Chawla (2019b). “Time Critic Policy Gradient Methods for Traffic Signal Control in Complex and Congested Scenarios”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1654–1664.

Rosyadi, Andhika Rizky, Tjokorda Agung Budi Wirayuda, and Said Al-Faraby (2016). “Intelligent traffic light control using collaborative Q-Learning algorithms”. In: *2016 4th International Conference on Information and Communication Technology (ICoICT)*. IEEE, pp. 1–6.

Sadigh, Dorsa, Eric S Kim, Samuel Coogan, S Shankar Sastry, and Sanjit A Seshia (2014). “A learning based approach to control synthesis of markov decision processes for linear temporal logic specifications”. In: *53rd IEEE Conference on Decision and Control*. IEEE, pp. 1091–1096.

Salkham, As’ ad and Vinny Cahill (2010). “Soilse: A decentralized approach to optimization of fluctuating urban traffic using reinforcement learning”. In: *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, pp. 531–538.

Salkham, As’ ad, Raymond Cunningham, Anurag Garg, and Vinny Cahill (2008). “A collaborative reinforcement learning approach to urban traffic control optimization”. In: *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. Vol. 2. IEEE, pp. 560–566.

Shabestray, Soheil Mohamad Alizadeh and Baher Abdulhai (2019). “Multimodal iNtelligent Deep (MiND) Traffic Signal Controller”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 4532–4539.

Shen, Mengjia (2016). “A fast method to prevent traffic blockage by signal control based on reinforcement learning”. In: *International Conference on Communication and Electronic Information Engineering (CEIE 2016)*. Atlantis Press.

Shi, Saijiang and Feng Chen (2018). “Deep Recurrent Q-learning Method for Area Traffic Coordination Control”. In: *Journal of Advances in Mathematics and Computer Science*, pp. 1–11.

Shu, Lingzhou, Jia Wu, and Ziyan Li (2019). “Hierarchical Regional Control for Traffic Grid Signal Optimization”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 3547–3552.

Song, Jiong, Zhao Jin, and WenJun Zhu (2011). “Implementing traffic signal optimal control by multiagent reinforcement learning”. In: *Proceedings of 2011 International Conference on Computer Science and Network Technology*. Vol. 4. IEEE, pp. 2578–2582.

Srinivasan, Dipti, Min Chee Choy, and Ruey Long Cheu (2006). “Neural networks for real-time traffic signal control”. In: *IEEE Transactions on intelligent transportation systems* 7.3, pp. 261–272.

Su, Shiyan and Chen-Khong Tham (2007). “SensorGrid for real-time traffic management”. In: *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. IEEE, pp. 443–448.

Tahifa, Mohammed, Jaouad Boumhidi, and Ali Yahyaouy (2015). “Swarm reinforcement learning for traffic signal control based on cooperative multi-agent framework”. In: *2015 Intelligent Systems and Computer Vision (ISCV)*. IEEE, pp. 1–6.

Tan, Tian, Feng Bao, Yue Deng, Alex Jin, Qionghai Dai, and Jie Wang (2019). “Cooperative deep reinforcement learning for large-scale traffic grid signal control”. In: *IEEE transactions on cybernetics* 50.6, pp. 2687–2700.

El-Tantawy, Samah and Baher Abdulhai (2010). “Towards multi-agent reinforcement learning for integrated network of optimal traffic controllers (MARLIN-OTC)”. In: *Transportation Letters* 2.2, pp. 89–110.

El-Tantawy, Samah, Baher Abdulhai, and Hossam Abdelgawad (2013). “Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): methodology and large-scale application on downtown Toronto”. In: *IEEE Transactions on Intelligent Transportation Systems* 14.3, pp. 1140–1150.

El-Tantawy, Samah, Baher Abdulhai, and Hossam Abdelgawad (2014). “Design of reinforcement learning parameters for seamless application of adaptive traffic signal control”. In: *Journal of Intelligent Transportation Systems* 18.3, pp. 227–245.

El-Tantawy, Samah, Kasra Rezaee, and Baher Abdulhai (2015). “Closed loop optimal adaptive traffic signal and ramp control: A case study on downtown Toronto”. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, pp. 2398–2404.

Teo, Kenneth Tze Kin, Kiam Beng Yeo, Yit Kwong Chin, Helen Sin Ee Chuo, and Min Keng Tan (2014). “Agent-based optimization for multiple signalized intersections using Q-learning”. In: *International Journal of Simulation: Systems, Science & Technology* 15.6, pp. 90–96.

Torabi, Behnam, Rym Z Wenkstern, and Robert Saylor (2018a). “A Self-Adaptive Collaborative Multi-Agent based Traffic Signal Timing System”. In: *2018 IEEE International Smart Cities Conference (ISC2)*. IEEE, pp. 1–8.

Vidhate, Deepak A and Parag Kulkarni (2017). “Exploring Cooperative Multi-agent Reinforcement Learning Algorithm (CMRLA) for Intelligent Traffic Signal Control”. In: *International Conference on Smart Trends for Information Technology and Computer Communications*. Springer, pp. 71–81.

Vinitsky, Eugene, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M Bayen (2018). “Benchmarks for reinforcement learning in mixed-autonomy traffic”. In: *Conference on Robot Learning*, pp. 399–409.

Wang, Yizhe, Xiaoguang Yang, Yangdong Liu, and Hailun Liang (2018b). “Evaluation and Application of Urban Traffic Signal Optimizing Control Strategy Based on Reinforcement Learning”. In: *Journal of Advanced Transportation*.

Wang, Yongheng, Shaofeng Geng, and Qian Li (2016). "Intelligent Transportation Control based on Proactive Complex Event Processing". In: *MATEC Web of Conferences*. Vol. 77. EDP Sciences, p. 09004.

Waskow, Samuel Justo and Ana Bazzan (2010). "Improving space representation in multiagent learning via tile coding". In: *Brazilian Symposium on Artificial Intelligence*. Springer, pp. 153–162.

Wei, Hua, Chacha Chen, Guanjie Zheng, Kan Wu, Vikash Gayah, Kai Xu, and Zhenhui Li (2019a). "Presslight: Learning max pressure control to coordinate traffic signals in arterial network". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1290–1298.

Wei, Hua, Nan Xu, Huichu Zhang, Guanjie Zheng, Xinshi Zang, Chacha Chen, Weinan Zhang, Yanmin Zhu, Kai Xu, and Zhenhui Li (2019b). "Colight: Learning network-level cooperation for traffic signal control". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1913–1922.

Wei, Hua, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li (2018). "Intellilight: A reinforcement learning approach for intelligent traffic light control". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2496–2505.

Wiering, Marco, Jilles Vreeken, Jelle Van Veenen, and Arne Koopman (2004). "Simulation and optimization of traffic in a city". In: *IEEE Intelligent Vehicles Symposium, 2004*. IEEE, pp. 453–458.

Wiering, Marco A (2000). "Multi-agent reinforcement learning for traffic light control". In: *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pp. 1151–1158.

Wu, Qiang, Jun Shen, Binbin Yong, Jianqing Wu, Fucun Li, Jinqiang Wang, and Qingguo Zhou (2019). "Smart fog based workflow for traffic control networks". In: *Future Generation Computer Systems* 97, pp. 825–835.

Wu, Wei, Gong Shufeng, and Liu Hongxiu (2009). "A coordinated urban traffic signal control approach based on multi-agent". In: *2009 International Conference on Intelligent Engineering Systems*. IEEE, pp. 263–267.

Xiang, Junping and Zonghai Chen (2015). "Adaptive Traffic Signal Control of Bottleneck Subzone based on Grey Qualitative Reinforcement Learning Algorithm." In: *ICPRAM (2)*, pp. 295–301.

Xinhai, Xia and Xu Lunhui (2009). "Traffic signal control agent interaction model based on game theory and reinforcement learning". In: *2009 International Forum on Computer Science-Technology and Applications*. Vol. 1. IEEE, pp. 164–168.

Xu, Lun-Hui, Xin-Hai Xia, and Qiang Luo (2013). "The study of reinforcement learning for traffic self-adaptive control under multiagent markov game environment". In: *Mathematical Problems in Engineering* 2013.

Xu, Ming, Jianping Wu, Ling Huang, Rui Zhou, Tian Wang, and Dongmei Hu (2020). "Network-wide traffic signal control based on the discovery of critical nodes and deep reinforcement learning". In: *Journal of Intelligent Transportation Systems* 24.1, pp. 1–10.

Xu, Nan, Guanjie Zheng, Kai Xu, Yanmin Zhu, and Zhenhui Li (2019). "Targeted knowledge transfer for learning traffic signal plans". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, pp. 175–187.

Xu, Wuxiong, Dong Zhong, Siqing Wu, and Hao Ni (2015). "A Control Method of Traffic flow Based on Region Coordination". In: *2015 International Conference on Management, Education, Information and Control*. Atlantis Press.

Yang, Shantian, Bo Yang, Hau-San Wong, and Zhongfeng Kang (2019). "Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm". In: *Knowledge-Based Systems* 183, p. 104855.

Yin, Biao, Mahjoub Dridi, and Abdellah El Moudni (2015). "Adaptive traffic signal control for multi-intersection based on microscopic model". In: *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, pp. 49–55.

Yin, Biao, Mahjoub Dridi, and Abdellah El Moudni (2016). "Traffic network micro-simulation model and control algorithm based on approximate dynamic programming". In: *IET Intelligent Transport Systems* 10.3, pp. 186–196.

Zhang, Xinhua, Douglas Aberdeen, and SVN Vishwanathan (2007). "Conditional random fields for multi-agent reinforcement learning". In: *Proceedings of the 24th international conference on Machine learning*, pp. 1143–1150.

Zhao, Xiaohua, Zhenlong Li, Quan Yu, and Yanzhang Shang (2009). "A Study of the Cooperation Control of Two Adjacent Intersections Based on NBS Game Q-Learning Algorithm". In: *2009 Fifth International Conference on Natural Computation*. Vol. 6. IEEE, pp. 551–557.

Zhao, Yi, Jianxiao Ma, Linghong Shen, and Yong Qian (2020). "Optimizing the Junction-Tree-Based Reinforcement Learning Algorithm for Network-Wide Signal Coordination". In: *Journal of Advanced Transportation* 2020.

Zheng, Guanjie, Yuanhao Xiong, Xinshi Zang, Jie Feng, Hua Wei, Huichu Zhang, Yong Li, Kai Xu, and Zhenhui Li (2019a). "Learning phase competition for traffic signal control". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1963–1972.

Zhou, Pengyuan, Tristan Braud, Ahmad Alhilal, Pan Hui, and Jussi Kangasharju (2019). "ERL: Edge based Reinforcement Learning for optimized urban Traffic light control". In: *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, pp. 849–854.

Zhu, Feng, HM Abdul Aziz, Xinwu Qian, and Satish V Ukkusuri (2015). "A junction-tree based learning algorithm to optimize network wide traffic control: A coordinated multi-agent framework". In: *Transportation Research Part C: Emerging Technologies* 58, pp. 487–501.

References

Abad, Zahra Shakeri Hossein, Vincenzo Gervasi, Didar Zowghi, and Behrouz H Far (2019). "Supporting analysts by dynamic extraction and classification of requirements-related knowledge". In: *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, pp. 442–453.

Abad, Zahra Shakeri Hossein, Mohammad Noaeen, and Guenther Ruhe (2016). "Requirements engineering visualization: a systematic literature review". In: *2016 IEEE 24th International Requirements Engineering Conference (RE)*. IEEE, pp. 6–15.

Abdulhai, Baher and Lina Kattan (2003). "Reinforcement learning: Introduction to theory and potential for transport applications". In: *Canadian Journal of Civil Engineering* 30.6, pp. 981–991.

Abdulhai, Baher, Rob Pringle, and Grigoris J Karakoulas (2003). "Reinforcement learning for true adaptive traffic signal control". In: *Journal of Transportation Engineering* 129.3, pp. 278–285.

Anderson, Henry Junior and Samuel Amponsah Odei (2018). "The influence of public support on university-industry-government collaboration: the case of the Czech Republic, Slovakia, Hungary and Romania". In: *Statistika, volume 98, issue: 4*.

Bagnell, J Andrew, Nathan Ratliff, and Martin Zinkevich (2006). "Maximum margin planning". In: *Proceedings of the International Conference on Machine Learning (ICML)*. Citeseer.

Baird III, Leemon C (1999). *Reinforcement learning through gradient descent*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE.

Bakker, Bram, M Steingrover, Roelant Schouten, EHJ Nijhuis, LJHM Kester, et al. (2005). "Cooperative multi-agent reinforcement learning of traffic lights". In:

Baldazo, David, Juan Parras, and Santiago Zazo (2019). "Decentralized Multi-Agent deep reinforcement learning in swarms of drones for flood monitoring". In: *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 1–5.

Bazzan, Ana (2009). "Opportunities for multiagent systems and multiagent reinforcement learning in traffic control". In: *Autonomous Agents and Multi-Agent Systems* 18.3, p. 342.

Bazzan, Ana and Franziska Klügl (2014). "A review on agent-based technology for traffic and transportation". In: *The Knowledge Engineering Review* 29.3, pp. 375–403.

Bellemare, Marc G, Yavar Naddaf, Joel Veness, and Michael Bowling (2013). "The arcade learning environment: An evaluation platform for general agents". In: *Journal of Artificial Intelligence Research* 47, pp. 253–279.

Blockeel, Hendrik and Luc De Raedt (1998). "Top-down induction of first-order logical decision trees". In: *Artificial intelligence* 101.1-2, pp. 285–297.

Butz, Martin V, David E Goldberg, and Pier Luca Lanzi (2005). "Computational complexity of the XCS classifier system". In: *Foundations of Learning Classifier Systems*. Springer, pp. 91–125.

Central Intelligence Agency, Country Comparison Roadways (2020). URL: https://www.cia.gov/the-world-factbook/field/roadways/country-comparison.

Choy, Min Chee, Dipti Srinivasan, and Ruey Long Cheu (2006). "Neural networks for continuous online learning and control". In: *IEEE Transactions on Neural Networks* 17.6, pp. 1511–1531.

Claus, Caroline and Craig Boutilier (1998). "The dynamics of reinforcement learning in cooperative multiagent systems". In: *AAAI/IAAI* 1998.746-752, p. 2.

Cools, Seung-Bae, Carlos Gershenson, and Bart D'Hooghe (2013). "Self-organizing traffic lights: A realistic simulation". In: *Advances in applied self-organizing systems*. Springer, pp. 45–55.

Daeinabi, Ameneh, Akbar Ghaffar Pour Rahbar, and Ahmad Khademzadeh (2011). "VWCA: An efficient clustering algorithm in vehicular ad hoc networks". In: *Journal of Network and Computer Applications* 34.1, pp. 207–222.

Diakaki, Christina, Markos Papageorgiou, and Kostas Aboudolas (2002). "A multivariable regulator approach to traffic-responsive network-wide signal control". In: *Control Engineering Practice* 10.2, pp. 183–195.

Dietterich, Thomas G, Adam Ashenfelter, and Yaroslav Bulatov (2004). "Training conditional random fields via gradient tree boosting". In: *Proceedings of the twenty-first international conference on Machine learning*, p. 28.

Dresner, Kurt and Peter Stone (2008). "A multiagent approach to autonomous intersection management". In: *Journal of artificial intelligence research* 31, pp. 591–656.

Duan, Yan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel (2016). "Benchmarking deep reinforcement learning for continuous control". In: *International conference on machine learning*. PMLR, pp. 1329–1338.

Eom, Myungeun and Byung-In Kim (2020). "The traffic signal control problem for intersections: a review". In: *European Transport Research Review* 12.1, pp. 1–20.

Fink, Arlene (2019). *Conducting research literature reviews: From the internet to paper*. Sage publications.

Gao, Juntao, Yulong Shen, Jia Liu, Minoru Ito, and Norio Shiratori (2017). "Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network". In: *arXiv preprint arXiv:1705.02755*.

Gershenson, Carlos (2004). "Self-organizing traffic lights". In: *arXiv preprint nlin/0411066*.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio (2016). *Deep learning*. Vol. 1. 2. MIT press Cambridge.

Greenhalgh, Trisha and Richard Peacock (2005). "Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources". In: *Bmj* 331.7524, pp. 1064–1065.

Gregurić, Martin, Miroslav Vujić, Charalampos Alexopoulos, and Mladen Miletić (2020). "Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data". In: *Applied Sciences* 10.11, p. 4011.

Haydari, Ammar and Yasin Yilmaz (2020). "Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey". In: *arXiv preprint arXiv:2005.00935*.
Humphrys, Mark (1996). "Action selection methods using reinforcement learning". In: *From Animals to Animats* 4, pp. 135–144.
Hunt, PB, DI Robertson, RD Bretherton, and RI Winton (1981). *SCOOT-a traffic responsive method of coordinating signals*. Tech. rep.
Hüttenrauch, Maximilian, Sosic Adrian, Gerhard Neumann, et al. (2019). "Deep reinforcement learning for swarm systems". In: *Journal of Machine Learning Research* 20.54, pp. 1–31.
Index, Traffic (2014). In: URL: https://www.numbeo.com/traffic/rankings_by_country.jsp?title=2020-mid.
Inrix (Mar. 2020). *Scorecard*. URL: https://inrix.com/scorecard/.
Jácome, Luis, Leonardo Benavides, Diejo Jara, Gonzalo Riofrio, Fabricio Alvarado, and Manuel Pesantez (2018). "A Survey on Intelligent Traffic Lights". In: *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*. IEEE, pp. 1–6.
Keong, Chin Kian (1993). "The GLIDE system—Singapore's urban traffic control system". In: *Transport reviews* 13.4, pp. 295–305.
Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.
Kohonen, Teuvo (2012). *Self-organizing maps*. Vol. 30. Springer Science & Business Media.
Konda, Vijaymohan R and Vivek S Borkar (1999). "Actor-Critic–Type Learning Algorithms for Markov Decision Processes". In: *SIAM Journal on control and Optimization* 38.1, pp. 94–123.
Koonce, Peter and Lee Rodegerdts (2008). *Traffic signal timing manual*. Tech. rep. United States. Federal Highway Administration.
Lämmer, Stefan and Dirk Helbing (2008). "Self-control of traffic lights and vehicle flows in urban road networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.04, P04019.
Liang, Xiaoyuan, Tan Yan, Joyoung Lee, and Guiling Wang (2018). "A distributed intersection management protocol for safety, efficiency, and driver's comfort". In: *IEEE internet of things journal* 5.3, pp. 1924–1935.
Lillicrap, Timothy P, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra (2015). "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971*.
Little, John DC, Mark D Kelson, and Nathan H Gartner (1981). "MAXBAND: A versatile program for setting signals on arteries and triangular networks". In:
Liu, Zhiyong (2007). "A survey of intelligence methods in urban traffic signal control". In: *IJCSNS International Journal of Computer Science and Network Security* 7.7, pp. 105–112.
Mannion, Patrick, Jim Duggan, and Enda Howley (2016). "An experimental review of reinforcement learning algorithms for adaptive traffic signal control". In: *Autonomic road transport support systems*. Springer, pp. 47–66.
Miller, Alan J (1963). "Settings for fixed-cycle traffic signals". In: *Journal of the Operational Research Society* 14.4, pp. 373–386.
Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller (2013). "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602*.
Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. (2015). "Human-level control through deep reinforcement learning". In: *nature* 518.7540, pp. 529–533.
Ng, Andrew Y, Stuart J Russell, et al. (2000). "Algorithms for inverse reinforcement learning." In: *Icml*. Vol. 1, p. 2.
Ngai, Daniel Chi Kit and Nelson Hon Ching Yung (2011). "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers". In: *IEEE Transactions on Intelligent Transportation Systems* 12.2, pp. 509–522.
Nguyen, Hoang, Le-Minh Kieu, Tao Wen, and Chen Cai (2018). "Deep learning methods in transportation domain: a review". In: *IET Intelligent Transport Systems* 12.9, pp. 998–1004.
Okoli, Chitu and Kira Schabram (2010). "A guide to conducting a systematic literature review of information systems research". In:
OroojlooyJadid, Afshin and Davood Hajinezhad (2019). "A review of cooperative multi-agent deep reinforcement learning". In: *arXiv preprint arXiv:1908.03963*.
Richter, Silvia et al. (2006). "Learning road traffic control: towards practical traffic control using policy gradients". In:
Riedmiller, Martin (2005). "Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method". In: *European Conference on Machine Learning*. Springer, pp. 317–328.
Robertson, Dennis I (1969). "TRANSYT: a traffic network study tool". In:
Schutera, Mark, Niklas Goby, Stefan Smolarek, and Markus Reischl (2018). "Distributed traffic light control at uncoupled intersections with real-world topology by deep reinforcement learning". In: *arXiv preprint arXiv:1811.11233*.
Silva, Bruno Castro da, Ana Bazzan, Gustavo K Andriotti, Filipe Lopes, and Denise de Oliveira (2004). "ITSUMO: an intelligent transportation system for urban mobility". In: *International Workshop on Innovative Internet Community Systems*. Springer, pp. 224–235.
Sims, AG et al. (1981). "Scat the sydney co-ordinated adaptive traffic system". In: *Symposium on Computer Control of Transport 1981: Preprints of Papers*. Institution of Engineers, Australia, p. 22.
Spall, James C and Daniel C Chin (1997). "Traffic-responsive signal timing for system-wide traffic control". In: *Transportation Research Part C: Emerging Technologies* 5.3-4, pp. 153–163.
Srinivasan, Dipti and Min Chee Choy (2007). "Distributed problem solving using evolutionary learning in multi-agent systems". In: *Advances in Evolutionary Computing for System Design*. Springer, pp. 211–227.
Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Sutton, Richard S, Andrew G Barto, et al. (1998). *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge.
Sutton, Richard S and Andrew G Barto (1998). *Learning: an introduction*.
Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press.
Tahilyani, S, M Darbari, and PK Shukla (2013). *Soft computing approaches in traffic control systems: a review. AASRI Procedia 4: 206–211*.
Tan, Ming (1993). "Multi-agent reinforcement learning: Independent vs. cooperative agents". In: *Proceedings of the tenth international conference on machine learning*, pp. 330–337.
Torabi, Behnam, Mohammad Al-Zinati, and Rym Z Wenkstern (2018b). "Matisse 3.0: A large-scale multi-agent simulation system for intelligent transportation systems". In: *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, pp. 357–360.
Transportation., US Department of (2008). *Ngsim - next generation simulation*. URL: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.
Tricco, Andrea C, Erin Lillie, Wasifa Zarin, Kelly K O'Brien, Heather Colquhoun, Danielle Levac, David Moher, Micah DJ Peters, Tanya Horsley, Laura Weeks, et al. (2018). "PRISMA extension for scoping reviews (PRISMA-ScR): checklist and explanation". In: *Annals of internal medicine* 169.7, pp. 467–473.
Urbanik, Thomas, Alison Tanaka, Bailey Lozner, Eric Lindstrom, Kevin Lee, Shaun Quayle, Scott Beaird, Shing Tsoi, Paul Ryus, Doug Gettman, et al. (2015). *Signal timing manual*. Vol. 1. Transportation Research Board Washington, DC.
Van der Pol, Elise and Frans A Oliehoek (2016). "Coordinated deep reinforcement learners for traffic light control". In: *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*.
Varaiya, Pravin (2013). "The max-pressure controller for arbitrary networks of signalized intersections". In: *Advances in Dynamic Network Modeling in Complex Transportation Systems*. Springer, pp. 27–66.
Vu, Huan, Samir Aknine, and Sarvapali D Ramchurn (2018). "A Decentralised Approach to Intersection Traffic Management." In: *IJCAI*, pp. 527–533.
Wang, Yizhe, Xiaoguang Yang, Hailun Liang, and Yangdong Liu (2018a). "A review of the self-adaptive traffic signal control system based on future traffic environment". In: *Journal of Advanced Transportation* 2018.
Wang, Yuan, Dongxiang Zhang, Ying Liu, Bo Dai, and Loo Hay Lee (2019). "Enhancing transportation systems via deep learning: A survey". In: *Transportation research part C: emerging technologies* 99, pp. 144–163.
Wang, Zhigang, Lichuan Liu, MengChu Zhou, and Nirwan Ansari (2008). "A position-based clustering technique for ad hoc intervehicle communication". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.2, pp. 201–208.
Wang, Ziyu, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas (2016). "Dueling network architectures for deep reinforcement learning". In: *International conference on machine learning*. PMLR, pp. 1995–2003.
Watkins, Christopher J. C. H. and Peter Dayan (1992). "Q-learning". In: *Mach. Learn* 8, pp. 279–292.
Webster, Fo Vo (1958). *Traffic signal settings*. Tech. rep.
Wei, Hua, Guanjie Zheng, Vikash Gayah, and Zhenhui Li (2019c). "A survey on traffic signal control methods". In: *arXiv preprint arXiv:1904.08117*.
Wei, Hua, Guanjie Zheng, Vikash Gayah, and Zhenhui Li (2021). "Recent Advances in Reinforcement Learning for Traffic Signal Control: A Survey of Models and Evaluation". In: *ACM SIGKDD Explorations Newsletter* 22.2, pp. 12–18.
Wu, Cathy, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen (2017). "Flow: Architecture and benchmarking for reinforcement learning in traffic control". In: *arXiv preprint arXiv:1710.05465*, p. 10.
Wunderlich, R, I Elhanany, and T Urbanik (2007). "A stable longest queue first signal scheduling algorithm for an isolated intersection". In: *2007 IEEE International Conference on Vehicular Electronics and Safety*. IEEE, pp. 1–6.
Yau, Kok-Lim Alvin, Junaid Qadir, Hooi Ling Khoo, Mee Hong Ling, and Peter Komisarczuk (2017). "A survey on reinforcement learning models and algorithms for traffic signal control". In: *ACM Computing Surveys (CSUR)* 50.3, pp. 1–38.
Yen, Gary, Fengming Yang, and Travis Hickey (2002). "Coordination of exploration and exploitation in a dynamic environment". In: *International Journal of Smart Engineering System Design* 4.3, pp. 177–182.
Zhao, Dongbin, Yujie Dai, and Zhen Zhang (2011). "Computational intelligence in urban traffic signal control: A survey". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.4, pp. 485–494.
Zheng, Guanjie, Xinshi Zang, Nan Xu, Hua Wei, Zhengyao Yu, Vikash Gayah, Kai Xu, and Zhenhui Li (2019b). "Diagnosing reinforcement learning for traffic signal control". In: *arXiv preprint arXiv:1905.04716*.