**SCMA: A Sparse Cooperative Multi-Agent Framework for Adaptive Traffic Signal Control**

1st Mahmoud Abo Ayaad

*Electrical Engineering Department Shoubra Faculty of Engineering* Benha University, Cairo, Egypt mahmoud.taha17@feng.bu.edu.eg

2nd Ahmed Zakey

*Electrical Engineering Department Shoubra Faculty of Engineering* Benha University, Cairo, Egypt ahmed.zaky@feng.bu.edu.eg

3rd Abdulwahab Al-Sammak *Electrical Engineering Department Shoubra Faculty of Engineering* Benha University, Cairo, Egypt [asammak@feng.bu.edu.eg](mailto:asammak@feng.bu.edu.eg)

***Abstract*—Building scalable, adaptive, and collaborative traffic signal control system still remains to be further explored across relevant research communities, including computer science and transportation groups. In this study, a scalable multi-agent frame- work is proposed based on the coordination graphs framework where the global objective is decomposed into a linear sum of local edge-based functions. The proposed edge-based decomposition scales linearly with edges in dense networks. A novel combination of max-plus joint action selection algorithm with two collaborative model-free methods, including sparse cooperative Q-learning (SparseQ) and relative sparse cooperative Q-learning (RSparseQ), is utilized to control multi-intersection networks. Extensive ex- periments are carried out, and their results demonstrate the effectiveness of our proposed framework. In comparison with independent Q-learning agents, our proposed framework achieves superior performance in terms of vehicle trip time, waiting time and jam length. In addition, the reported results show that the proposed RSparseQ outperforms SparseQ in avoiding vehicles teleports, which leads to better driver satisfaction.**

***Keywords*—*Adaptive traffic signals; sparse cooperative Q- learning; cooperative multi-agent***

1. INTRODUCTION

In 2050, the world population would reach 9.3 billion people, and the majority of this population (70%) would live in urban areas [1]. This increase in population would lead to a higher mobility demand, and the traffic congestion cost would gradually increase. In our daily life, traffic jam wastes our times, ties up people and their economic activities and even harms the environment. Traditionally, the congestion was eliminated by infrastructure enhancements; however, the constraints on the available resources raised the need for alternative solutions that utilize the available infrastructure without building new roads or expanding the current ones.

Intelligent Transportation Systems (ITS) would be a poten- tial solution; ITS utilize the existing infrastructure by applying modern smart techniques in traffic control. For example, in urban areas, ITS could be used in traffic signal control to achieve different objectives such as eliminating intersection queues or minimizing vehicles delay.

At present, pre-timed and actuated signals are the most commonly used traffic signal control systems. While pre- timed signals are simple and easy to operate, they usually

*§*c 2019 IEEE

require extended periods to build an appropriate timing plan. Moreover, pre-timed signals do not respond to flow fluctuations and usually become outdated for months or years. Besides, although the actuated signals handle different demands by applying minimum and maximum green time constraints, they may cause long queues in the grid-like networks [2].

The adaptive traffic control system (ATCS) handles traffic fluctuations by adjusting the signal timing parameters. For example, systems such as Sydney Coordinated Adaptive Traffic System (SCATS) [3] and Split Cycle and Offset Optimization Technique (SCOOT) [4] improved traffic flow in many coun- tries and outperformed pre-timed and actuated signals. Despite their improvements, they have many limitations, e.g., high implementation, maintenance, and operation costs. Moreover, these systems failed to gain acceptance in some countries as they depend on accurate traffic models that require highly- skilled engineers to build and maintain [5] [6].

Reinforcement learning (RL) has shown recently good potential in controlling traffic signals, specifically isolated intersections. RL agents have the advantage of being able to learn and improve over time by iteratively receiving feedbacks from the environment. In real life, multiple intersections often coexist in the same urban area; i.e., controllers influence each other; therefore, the coordination between controllers is substantial to achieve the global network-wide objective. Multi-agent RL (MARL), which is the extension of RL to multiple agent systems, could be used in such circumstances. The recent research efforts in applying MARL for collaborative traffic signals control could be shortened by three steps [7]: (i) each agent observes a local representation of the environment;

(ii) agents update their local Q-values based on information received from neighbors; and (iii) a joint action is selected in a way that maximizes the global reward. There are different challenges for such agent-based systems. For example, the agent Q-values are updated based on information received from neighbors only; the agent may use neighbors rewards to calculate a weighted reward or use neighbors Q-values to update its local values. Consequently, local values are not updated based on the global objective. Another related weakness of agent-based structure is the exponential growth with the number of neighbors, in case that the agent considered neighbors information in its Q-value representation.

We propose a scalable multi-agent framework for adaptive traffic signal control based on sparse cooperative Q-learning (SCMA). SCMA employs a unique combination of the max-

Accepted in the 14*th* IEEE International Conference on Computer Engineering and Systems (ICCES 2019)

plus algorithm with two collaborative model-free Q-learning methods: SparseQ and RSparseQ. In comparison with the existing research efforts, the novelty of our contributions can be highlighted as: firstly, the global objective is decomposed into a linear sum of edge-based Q-value functions. Secondly, the edge-based Q-values are updated by sparse cooperative methods based on contributions to the global reward; conse- quently, the individual agents do not store any information. Lastly, the max-plus messages are estimated explicitly from the edge-based values.

The remainder of this paper is organized as follows. The proposed framework is described in section II. In section III, the experimental results are reported. Finally, the work is summarized, and the future directions are outlined in section IV.

1. PROPOSED FRAMEWORK

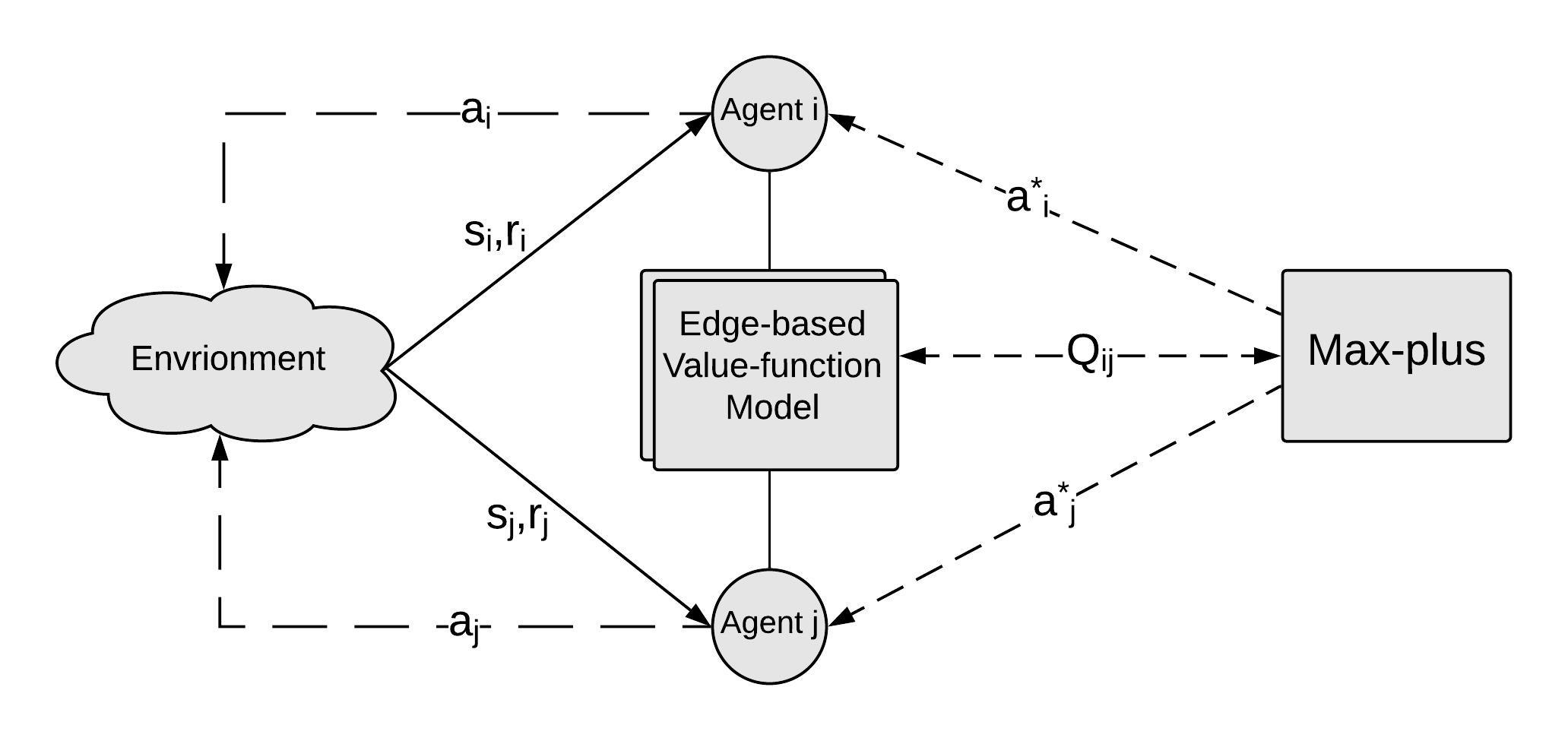


Fig. 1: Proposed framework architecture

The traffic light control problem could be formulated as RL optimization problem where each intersection is controlled by a RL agent, and at every time step *t*, the agent observes the environment state *st*, then takes an action *at*, and finally receives a delayed reward *rt*+1. Meanwhile, the agent aims at learning an optimal policy *πv* that maximizes the expected discounted reward. A popular approach to learn *πv* is finding optimal state-action values *Qv*(*s, a*) using the Q-learning al- gorithm [8]. *Q*(*s, a*) estimates the expected discounted reward by following the policy *π* after taking action *a* in state *s*. In Q- learning, we start with random *Q*(*s, a*) that approximates the optimal *Qv*(*s, a*) and update it using the bellman equation:

*Q*(*st, at*) ← *Q*(*st, at*) + *α*[*rt*+1 + *γ* max *Q*(*st*+1*, a*) − *Q*(*st, at*)] *γ* discount factor; 0 ≤ *γ* ≤ 1

*a*

*α* learning rate; 0 *< α <* 1

(1)

Consider a group of signal-controlled intersections where each intersection is controlled by an RL controller, and the controllers aim to optimize global performance measures, e.g., reducing the vehicles waiting time, increasing the total throughput, etc. This situation could be described by a collab- orative multi-agent system (MAS) which is formulated in [9] as follows:

* A set of *n* agents {*i, j, . . . , n*}*.*

A set of actions *Ai* for every agent *i*. The action selected by agent *i* is *ai Ai*; hence, the joint action **a** *A* = *A*1 *A*2 *An*.

A set of state variables *Si* for every agent *i*; hence, the global state **s** *S* = *S*1*S*2 *Sn.*

∈ × · · · ×

•

× · · · ×

∈ ∈

•

Reward function *Ri* : *S A* R defines the agent *i* reward *Ri*(**s***,* **a**) based on the global state **s** and the joint action **a**; hence, the global reward *R*(**s***,* **a**) = *Ri*(**s***,* **a**). State transition function *p*(**s**r **s***,* **a**) defines the probability to reach any global state **s**r, given current global state **s** and the joint action **a**.

Σ

• |

• × →

In principle, MAS could be considered as a large RL agent with states **s** and actions **a**; hence, the agent learns the global Q-function *Q*(**s***,* **a**) like any typical RL agent; however, this is impractical as the state-action space would be exponential in the number of agents *n*. Moreover, this vast space would result in a low convergence speed.

An alternative approach is using the coordination graphs CGs framework, CG is an undirected graph with a set of vertices *V* and a set of edges *E* where each edge (*i, j*) *E* indicates a dependency between the two corresponding agents *i* and *j*; i.e., *i* Γ(*j*) and *j* Γ(*i*). In such formulation, the global *Q*(**s***,* **a**) is decomposed into a sum of local Q- functions. While this solution could scale efficiently, this depends on three decisions: (i) how the global Q-function is decomposed: agent-based or edge-based; (ii) how the joint action is negotiated; and (iii) how the local Q-functions are learned from the global values.

∈

∈ ∈

In this paper, the problem of coordinated traffic signal control is defined as a collaborative multi-agent RL joint op- timization problem where we propose a solution based on the coordination graphs framework such that the global objective is decomposed into a linear sum of edge-based functions. In addition, the max-plus joint action selection algorithm is in- tegrated with two collaborative model-free methods, including sparse cooperative Q-learning (SparseQ) and relative sparse cooperative Q-learning (RSparseQ), into SCMA framework to control multi-intersection networks. A major contribution of SCMA is that agents do not store any local values; i.e., the Q-values are defined over edges; therefore, the max-plus messages are calculated from the edge-based values, and the system is scalable. Besides, the agents utilize a multi-objective reward function that has been evaluated on isolated intersection scenarios. The complete framework is shown in Fig. 1.

1. *The proposed multi-agent RL algorithm*

There are many ways to adopt RL for the multi-agent problem; in this work, two different modes are used: coor- dinated and independent modes. In the coordinated mode, the coordination graphs framework is utilized such that the graph shows the dependencies between connected intersections only, and the agents coordinate their actions to maximize the total reward. The core of coordinated mode is SCMA framework. In the independent mode, the learned model of an isolated agent is transferred to multi-agent traffic networks where the agents act in the network without any coordination. This mode is used as a reference for the coordinated mode. Fig. 1 shows SCMA, it is a model-free collaborative RL framework. SCMA is based on coordination graphs and could be outlined by three main

building blocks: the coordination graph decomposition method, the Q-values update technique, and the joint action selection algorithm.

* 1. *Coordination graph decomposition technique:* The edge-based approach is adopted; i.e., the global Q-function is decomposed into a linear combination of local *Qij* functions; each local *Qij* function is defined over an edge (*i, j*) of CG (*V, E*) :

RL proposed by Wiering [12] where the transition model is estimated and used in the update of the agents vehicle-based value functions. Moreover, Kuyer calculated *µij* (*aj* ) from these agents vehicle-based values. On the contrary, the pro- posed agent state is a junction-based representation and is used only to construct the joint state **s***ij* . A unique characteristic in the proposed work is that agents neither store the Q-values nor learn their model; we only learn the model the edge-based *Qij*

*Q*(**s***,* **a**) = Σ *Q* (**s** *, a , a* ) (2)

functions. Therefore, max-plus messages *µij* (*aj* ) are explicitly

*ij ij i j*

(*i,j*)∈*E*

This decomposition has two advantages over the agent- based approach. First, the local functions depend only on the state variables and the actions of two agents *i* and *j*; therefore, the decomposition scales linearly with the number of edges. Second, it will promote using max-plus algorithm [10] for determining optimal joint action **a**∗.

* 1. *Joint action selection algorithm:* Given the current global ,

calculated from *Qij* functions. Moreover, SCMA depends on model-free collaborative Q-learning methods where the *Qij* model is updated based on samples from the environment.

* 1. *Q-values learning technique:* A rising question is how the local edge-based *Qij* function could be updated based on the agent-based reward; i.e. how the agent-based reward could be propagated to the edges since an agent may have many dependencies, but it receives only one reward. One simple solution is to divide the reward of an agent equally between its connected edges; thus, each edge (*i, j*) ∈ *E* gains

state **s** ∈ *S* and all action combinations *A* the agents

two proportional parts of the rewards received by the two

should determine an action **a**∗ *A* that maximizes the global return from the environment. Since the global value function is decomposed into a linear combination of local

∈

corresponding agents *i* and *j* :

*Ri*(**s***,* **a**) *Rj* (**s***,* **a**)

functions, variable elimination (VE) algorithm could find the optimal joint action by eliminating the agents one by one after performing a local maximization step [9]. However, the main

*Rij* (**s***,* **a**) =

+

Γ(*i*)

(5)

Γ(*j*)

drawback of VE is that it is exponential in the induced tree width.

Max-plus is a message passing algorithm developed for inference in probabilistic graphical models and could be used as an approximate alternative to VE since it scales to graphs with a large number of dependencies [10]. In max-plus, the agent *i* iteratively sends to its neighbor *j* a message *µij* (*aj* ) which represents the maximum payoff agent *i* could gain if the agent *j* takes action *aj* . We define the message *µij* (*aj* ) as

:

With this reward definition, Q-learning could be extended

for multi-agent scenarios through two model-free methods: sparse cooperative Q-learning and relative sparse cooperative Q-learning.

* 1. *Sparse cooperative Q-learning:* Sparse Q-learning (SparseQ) updates the local value functions based on the local contributions to the global values [13]. Hence, the agent that contributed more to the global return would not be updated like the others. In other words, this update method helps the coordinated agents to identify the agent that is responsible

this method, the edge-based *Qij* is updated based on the edge reward described by (5) and based on its contribution to next

*µ* (*a* ) = max{*Q* (**s** *, a , a* ) + Σ *µ* (*a* )} + *c* (3)

*ij*

*j*

*ai*

*ij*

*ij*

*i*

*j*

*k*∈Γ(*i*)\*j*

*ki*

*i*

*ij*

for the global values and to reward that promising agent. In

where the joint state **s***ij* is constructed by direct concatenation of the individual corresponding states.

This definition is the main reason for scalability of max- plus, as the agent sums over the incoming messages from neighbors only, and these messages are defined over individual actions. The term *cij* which is a negative value is used to normalize any sent message to prevent messages from getting very large. Edge-based decomposition facilitates the usage of max-plus, as an alternative to VE, for the negotiation of the joint action. We limit the max-plus run time to ten iterations; after each iteration, the individual optimal action is updated according to:

{ Σ

optimal joint Q-value.

*Qij* (**s***ij, ai, aj* ) ← *Qij* (**s***ij, ai, aj* )

+ *α*[*Rij* (**s***,* **a**) + *γQij* (**s**r*ij , a*∗*i , a*∗*j* ) *Qij* (**s***ij, ai, aj* )]

−

(6)

The edge contribution to the next optimal global Q-value is defined by *Qij* (**s**r*ij ,* **a**∗*i ,* **a**∗*j* ) and can be calculated by first determining the next optimal joint action **a**∗ using max-plus,

then the local contribution is computed, which may be lower than the maximum of the local Q-values since this local action **a**∗*i* is part of the optimal joint action **a**∗

* 1. *Relative sparse cooperative Q-learning:* We further

*a*∗*i* = arg max

*ai*

*µki*(*ai*)} (4)

propose a modified version of SparseQ, relative sparse Q-

*k*∈Γ(*i*)

Although the approach of applying max-plus with edge- based decomposition has been previously used by Kuyer et al. [11], the proposed architecture is different from Kuyer’s

learning (RSparseQ). The main feature of RSparseQ is that the update of the local Q-function *Qij* is based on the relative contribution to the global values. For a set of local functions *fi*, the relative contribution function *fri* is defined as:

work in many points. First, Kuyer et al. used the vehicle-based

Σ

*fi*

(7)

state representation. Second, they extended the model-based

*fri* =

*k*

|*fk*|

Hence, the proposed modification of (6) is :

*Qij* (**s***ij, ai, aj* ) ← *Qij* (**s***ij, ai, aj* )

*Rij* (**s***,* **a**)

must insert the suitable yellow phase for three seconds before applying the new phase. This results in safe transitions in the intersection. Moreover, there’s no upper boundary for action selection times; while the action definition results in a

+ *α*[ Σ

(*k,l*)∈*E*

|*Rkl*

(**s***,* **a**)|

minimum green time of ten seconds followed by three seconds for the yellow phase, the maximum green time depends on the

*Qij* (**s**r*ij , a*∗*i , a*∗*j* )

(8)

traffic demand. Finally, during training, the action is selected

+ *γ* Σ

by the *s* greedy strategy with constant *s* = 0*.*1.

|*Q* (**s**r *, a*∗ *, a*∗)|

*Qij* (**s***ij, ai, aj* ) ]

(*k,l*)∈*E*

*kl*

*kl*

*k*

*l*

(*k,l*)∈*E* |*Qkl*(**s***kl, ak, al*)|

− Σ

* + 1. *Reward function:* There is no ultimate goal in traffic control problem; an agent could learn to optimize different

The two techniques SparseQ and RSparseQ have been imple- mented and tested on different traffic scenarios. Algorithm 1 illustrates the proposed training algorithm of the multi-agent framework.

**Algorithm 1** The proposed multi-agent algorithm

1: Initialize *Qij* (**s***ij, ai, aj* ) model to random values

2: **for** every *N* seconds **do**

3: **for each** agent *i* **do**

4: Observe state *Si* and reward *Ri*(**s***,* **a**)

5: **end for**

6: Run max-plus to find next optimal joint action **a**∗

7: **for each** edge (*i, j*) *E* **do**

∈

8: Update *Qij* (**s***ij, ai, aj* ) model according to (6) or (8)

9: **end for**

10: **for each** agent *i* **do**

11: Select between *a*∗*i* or random *ai* according to *s* greedy strategy

measures, e.g., queue length, travel time, delay, etc [7]. We adopt a multi-objective function, including the waiting vehicles count and the sum of waiting times where the reward is designed as the difference between the current and previous values. For an intersection with the number of waiting vehicles *N* and the sum of waiting times *W* , the reward function is:

*rt* = *θ*0(*Nt* − *Nt*−1) + *θ*1(*Wt* − *Wt*−1) (9)

The weighting parameters *θ*0 and *θ*1 were found, by a rough search, to be 0.9 and 0.1, respectively.

* + 1. *Value function approximation:* For the purpose of demonstration of the coordination framework, the linear ap- proximation is used, and our future work would consider using deep neural network to model the value function. The learning rate *α* is designed to decay with time as follows:

log(*t*)

≥

12: **end for**

13: **end for**

*α* = *t* 1 (10)

*t*

1. *The proposed RL agent parameters*

As aforementioned, the edge-based state and reward are functions of the agent state and reward; therefore, the isolated RL agent is designed and tested on a single intersection as follows:

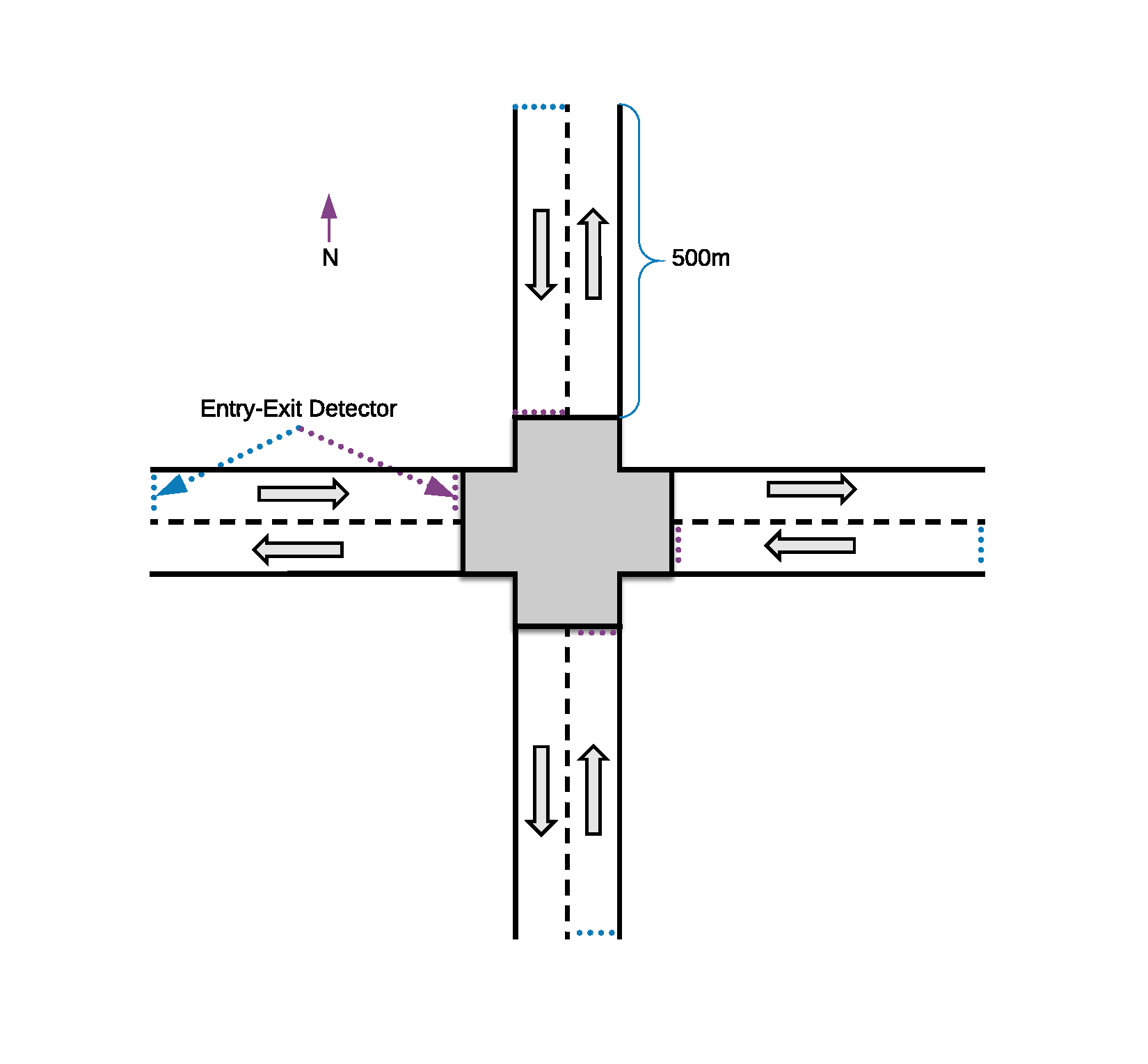
* 1. *State definition:* To demonstrate the proposed co- ordination framework, we adopt a junction-based state repre- sentation where the agent state consists of two main features: the number of waiting vehicles and the sum of waiting times. These two features are collected for every incoming approach to the intersection. This results in eight features vector for a typical four approaches intersection. After that, every feature is normalized to the total of the corresponding ones. As a final step, the current running action is encoded using one-hot encoder, then it is appended to the feature vector.
  2. *Actions definition:* The agent commits an action every ten seconds, and the action is defined as the signal phase index which will be applied in the intersection. Pol and Oliehoek proposed that the RL agent could take action every time step (one second) [14]; however, this is impractical and requires more information to be added in the state definition to prevent flickering situations. Table I shows the signal phases and the corresponding actions. It shows that not all phases are mapped to possible actions. The agent chooses which approaches would get the right-of-way or the green light; i.e., the yellow phases are not mapped to actions. As a result, if the new action is different from the running one, the agent

1. EXPERIMENTS AND RESULTS ANALYSIS

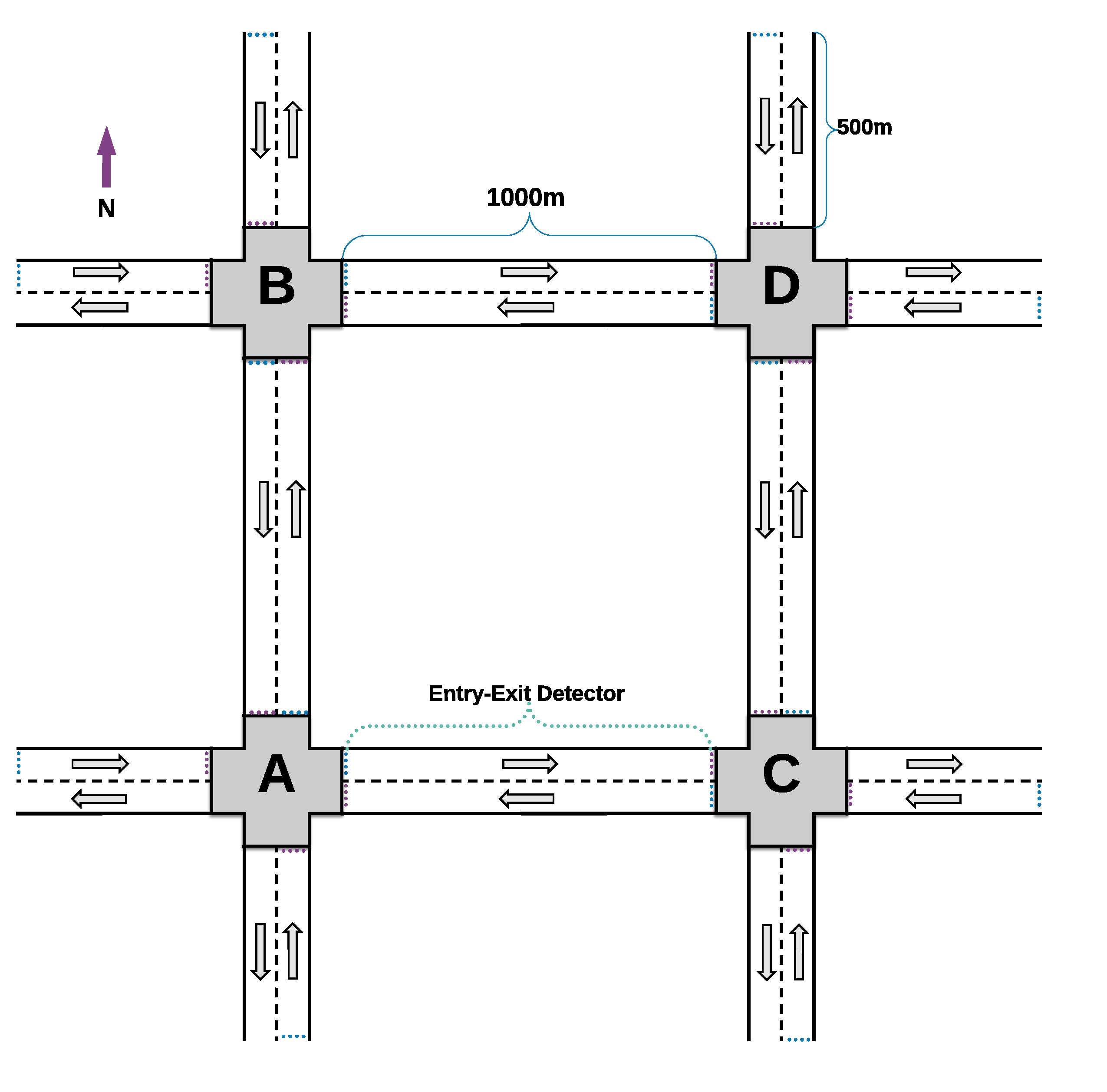
In general, all the experiments are conducted on a stan- dard traffic control simulator, Simulation of Urban Mobility (SUMO) [15]. SUMO is a widely used open source simulator that supports the microscopic model, and it runs in either con- sole or graphical modes. SUMO provides a powerful interface (TRACI) to monitor and control the simulation from external programs. To evaluate the proposed SCMA framework, we carry out two phases of extensive experiments: First, the single RL agent experiment is developed where RL agent is used to control an isolated intersection; the RL agent is tested based on different measures: travel time, link jam length and vehicle waiting time. Furthermore, particular experiments are carried out to evaluate the agent ability to handle traffic fluctuations, and finally, the proposed agent is evaluated using a special scenario where the vehicles routes are determined by dynamic routing at the simulation runtime. Second, the two multi-agent modes experiments, independent and coordinated, are carried out in two different networks. To reveal the contribution of each method designed in the coordinated mode, SparseQ and RSparseQ methods are compared with the independent mode in terms of travel time, jam length and vehicle waiting time. Finally, the impact of traffic fluctuation on the agents behavior is demonstrated using a special scenario.

1. *Assessment of Single Agent Parameters*

*Traffic scenario and implementation details:* The traffic scenario reported by [14] and [16] is used and could be described as follows.



* 1. Single agent network (turns are prohibited)



* 1. Four intersections network (turns are prohibited)

Fig. 2: Experiments networks

* + 1. *Network:* The traffic network is shown in Fig. 2a, it consists of a single intersection with four edges where each edge has one incoming and one outgoing lane. All lanes are 500 meters long, and all turns at the intersection are prohibited. SUMO multi-entry multi-exit detectors are used to observe the intersection state; they are inserted a ten meters far from the edges beginning.
    2. *Traffic configuration:* The intersection is controlled by a four-phase traffic signal where only straight-going phases are allowed, and each phase indicates the states red (r), green

{

(G) or yellow (y) of the oncoming lanes. Table I shows the four phases; two phases are mapped to the agent actions, and the others are considered safety transition phases.

}

* + 1. *Traffic demand:* Since turns at the intersection are prohibited, there are four possible routes for any vehicle trip (North-South, East-West and vice versa). SUMO accepts the traffic demand via routes file that defines the departure time and the complete path for every vehicle in the simulation. Algorithm 2 describes the demand generation of the basic scenario; it simulates one hour of traffic. The vehicles are inserted according to uniform distribution with probability 0.1 to insert vehicles every second at any possible routes; i.e., the expected number of inserted vehicles at the network is

0*.*1 × 3600 × 4 = 1440 v/h.

TABLE I: Traffic configuration and corresponding actions of single agent

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Right-of-way | Action | Safety transition phase |
| GrGr  rGrG | N-S, S-N  E-W, W-E | 1  2 | yryr  ryry |

**Algorithm 2** Traffic demand generation algorithm

1: prepare the legal routes list

2: initialize *t* = 0

3: **while** *t < N* **do**

4: **for each** possible route *x* **do** 5: generate random value *r* 6: **if** *r < p* **then**

7: insert vehicle at time *t* with route *x*

8: **end if**

9: **end for**

10: **end while**

*Performance evaluation:* The single agent is trained using a set of one-hour scenarios generated once according to a predefined seed, and after every 10,000 training steps, the agent greedy policy is evaluated on a set of 16 scenarios generated from the same training distribution, but with a different seed. The average vehicle travel time (trip duration) is used to evaluate the agent performance; it is considered because travel time is not directly related to the reward function. Fig. 3 shows the evaluation of the proposed agent by different measures. Fig. 3a shows that reported average travel time after 10,000 training steps is 158.46 seconds, and it converges with training; this indicates that agent learns well. The average waiting time per vehicle is additionally used to evaluate the proposed agent. Fig. 3b shows the reported average waiting time with training time; it demonstrates that the waiting time is reduced by 35% after 106 training steps. The widely used jam length measure is additionally used to evaluate the agent policy. Fig. 3c illustrate the effect of learning on the jam length; it shows that after 106 training steps, the average link jam length is reduced by 35%.

*Testing the proposed agent in a congested scenario with dynamic routing:* We modified the main scenario as follows. First, right turns at the intersection are allowed; this results in a variety of routes for each trip. Since there are four source edges and four destination edges, there exist 16 source-destination pairs. Second, the demand is generated in a way that 360 v/h is inserted per source-destination pair; i.e., the total vehicles inserted at the network is 360 16 = 5760 v/h. Finally, the demand is described by source and destination only; hence, the dynamic routing capability of SUMO is used; i.e., the route is computed by SUMO at simulation runtime based on the recent state of traffic in the network. The results of the experiment are shown in Fig. 4; it demonstrates that the agent greedy policy is improved over training time regardless of the congested demand and the dynamic routing assignment. After 106 training steps, the reduction in travel time, waiting time and link jam length is 61%, 63%, and 25% respectively.

×

1. *Multi-agent experiments*

Two grid networks with four and nine intersections are used; they are identical, except in the intersections number;

145

Average Travel Time (Seconds)

Average Waiting Time (Seconds)

Average Jam Length per Link (Vehicles)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 155 |  | Prop | osed Single | Agent | 34  32 |  | Prop | osed Single | Agent | 4.0  3.8 |  | Pro | posed Single | Agent |
| 150 |  |  |  |  | 30 |  |  |  |  | 3.6 |  |  |  |  |
|  |  |  |  |  | 28 |  |  |  |  | 3.4 |  |  |  |  |
| 140 |  |  |  |  | 26 |  |  |  |  | 3.2  3.0 |  |  |  |  |
|  |  |  |  |  | 24 |  |  |  |  | 2.8 |  |  |  |  |
| 135 |  |  |  |  | 22 |  |  |  |  | 2.6 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* 1. Average travel time

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* 1. Average waiting time

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* 1. Average jam length

Fig. 3: Results of single agent during testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 600 | Proposed Agent | 180 | Proposed Agent | 19 | Proposed Agent |
| 550 |  | 160 |  | 18 |  |
| 500 |  |  |  | 17 |  |
| 450 |  | 140 |  | 16 |  |
| 400 |  | 120 |  | 15 |  |
| 350 |  | 100 |  | 14 | |
| 300  250 |  | 80  60 |  | 13  12 | |

0.0 0.2 0.4 0.6 0.8 1.0

Average Travel Time (Seconds)

Average Waiting Time (Seconds)

Average Jam Length per Link (Vehicles)

Training Steps 1e6

1. Average travel time

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

1. Average waiting time

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

1. Average jam length

Fig. 4: Results of single agent during testing in congested scenario with dynamic routing feature

the four-intersection network is shown in Fig. 2b. Each net- work is constructed from identical intersections where every intersection is controlled by a four-phase signal as described in subsection III-A. Since turns at intersections are prohibited, there are eight and twelve possible routes for the four and nine networks respectively; the demand for the networks is generated from the uniform distribution as explained in Algorithm 2.

1. *Independent mode:* In this mode, the model of the Q-values is transferred from the isolated intersection agent, after 106 training steps, to every agent in the multi-intersection networks. There is no coordination between agents in this mode; thus, it is mainly used as a reference for the coordinated mode.
2. *Coordinated mode:* As illustrated in algorithm 1, the agents start training from a random phase and every ten seconds the max-plus algorithm is used to determine the optimal global action, and each agent could select either the best action or a random one. If the elected phase is different from the current one, a suitable transition phase is inserted for three seconds to ensure safety. Therefore, the minimum duration of any action is seven seconds, and the maximum is a multiple of ten in addition to the final seven seconds. In SCMA framework, the max-plus algorithm is limited to ten iterations. It is important to mention that when max-plus is not allowed to reach a fixed point of convergence, it is necessary

to update *a*∗*i* only if the global *Q*(**s***,* **a**) get improved from the last iteration.

*Performance evaluation of coordinated mode:* Like the single agent, two different sets of scenarios generated from the same distribution are used to train and test the proposed

framework every 10,000 training steps. Same demands are used to train and evaluate SparseQ and RSparseQ techniques where the coordination strategies are evaluated using different measures: the average trip time, average waiting time and the average jam length per intersection.

To evaluate the proposed SparseQ and RSparseQ methods, we consider the independent mode as a reference. Fig. 5 and Fig. 6 illustrate that both coordinated strategies outperform the independent agents in terms of trip time, waiting time, and jam length measures. Table II reports the summary of the conducted experiments; it demonstrate that in the four agents experiment SparseQ and RSparseQ reduce the trip time by 9% and 2% and the waiting timing by 37% and 23% respectively. Moreover, the coordinated methods reduce the jam length by 34% and 14%.

In the nine agents experiment, SparseQ and RSparseQ reduce the trip time by 8% and 4% and the waiting time by 37% and 28% respectively. Furthermore, the coordinated methods reduce the jam length by 43% and 19%. These results indicate that coordinated methods show significant enhancement over the independent agents in the average jam length, and although the reduction in trip time is less than 10%, the reduction in waiting time is about 37%. Furthermore, the coordinated methods outperform the independent method within less training time (at most 15%). Another significant simulation output from SUMO is the number of teleported vehicles. Vehicles teleport due to different reasons such as long waiting time or collision. Hence, a good signal timing plan should prevent teleports. Table III shows that RSparseQ prevents vehicles from teleporting.

255

Independent Agents

Sparse Cooperative RL

Relative Sparse Cooperative RL

250

Average Travel Time (Seconds)

245

240

235

230

225

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* 1. Average travel time

50

45

Independent Agents

Sparse Cooperative RL

Relative Sparse Cooperative RL

Average Waiting Time (Seconds)

40

35

30

25

0.0

0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* 1. Average waiting time

12

11

Independent Agents

Sparse Cooperative

Relative Sparse Cooperative

Average Jam Length per Junction (Vehicles)

10

9

8

7

6

5

0.0

0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* 1. Average jam length

Fig. 5: Results of four agents network during evaluation

370

Independent Agents

Sparse Cooperative RL

Relative Sparse Cooperative RL

360

Average Travel Time (Seconds)

350

340

330

320

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* + 1. Average travel time

70

65

Independent Agents

Sparse Cooperative RL Relative Sparse Cooperative RL

Average Waiting Time (Seconds)

60

55

50

45

40

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* + 1. Average waiting time

12

11

Independent Agents

Sparse cooperative

Relative sparse cooperative

Average Jam Length per Junction (Vehicles)

10

9

8

7

6

5

0.0 0.2 0.4 0.6 0.8 1.0

Training Steps 1e6

* + 1. Average jam length

Fig. 6: Results of nine agents network during evaluation

*Testing with non-uniform demand:* Adaptive agents should change the timing plan according to the traffic con- ditions. To evaluate this feature in the proposed framework, the greedy policy of the SparseQ agents has been tested in

120

100

Frequency

80

60

40

20

0

120

100

Junction B

First Phase Second Phase

80

Frequency

60

40

20

0

two different four intersections scenarios, the first scenario with uniform demand across all links; the other includes high demand across East-West and West-East directions between

140

120

100

Frequency

80

60

40

20

1 2 3 4 5 6

Consecutive length of phases

Junction A

First Phase Second Phase

Junction C

First Phase Second Phase

140

120

100

Frequency

80

60

40

20

1 2 3 4

Consecutive length of phases

1

2

3

4

5

6

Junction D

First Phase Second Phase

intersections B and D. Since the proposed approach depends

0 1 2 3 4 5 6 0

Consecutive length of phases

Consecutive length of phases

on fixed green time for any phase, Fig. 7 shows the comparison between the two scenarios in terms of the histogram of the consecutive length of phases. It shows that in case of the uniform demand, all agents typically alternate between the two phases. In the other scenario, intersections B and D are biased towards the second phase which gives the right-of-way for

100

80

Frequency

60

40

20

1. Uniform demand

50

Junction A

First Phase Second Phase

Junction B

First Phase Second Phase

1

2

3

4

5

40

Frequency

30

20

10

East-West and West-East directions. This results is confirmed

0 1 2 3 4 5 0

Consecutive length of phases

First Phase 60

Consecutive length of phases

First Phase

by the timing diagram of ten consecutive actions which is shown in Fig. 8

80 Junction C

60

Frequency

40

20

Second Phase 50

40

Frequency

30

20

10

Junction D

Second Phase

0 1 2 3 4 5

Consecutive length of phases

0 1 2 3 4 5

Consecutive length of phases

TABLE II: Comparison of Sparse methods against independent agents

1. Non-uniform demand

**Network Method Waiting Trip Jam length**

**time (s) time (s) (vehicles)**

SparseQ 27.7 225 6.85

Fig. 7: Histogram of length of actions

Four agents RSparseQ 34.05 241.61 8.9

Independent 44.49 246 10.36

SparseQ 40.54 329.63 5.50

1. CONCLUSIONS AND FUTURE DIRECTIONS

Nine agents RSparseQ 46.22 346.87 7.86

In this paper, the model-free RL is utilized to control both

Independent 64.36 359.68 9.71

an isolated intersection and multi-intersection traffic networks.

Phase 1 Junction A Phase 2

Phase 1 Junction B Phase 2

Time (Seconds)

Phase 1 Junction C Phase 2

Time (Seconds)

Phase 1 Junction D Phase 2

Phase 1 Junction A Phase 2

Phase 1 Junction B Phase 2

Phase 1 Junction C Phase 2

Phase 1 Junction D Phase 2

0 10 13 20 23 30 33 40 43 50 53 60 63 70 73 80 83 90 93 100

Time (Seconds)

Time (Seconds)

10 1

3 2

0 2

3 30 3

3 4

0 4

3 50 5

3 6

0 6

3 70 7

3 8

0 8

3 90 9

3 10

1

0 1

3 20 2

3 3

0 3

3 40 4

3 5

0 5

3 60 6

3 7

0 7

3 80 8

3 9

0 9

3 10

1

0 1

3 20 2

3 3

0 3

3 40 4

3 5

0 5

3 60 6

3 7

0 7

3 80 8

3 9

0 9

3 10

0

0

0

0

0

0

* 1. Uniform demand

Time (Seconds)

10 1

3 20

3

0 3

3 40 43 50 5

3 6

0 6

3 70 73 80 83 90 9

3 10

1

0 1

3 20

3

0 3

3 40

5

0 5

3 60 6

3 7

0

80

9

0 9

3 10

1

0 1

3 20 23 30 3

3 40 43 5

0 5

3 60 63 70 73 80 8

3 9

0 9

3 10

0

0

0

0

0

0

0 10 13 20 30 33 40 50 53 60 70 80 83 90 93 100

Time (Seconds)

Time (Seconds)

Time (Seconds)

* 1. Non-uniform demand

Fig. 8: Timing diagram

environment and road users). Future investigations could be pursed to apply transfer learning to increase the learning speed of multi-agent scenarios. Enhancements to max-plus would be a future direction; the messages exchange overhead could be minimized by exchanging only ones that improve system performance.

REFERENCES

* 1. A. Pizam *et al.*, “Life and tourism in the year 2050,” *International journal of hospitality management*, vol. 18, no. 4, pp. 331–343, 1999.
  2. L. Zhang, H. Li, and P. D. Prevedouros, “Signal control for oversatu- rated intersections using fuzzy logic,” in *Transportation and Develop- ment Innovative Best Practices 2008*, 2008, pp. 179–184.
  3. P. Lowrie, “Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic,” Roads and Traffic Authority NSW, Tech. Rep., 1990.
  4. P. Hunt, D. Robertson, R. Bretherton, and R. Winton, “Scoot-a traffic responsive method of coordinating signals,” Tech. Rep., 1981.
  5. K. Fehon, “Adaptive traffic signals are we missing the boat,” *ITE District*, vol. 6, 2004.
  6. R. Cunningham, A. Garg, V. Cahill *et al.*, “A collaborative rein- forcement learning approach to urban traffic control optimization,” in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT’08.*, vol. 2. IEEE, 2008, pp. 560–566.
  7. K.-L. A. Yau, J. Qadir, H. L. Khoo, M. H. Ling, and P. Komisarczuk,

TABLE III: Comparison of Sparse methods in terms of tele- ports count

**Network Vehicles count Method Teleports count**

SparseQ 1827

Four agents ≈ 4*.*6*m* RSparseQ 0

“A survey on reinforcement learning models and algorithms for traffic signal control,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, p. 34, 2017.

* 1. C. J. C. H. Watkins, “Learning from delayed rewards,” Ph.D. disserta- tion, King’s College, Cambridge, 1989.
  2. C. Guestrin, M. Lagoudakis, and R. Parr, “Coordinated reinforcement

Nine agents

≈ 7*m*

SparseQ 3137

RSparseQ 1

learning,” in *ICML*, vol. 2, 2002, pp. 227–234.

* 1. J. R. Kok and N. Vlassis, “Using the max-plus algorithm for multiagent decision making in coordination graphs,” in *Robot Soccer World Cup*.

The single agent is characterized by a junction-based state representation and a multi-objective reward; it showed a solid performance in different scenarios. The problem of coordinated signals is described by the collaborative multi-agent RL frame- work where two multi-agent modes have been studied, inde- pendent and coordinated agents. In the coordinated mode, two different coordination strategies have been proposed, SparseQ and RSparseQ. Coordinated methods have led to better timing plans compared with the independent agents in terms of trip time, waiting time and jam length. The sparse methods are integrated with the max-plus algorithm into SCMA framework. SCMA employs the edge-based coordination graph decom- position; this enables the system to scale linearly in large networks. Our future work would consider using deep neural networks to model the edge-based value functions. It would additionally consider testing the SCMA framework in more complex networks that contain a large number of intersections with a high demand traffic and accustomed factors (e.g.,

Springer, 2005, pp. 1–12.

* 1. L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, “Multiagent rein- forcement learning for urban traffic control using coordination graphs,” *Machine learning and knowledge discovery in databases*, pp. 656–671, 2008.
  2. M. Wiering, “Multi-agent reinforcement learning for traffic light con- trol,” in *Machine Learning: Proceedings of the Seventeenth Interna- tional Conference (ICML’2000)*, 2000, pp. 1151–1158.
  3. J. R. Kok and N. Vlassis, “Collaborative multiagent reinforcement learn- ing by payoff propagation,” *Journal of Machine Learning Research*, vol. 7, no. Sep, pp. 1789–1828, 2006.
  4. E. Van der Pol and F. A. Oliehoek, “Coordinated deep reinforcement learners for traffic light control,” *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*, 2016.
  5. D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent devel- opment and applications of SUMO - Simulation of Urban MObility,” *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.
  6. S. S. Mousavi, M. Schukat, and E. Howley, “Traffic light control using deep policy-gradient and value-function-based reinforcement learning,” *IET Intelligent Transport Systems*, vol. 11, no. 7, pp. 417–423, 2017.