2023 IEEE Smart World Congress (SWC) | 979-8-3503-1980-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/SWC57546.2023.10448974

Traffic Signal Optimization by Integrating Reinforcement Learning and Digital Twins

Vijayalakshmi K Kumarasamy[a], Abhilasha Jairam Saroj[b], Yu Liang[a*], Dalei Wu[a*]
Michael P. Hunter[c], Angshuman Guin[c], Mina Sartipi[a]
[a] Department of Computer Science and Engineering, University of Tennessee at Chattanooga, USA
[b] Applied Research for Mobility Systems, Oak Ridge National Laboratory, USA
[c] School of Civil and Environmental Engineering, Georgia Institute of Technology, USA
Email: lry466@mocs.utc.edu; sarojaj@ornl.gov; {yu-liang, dalei-wu}@utc.edu;
{michael.hunter, angshuman.guin}@ce.gatech.edu; mina-sartipi@utc.edu
*Corresponding authors

***Abstract*—Machine learning (ML) methods, especially reinforcement learning (RL), have been widely considered for traffic signal optimization in intelligent transportation systems. Most of these ML methods are centralized, lacking in scalability and adaptability in large traffic networks. Further, it is challenging to train such ML models due to the lack of training platforms and/or the cost of deploying and training in a real traffic networks. This paper presents an approach for the integration of decentralized graph-based multi-agent reinforcement learning (DGMARL) with a Digital Twin (DT) to optimize traffic signals for the reduction of traffic congestion and network-wide fuel consumption related to stopping. Specifically, the DGMARL agents learn traffic state patterns and make decisions regarding traffic signal control with assistance from a Digital Twin module, which simulates and replicates the traffic behaviors of a real traffic network. The proposed approach was evaluated using PTV-Vissim [1], a microscopic traffic simulation platform. PTV-Vissim is also the simulation engine of the DT, enabling emulation and optimization of the traffic signals on the MLK Smart Corridor in Chattanooga, Tennessee. Compared to an actuated signal control baseline approach, experiment results show that Eco_PI, a developed performance measure capturing the impact of stops on fuel consumption, was reduced by 44.27% in a 24-hour and an average of 29.88% in a PM-peak-hour scenario.**

***Index Terms*—Multi-Agent Reinforcement Learning, Digital Twin, Graph Neural Network, Traffic Signal Optimization, Fuel Consumption**

I. Introduction

Cities across the world are deploying Intelligent Transportation System (ITS) technologies to smart corridors, creating smart and data-driven transportation systems [2]. The potential to leverage smart corridor high-resolution and high-frequency vehicle and infrastructure data, along with advancements in machine learning, artificial intelligence, and high-performance computing, is being explored to solve safety, mobility, and environmental transportation system challenges [3]–[5]. A potential solution for optimizing and improving transportation systems is the application of Digital Twin assisted decentralized multi-agent Reinforcement Learning (RL). The implementation of such an application requires a seamless handshake between the Digital Twin of the physical system and the decentralized multi-agent RL. The objective of this effort is to demonstrate this integration and utilize the resulting application to optimize traffic signal timing to reduce selected Eco_PI performance measure in a real-world cased study. Eco_PI performance measure captures the impact of stops on fuel consumption and delay. Additional details on Eco_PI may be found in [6]–[9].

II. Background

The concept of DT has gained significant attention in recent years, both in academia and industry, as a promising approach to improve the performance of physical systems through the use of virtual models [10], [11]. A DT is essentially a virtual representation of a physical entity, such as a machine, building, a smart corridor, or even an entire city. In a real-time application, the DT is continuously updated with data from sensors and other sources to reflect the current and historical state of the entire physical entity through different modeling approaches. This real-time and accurate representation of the physical entity, with different operational scenarios, allows for better prediction of future behavior, refinement of control, and optimization of operations. Diverse applications of DTa, including transportation, as well as modeling techniques and the benefits of integrating DT in system design are discussed in [12]. In contrast to traditional simulation models, which are often based on assumptions and simplified models, DTs is based on actual data and can provide a more accurate and realistic representation of the physical entity. This can be particularly useful in real world deployments and industries such as ITS technologies, where the performance and reliability of complex systems are critical.

This manuscript has been partly authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The publisher acknowledges the US government license to provide public access under the DOE Public Access Plan (https://energy.gov/downloads/doe-public-access-plan).

979-8-3503-1980-4/23/$31.00 ©2023 IEEE

Within transportation applications, DT simulations provide a realistic representation of real-world transportation systems. Thus, a DT may act as a crucial test bed to develop real-time machine learning based traffic operations applications, providing a safe, efficient, and economic environment to train and test AI/ML algorithms.

Previous studies such as [13] have developed DT for transportation systems that leverage real-time smart corridor data to model the current traffic state and provide dynamic traffic and performance measure updates. Data from various sources such as detectors and cameras, is continuously collected and fed to the DT. This enables the DT to accurately model the real-time traffic state and identify congested network sections. Such information can be used to make real-time or time-sensitive decisions. For example, in this paper, one such application is developed: intersection signal timing optimization to reduce the number of stops and stop delay.

DT of transportation systems can provide significant advantages in transportation management by enabling real-time monitoring and control, improving coordination across different parts of the transportation system, and increase traffic efficiency [14]. DT can enhance deep learning and reinforcement learning algorithms for real-time adaptive, precision-centric, and predictive traffic monitoring [15]. DT can assist reinforcement learning algorithms in learning the dynamic traffic state and making better real-time decisions through adaptive signal control [16] by utilizing data from various transportation components, such as detector occupancy and pedestrian recall time, and generating higher resolution representation on traffic states. It has been shown that integrating DT with RL agents could increase decision-making efficiency and enable agents to learn from past experience for future decisions [15].

This paper proposes a novel approach of DT assisted decentralized graph-based multi-agent RL (DGMARL) to learn the dynamic traffic state in terms of different network features such as detector occupancy, approach-level vehicle counts, pedestrian recall times, etc. These information are being received and shared with the neighboring agents, that is, the upstream and downstream intersections. This information allows the DGMARL to understand the upcoming traffic conditions and optimize the intersection signal timing. The proposed DGMARL signal timing solution is compared with a baseline solution. The baseline solution models a vehicle coordinated-actuated signal timing plan, received from the City. Both the coordinated-actuated and DGMARL traffic signal timing plan use sensors to detect the presence of vehicles and adjust the timing of the signal phases. Actuated control is based on a gap seeking logic, switching to conflicting movements as the length of gaps in the traffic stream increase, indicating reduced vehicle processing efficiency. The goal of DGMARL signal timing plan is to reduce the chosen performance metric (Eco_PI in this effort), providing superior performance to that of the actuated control. Evaluation of the proposed DGMARL approach shows better performance compared to the vehicle actuated signal timing plan, highlighting the potential of this technology in improving transportation operations and environmental impact.

The technical features of the proposed model include:

1) Integration of Digital Twin (DT) and Decentralized Graph-based Multi-Agent Reinforcement Learning (DGMARL) to optimize traffic signal timing;

2) Multi-agent reinforcement learning agents are distributed at individual intersections to observe traffic state features such as detector occupancy and exchange this information with neighboring agents. This information is used to find an optimal policy and subsequently choose the best action to control the traffic signals. The implementation of action is validated with rules and constraints such as minimum green time and pedestrian recall time, to enforce safe mobility for all users;

3) Proposed DGMARL model has the capability to handle heterogeneous data including detector occupancy, approach level vehicle count aggregates, pedestrian recall times, and current signal state, from current, upstream and downstream intersections;

4) Component Object Model (COM) interface of PTV-Vissim to take actions and control signal timing through DT;

III. Digital Twin System for Traffic Network

A. Physical Environment and Digital Twin

1) Digital Twin Architecture : Smart corridor Digital Twins are typically driven using real-time and historic vehicle and infrastructure data from the corridor [13], [16], [17]. In this study, the DT is developed using vehicle real-time and historic volume count, turn count, and Signal Phasing and Timing (SPaT) data available from approximately 2.1 miles of Martin Luther King Smart Corridor, Chattanooga, Tennessee, consisting of 11 signalized intersections. A smart corridor DT model architecture typically includes four key components as shown in Figure 1:

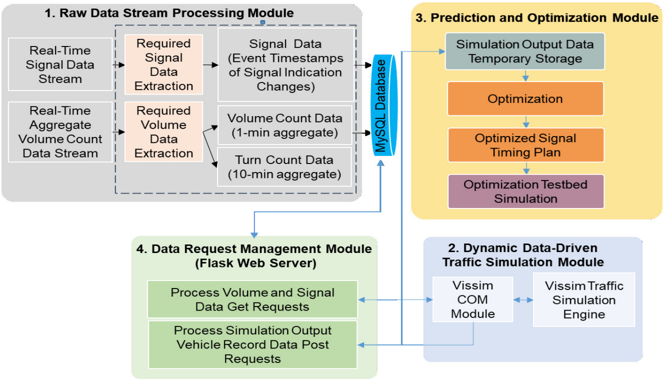

1. Raw Data Stream Processing Module
Real-Time Signal Data Stream
Required Signal Data Extraction
Signal Data (Event Timestamps of Signal Indication Changes)
Real-Time Aggregate Volume Count Data Stream
Required Volume Data Extraction
Volume Count Data (1-min aggregate)
Turn Count Data (10-min aggregate)
MySQL Database
3. Prediction and Optimization Module
Simulation Output Data Temporary Storage
Optimization
Optimized Signal Timing Plan
Optimization Testbed Simulation
4. Data Request Management Module (Flask Web Server)
Process Volume and Signal Data Get Requests
Process Simulation Output Vehicle Record Data Post Requests
2. Dynamic Data-Driven Traffic Simulation Module
Vissim COM Module
Vissim Traffic Simulation Engine

Fig. 1: Digital Twin Architecture

Module 1: Raw Data Stream Processing Module - includes processing of raw data to parse, format, and store the data in a database. From the physical MLK Smart Corridor, the left, through, and right turn vehicle counts per lane at the 11 intersections are obtained. This data is processed to obtain approach level (Eastbound, Westbound, Northbound, and Southbound) volume and turn counts. Further, 10 Hz Signal Phase and Timing (SPaT) data is obtained from the signal controllers in the corridor.

Module 2: Dynamic Data-Driven Traffic Simulation Module - includes PTV-Vissim microscopic traffic simulation model of the Smart Corridor, dynamically driven using volume, turn movement ratios, and signal indications data (from Module 1). In this implementation intersection approach level 1-minute aggregate volume counts, 10-minute aggregate turn counts data, and signal timing are dynamically driven using PTV-Vissim's COM module. Using COM the signal indications can be driven using external SPaT (Signal Phasing and Timing) data or PTV-Vissim's internal Ring Barrier Controller (RBC) module.

Module 3: Simulation Testbed Application Module - consists of tools and algorithms to process simulation outputs based on the requirements of the application. This module contains processes or algorithms that are driven using outputs from the DT simulation. In this study, the outputs generated from the PTV-Vissim simulation model are used as inputs for prediction and optimization for the signal timing plan.

Module 4: Real Time Data Broker Module - handles real time dynamic data transactions between modules. This module consists of a Flask based web service to handle data transactions/communication between other three modules.

2) Muti-tier Incremental Approach for Digital Twin Development: Smart corridor DT development requires integration and synchronization of multiple components within the DT architecture described in previous section. This makes DT development a time consuming process susceptible to coding, integration, implementation, data processing, and other errors. To tackle this, a three-tier incremental approach is used in this study that allows for a parallel workflow. The DT development process is broken into three tiers with increasing communication and infrastructure integration complexity. Such an approach enables training and testing of ML/RL based applications early on, on the initial tiers, thus, reducing the wait time required for development of a fully operational real-time DT. The three-tier incremental approach includes the following simulation model versions:

Tier 1 - Prepopulated model: traditional simulation model prepopulated with archived data. This version includes automation of raw data extraction and ingestion of extracted data by the PTV-Vissim model. Automation of the data handling in this tier is critical to the overall usability and effectiveness of this model version in training and testing the DGMARL model.

Tier 2 - Pseudo Digital Twin: simulation model driven dynamically using archived data. In this tier, the data is dynamically fed into the simulation. A significant advance in Tier-2 is the development of the dynamic links between the modules shown in Figure 1. Further, in this tier the signal indications are controlled using field received SPaT messages. This platform thus provides a test bed to develop the interface that integrates the DGMARL optimization algorithm with data driven DT simulation.

Tier 3 - Real time Digital Twin: online simulation model driven dynamically using real time field data. In this tier the simulation is driven using real-time data. The Tier-2 platform is modified and updated to stream real-time data. This platform will be used to develop the interface between the physical system represented by the DT and the optimization development algorithm.

In this study the interface between the RL optimization algorithm and physical system is initially developed using the Tier-1 platform. The developed RL algorithm in future will also be integrated with the Tier-2 and Tier-3 platforms to test and further improve the algorithm.

B. Digital Twin and RL

The integration of DGMARL model and its learning as shown in Figure 2 focuses on training agents associated with each intersection by learning from DT to make decisions to optimize the signal timing based on the observed traffic state.

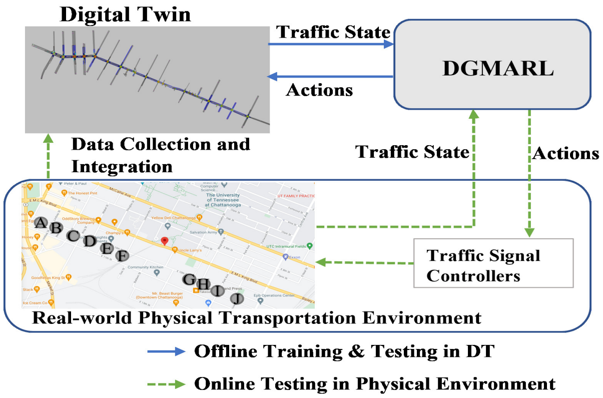

Digital Twin
Traffic State
Actions
DGMARL
Data Collection and Integration
Traffic State
Actions
A
B
C
D
E
F
G
H
I
J
Traffic Signal Controllers
Real-world Physical Transportation Environment
Offline Training & Testing in DT
Online Testing in Physical Environment

Fig. 2: Digital Twin assisted DGMARL Learning

The integration between the DT and the distributed multi-agent reinforcement learning algorithm is described as following:

1) The distributed multi-agent reinforcement learning algorithm takes the input, such as traffic occupancy, current phase state, pedestrian recall time, etc., from the DT and makes the decision of staying in current phase or switching to the phase with the upcoming high traffic occupancy based on its current state and the desired objective, that is to reduce the Eco_PI measure.

2) The decision made by the distributed multi-agent reinforcement learning algorithm is then fed back to the DT, which updates its simulation based on the decision. The updated simulation is then used to provide new input to the distributed multi-agent reinforcement learning algorithm, and the process continues until the desired objective is achieved.

This integration of DT and DGMARL avoids tedious field training of the DGMARL model in signal timing optimization, leading to efficient and safe model training and testing. The digital twin's data accumulation enables efficient visualization and analysis of the traffic state.

IV. Implementation of Intelligent Agents to Optimize the Global Transportation

Motivations: AI enabled intelligent agents can enhance transportation networks by analyzing data and making optimized decisions, leading to increased efficiency, reliability, and safety. This can have positive impacts on people's quality of life, the environment, and economic growth. Intelligent agents can optimize transportation networks by monitoring real-time traffic and making recommendations. They can suggest alternate routes to avoid congestion, adjust traffic signals to improve traffic state, and predict maintenance needs.

Graph Representation of the Transportation Network: Using graph representation along with intelligent agents can help provide comprehensive situational awareness by monitoring the entire network efficiently [18], [19]. A transportation network's graph representation uses nodes for intersections and edges for routes. Intelligent agents can use this graph to predict congestion and optimize traffic state by analyzing sensor data from key locations and communicating with local signal controllers [20]. By incorporating machine learning algorithms, such as reinforcement learning algorithms, agents can learn traffic patterns from current and historical data, detect anomalies, and optimize traffic state by controlling traffic signals to avoid congestion.

Scalability Signal timing optimization coordinates traffic signals at intersections to enhance traffic flow and reduce congestion. However, scalability is a crucial factor in this process as the transportation network's size increases along with the number of intersections and traffic signals. In a single-agent architecture, a centralized agent optimizes traffic signal timings across the network by using data from sensors and other sources. Although effective for smaller transportation networks, this architecture may face difficulties in scaling to larger networks due to increased processing, communication requirements, and latency.

In a multi-agent architecture, multiple agents optimize traffic signal timings at different intersections in the transportation network [21], [22]. Multiple agents coordinate signal timing across intersections by processing local sensor data and communicating with each other. Asynchronous communication protocols such as message passing and attention mechanisms can reduce communication overhead, making the architecture more scalable. By distributing the workload and utilizing local data more efficiently, this approach can handle larger volumes of data and more intersections.

A. Graph NN Oriented Formulation about Traffic Network

The proposed approach models the traffic environment as a network using a bi-directional graph $G(\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ represents a set of intersections modeled as agents, and $\mathcal{E}$ represents a set of roads considered links, where $e_{i,j} \in \mathcal{E}$ is a link that connects intersections i and j. The static features of each intersection i include approach links, signal controllers, signal phases, detectors, the number of lanes associated with each link, uncontrolled approaching links, and neighboring intersections $\mathcal{N}i \subset \mathcal{V}$. Each intersection's signal controller is linked to a set of signal phases ϕ_i, each of which is associated with a set of static features such as a list of signals, a minimum mandatory green serving time, yellow time, red clearance time, pedestrian recall time, and priority phase.

B. Infrastructure of DGMARL

Figure 3 shows the architecture of a DT assisted multi-agent reinforcement learning empowered traffic environment. Each intersection of the traffic network was designed as a local agent. The multi-intersection traffic network signal timing optimization problem is addressed with distributed multi-agent reinforcement learning. The traffic signal control problem is formulated as a Markov Decision Process (MDP): $(\mathcal{S}, \mathcal{A}, p, r)$ where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ represents the action space, and r is the reward that measures the benefit brought about by a specific action. The objective is to learn the optimal policy p that generates the best action for the next step and maximizes the subsequent accumulative discounted rewards produced by the action. To improve the learning efficiency of agents and choose the best actions based on approaching traffic from upstream and downstream intersections, the visibility of neighboring agents' states was increased by sharing local observations through message passing.

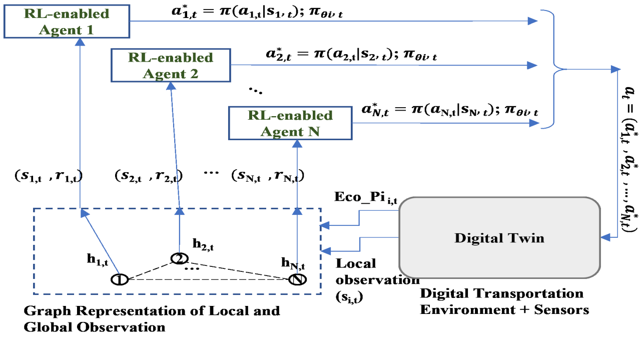

RL-enabled Agent 1
RL-enabled Agent 2
RL-enabled Agent N
$a^*_{1,t} = \pi(a_{1,t}|s_{1,t}); \pi_{\theta i,t}$
$a^*_{2,t} = \pi(a_{2,t}|s_{2,t}); \pi_{\theta i,t}$
$a^*_{N,t} = \pi(a_{N,t}|s_{N,t}); \pi_{\theta i,t}$
$a_t = (a^*_{1,t}, a^*_{2,t}, \dots, a^*_{N,t})$
$(s_{1,t}, r_{1,t})$
$(s_{2,t}, r_{2,t})$
…
$(s_{N,t}, r_{N,t})$
Eco_Pi i,t
Digital Twin
$h_{1,t}$
$h_{2,t}$
$h_{N,t}$
1
2
N
Local observation $(s_{i,t})$
Digital Transportation Environment + Sensors
Graph Representation of Local and Global Observation

Fig. 3: Digital Twin assisted DGMARL Learning

Local state Space: The state of the global traffic network at time t for the traffic network is defined as

$$S_t = \{s_{i,t}\}_{i=1}^{|\mathcal{V}|}. \quad (1)$$

where $\{s_{i,t}\}$ is the state of the intersection i at time t which is the heterogeneous observation of traffic states and traffic signal phase state.

Action Space: The actions of an intersection i are to switch or not switch from the current phase ϕ_i and are defined as

$$a'_{i,t} = \begin{cases} a_{i,t}, & \text{if } \phi d_{i,t} \geq \max(g_{i,\phi_i}, pd_{i,\phi_i}) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $a'_{i,t} = 0$ indicates that the agent does not take action, $a_{i,t}$ was the initially defined action and is evaluated against the physical constraints of the minimum green serving time g_{i,ϕ_i} of the current phase active ϕ_i at intersection i and the pedestrian serving time pd_{i,ϕ_i} based on the current phase duration $\phi d_{i,t}$, to ensure the safety of all users of the transportation network.

Reward based on Eco_PI: The reward function was formulated as $Eco_P i$ by measuring the number of stops and stop delay that occurred in every traffic approach, following an existing fuel consumption model proposed in [6], [7]. The number of stops a vehicle makes is calculated by counting the number of times the vehicle is stopped in a queue while approaching from all directions in the intersection. The stop delay is calculated as the amount of time a vehicle is stationary in the queue before it reaches the intersection. For example, as shown in Figure 4, at the Cater intersection in MLK Smart Corridor, vehicle stops and stop delays are calculated on the eastbound, southbound, westbound, and northbound approaching links. These metrics are then used to calculate the Eco_PI index, which serves as an indicator of fuel consumption related to stopping. The immediate reward r_i is calculated for each traffic movement of intersection i as

$$r_{i,t} = Eco\_PI_i = -(\sum_{l=1}^{L_i} d_{i,l,t} + (K_{i,l,t} * N_{i,l,t})) \quad (3)$$

where $d_{i,l,t}$ is the average stop delay that occurred in link l_i, $N_{i,l,t}$ is the number of stops, and $K_{i,l,t}$ is the average stop penalty to penalize every stops [8], [9]. The policy of each agent i is optimized to maximize the global long-term return $E[R_0^\pi]$, where $R^\pi_{i,t} = \sum_\tau^T \gamma^{\tau - t} r_{i,t}$ is the return at time t, with a discount factor γ.

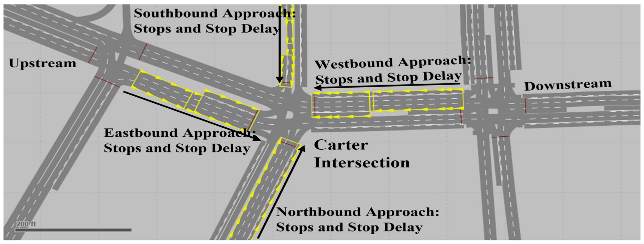


Fig. 4: Vehicles Stops and Stop Delay at each approach

Spatio-temporal Multi-agent RL: Each intersection's behavior is modeled using a decentralized graph network and state and action spaces. Agents use the Advantage Actor-Critic algorithm, with Actor and Critic designed using a graph neural network. Agents learn spatial and temporal dependencies through asynchronous communication protocols and make decisions based on their current state and policies. Policies are updated based on optimal long-term return values and evaluated and updated based on physical constraints. Each agent's state, action, and reward are communicated to the neighbors through message passing, and the reward is stored to measure the global return for each agent. Therefore, the multi-agent MDP was updated as (G, S, A, M, p, r, S') where $m_{ji,t} \in M_{ji}$ is the message passed from agent j to agent i including the states, actions, and rewards of the neighboring agent i at time t. $\mathcal{N}_i = \{j \in \mathcal{V} | ij \in \mathcal{E}\}$ represents the set of neighboring agents that are connected to agent i by links $l_{i,j}$ in the communication graph $(\mathcal{V}, \mathcal{E})$. Then the local agent state is updated as $s'_{i,t} \in S'$ which is the joint state of the agent's current state and the neighbors state.

At time t, the state $s_{i,t}$ of intersection i includes traffic state such as volume, detector occupancy, average waiting time, delay, and velocity, as well as traffic signal state such as current phase state, duration, and pedestrian recall time. The states of neighboring agents $\mathcal{N}_\rangle$ are obtained through message passing, including the aggregation of the agent's state and policy.

$$m_{i,t} = g(s_{j,t} \cup h_{j,t-1} \cup \pi_{j,t-1}, \; \forall \; j \in \mathcal{N}_i) \quad (4)$$

Then the intersection i state is updated by the linear transformation with a rectified linear function, with the dimensions of the traffic state and traffic signal state input varying for each intersection. The hidden state of temporal traffic information is extracted by the LSTM layer.

$$h'_{i,t} = \xi(s_{i,t} \cup h_{i,t-1} \cup \pi_{i,t-1} \cup m_{i,t}) \quad (5)$$

Then a linear transformation with a rectified linear function is applied to the hidden graphs to identify the optimal policy, π_i. And the softmax function is applied to generate actions a'_i. The policy is evaluated and adjusted by considering mandatory physical constraints.

A2C with a Graph Neural Network (GNN) stabilizes the learning process and enhances the performance of the proposed model in identifying the optimal policy for maximizing the expected cumulative discounted reward $E[R^\pi_{i,0}]$ over time steps for intersection i. The advantage function $A^\pi_i(s'_{i,t}, a'_{i,t})$ evaluates the benefit of taking an action $a'_{i,t}$ in a state $s'_{i,t}$ compared to the average value at that state and serves as a reference point for the action-value function $Q^\pi_i(s'_{i,t}, a'_{i,t})$. The state-value function $V^\pi_i(s'_{i,t})$ defines the predicted cumulative discounted reward from a specific state under a given policy and is calculated as the weighted sum of the action-value function for all possible actions.

The policy distribution approximates the anticipated cumulative discounted reward from taking an action in a state under the policy π_i. The advantage function helps the critic network reinforce the selection of the most suitable action by updating the policy distribution with policy gradients as directed by the critic, which in turn increases the probability of actions proportional to the high expected return $E[R^\pi_{i,0}] = \sum_{s'_{i,t} \in S'} p(s'_{i,t}) V^\pi_i(s'_{i,t})$.

Learning from experiences: During each time step, the experience replay buffer $\mathcal{D}$ stores the information including the initial state, the updated state with neighbor networks, updated policies and values, the new state after taking action, and the step reward $(s_{i,t}, a'_{i,t}, m_{\mathcal{N}_i,t}, r_{i,t}, s''_{i,t}, v'_{i,t}, \pi_{\theta_{i,t}})$. In each subsequent time interval, the model learns the temporal dependency by utilizing the batch of experiences $\mathcal{B}$ is $\{(s'_{i,\tau}, m_{N_i,t,\tau}, a'_{i,\tau}, r_{i,\tau}, s''_{i,\tau}, v'_{i,\tau}, \pi'_{\theta_{i,\tau}})\}_{i \in V, \tau \in \mathcal{B}}$ stored in the replay buffer $\mathcal{D}$ and updates the graph neural network parameters based on the calculated losses. Where $\{\pi'_{\theta_i}\}_{i \in V}$ is stationary policy and value $\{V'_{\omega_i}\}_{i \in V}$ were updated after physical constraints evaluation of intersection i. Actor loss incorporates the negative log probability of the action that was sampled under the current policy, and the actor is updated based on the estimated advantage. And the Critic loss, which involves computing the mean squared error between the sampled action-value and the estimated state-value, is updated using the estimated state-value.

C. Digital Twin Assisted Method

To reduce congestion and Eco_PI, a PTV-Vissim COM interface was embedded with DT, which represents the distributed graph-based multi-agent reinforcement learning framework using a DT to optimize traffic signal control as shown in Figure 3. The DT represents the physical transportation environment, and each intersection in the DT is mapped to a corresponding reinforcement learning agent. The DT can interact with agents through COM interface and the physical environment within a tolerable time frame. Hence, each agent maintains its optimal policy and decides the best actions to control signal phases.

DT assisted DGMARL algorithm is shown in Algorithm 1. Each intersection in the DT is mapped to a corresponding reinforcement learning agent i as shown in Algorithm 1 ensure section. At time t the agent i observes various features through DT components, such as the detector occupancy, approach level vehicle count aggregates, vehicles velocity, and current signal state. Then collaborates with its neighbors $\mathcal{N}_i$ to share and receive their states through message passing as described in Algorithm 1 line-5. And then the updated state $s'_{i,t}$ of agent i is processed through a graph neural network to derive the optimal policy π_i and select actions to control the signal phase ϕ_i (line-6). Then agent i validates the actions (line-7), against the physical constraints configured in the DT, the minimum green serving time and pedestrian recall time, to ensure user safety. If the decision is to stay in the current phase in green, then no actions are applied back to the DT; otherwise, agent i validates other phases detector occupancy and selects the phase ϕ_j that has a higher upcoming traffic occupancy, then applies the signal phase change action to the signal controller in the DT (line-8), which updates the simulation. Once the decided action is applied, each agent i estimates the current reward r_i with the new observed traffic state $s_{i,t+1}$ (line-9), and stores the experiences in replay buffer (line-10). And when the buffer resize reaches minimum batch size the agent starts to learn from the collection of experiences at ever time step to minimizes the critic loss $L(\omega_i)$ and actor loss $\hat{J}(\theta)$ (lines 12-14). The agent i repeats the above processes until it achieves the desired objective of identifying optimal policy to choose best actions for reducing congestion and Eco_PI.

Due to the distributed agent environment, each agent makes different decisions based on their local and neighboring traffic state, so the convergence of an optimal policy is different for each agent and the efficiency of learning is increased. Since agents continue to interact with the real environment through the DT, the probability of arriving at an optimal policy is faster. Hence, by using a DT and reinforcement learning, the system can adapt to changing traffic conditions in real time, leading to more efficient signal control, and it can be further optimized to maximize its benefits.

Algorithm 1 Digital Twin assisted DGMARL Learning

```
Require α learning rate, β entropy coefficient.
Ensure: Initialize graph $G(\mathcal{V}, \mathcal{E})$, agent $i \in \mathcal{V}$, link $l_i \in \mathcal{E}$,
    physical constraints $i_c$, policy network parameters $\theta$ and
    value network parameters $\omega$.
1:  for $e = 1$ to episodes do
2:    Observe state $s_i$ from Digital Twin.
3:    for $t = 0$ to $T - 1$ do
4:      for agent $i = 1$ to $\mathcal{V}$ do
5:        Update state $s'_{i,t} \approx s_{i,t} \cup \pi_{N_{i,t-1}} \cup h_{N_{i,t-1}}$ through
          message passing.
6:        Select policy $\pi_{\theta_{i,t}}$, action $a_{i,t} \approx \pi(a|s'_{i,t})$, and get
          value $v(s'_{i,t}|\omega, a_{i,t})$.
7:        Evaluate agent's actions $a'_{i,t} = (a_{i,t}|i_c)$ and update
          value $v'(s'_{i,t}|\omega, a'_{i,t})$ and policy $\pi'(a'_{i,t}|v'_{i,t}, s'_{i,t})$.
8:        Take action $a'_{i,t}$ in Digital Twin
9:        Observe reward $r_{i,t}$ and new state $s'_{i,t+1}$.
10:       Replay buffer
          $D \leftarrow (s'_{i,t}, \pi'_{\theta_{i,t}}, a'_{i,t}, r_{i,t+1}, s'_{i,t+1}, v'_{\omega,i,t})$.
11:       if $t >= B$ sample batch size $B$ then
12:         Learn from random minibatch and obtain target
            return.
13:         Update critic by minimizing the loss $L(\omega_i)$
14:         Update actor using sampled policy gradient de-
            scent along with entropy loss $\hat{J}(\theta)$.
15:       end if
16:     end for
17:   end for
18: end for
```

V. Experiments

This section provides the details of the experiment setup using a real-world dataset and optimization results that show the efficiency of the DT assisted DGMARL model.

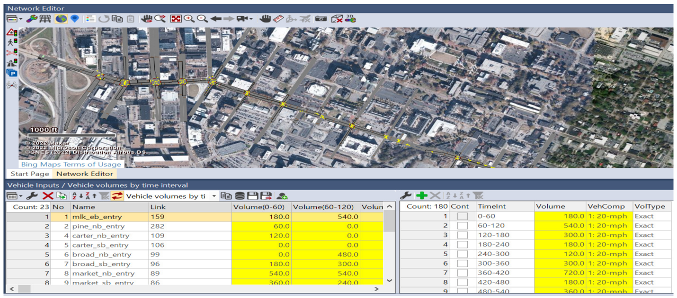
A. Experimental Setup

The experimental environment was set up using the real-world dataset collected by the Department of Computer Science and Engineering at the University of Tennessee, Chattanooga, USA [23].

Real-world dataset: The dataset is composed of the corridor that connects 11 intersections on MLK Smart Corridor with bidirectional traffic in East-West, West-East, North-South, and South-North directions, which includes intersection geometry, the traffic signal timing plans, camera and zone-detecting device parameters, SPaT, vehicle flow, velocity, detector occupancy, etc. Each intersection consists of a heterogeneous phase setup with different number of signal light configurations. To have a flexible and adaptive approach to traffic signal control, the action has been designed with two decisions: either staying in green in the current phase or switching to the phase that has the highest traffic occupancy. Before switching to the new phase, the current phase follows the configured yellow and red clearance times. The yellow and red clearance timing magnitudes are specific to for each phase at each intersection.

B. Digital Twin Setup

The Tier 1 DT platform is used in this experiment. The simulation model of 11 intersections of the MLK Smart Corridor in PTV-Vissim is developed following network creation guidelines in [24], as shown in Figure 5. The developed model is populated with archived December 15, 2022, one-minute volume counts at network entry edges, 10-minute turn percentages at each intersection approach, and signal timing plans received from the city. Two versions of the simulation model are created: 1) the PM peak model that simulates the December 15, 2022, 3:00 PM–6:00 PM scenario, and 2) the 24-hour model that simulates the December 15, 2022, 24 hour scenario. This model is prepopulated with data and runs faster than wall clock time.


Network Editor
1000 ft
© 2022 Maxar
© 2022 Microsoft Corporation
©CNES (2022) Distribution Airbus DS
Bing Maps Terms of Usage
Start Page
Network Editor
Vehicle Inputs / Vehicle volumes by time interval
Vehicle volumes by ti

Count: 23	No	Name	Link	Volume(0-60)	Volume(60-120)	Volun
1	1	mlk_eb_entry	159	180.0	540.0	
2	2	pine_nb_entry	282	60.0	0.0	
3	4	carter_nb_entry	109	120.0	0.0	
4	5	carter_sb_entry	106	0.0	0.0	
5	6	broad_nb_entry	99	0.0	480.0	
6	7	broad_sb_entry	96	180.0	300.0	
7	8	market_nb_entry	89	540.0	540.0	
8	9	market_sb_entry	86	360.0	240.0	

Count: 180	Cont	TimeInt	Volume	VehComp	VolType
1	□	0-60	180.0	1: 20-mph	Exact
2	□	60-120	540.0	1: 20-mph	Exact
3	□	120-180	300.0	1: 20-mph	Exact
4	□	180-240	180.0	1: 20-mph	Exact
5	□	240-300	120.0	1: 20-mph	Exact
6	□	300-360	300.0	1: 20-mph	Exact
7	□	360-420	720.0	1: 20-mph	Exact
8	□	420-480	180.0	1: 20-mph	Exact
9	□	480-540	360.0	1: 20-mph	Exact

Fig. 5: MLK Smart Corridor network layout in PTV-Vissim

C. Impact of the Application of the Proposed Model

Efficiency of DT assisted DGMARL is measured using the number of stops and stop delay at each intersection as the metrics. Also Eco_PI are calculated to measure the impact of fuel consumption related to stopping. DGMARL starts optimizing signal timing after a non-stationary period of 120 seconds. At each time step, the graph neural network updates each agent's current state with Relu activation in the message passing layer. Then, the actor and critic neural networks generate the value-assisted action probability. This process continues until the initial batch size of experiences is gathered. Afterward, at each time step, the model learns from experience with random samples and updates the graph neural network parameters to arrive at the optimal policy distribution. The model decay rate is customized based on the current learning episode. To achieve optimal results, the model was trained for 100 episodes using the dataset from the first hour of MLK Smart Corridor on Thursday, December 15, 2022. Each episode's simulation step was 3600 seconds, and the model learned from 240 batch sizes of experience replay in each episode for 3240 steps.

D. Experiment Results

The developed DGMARL signal timing plan was tested on MLK Smart Corridor for 24-hour and PM-peak hour scenarios of December 15, 2022. The performance of DGMARL signal timing plan was compared with the baseline actuated MLK Smart Corridor vehicle actuated signal timing plan.

24-hour scenarios: Figure 6 shows the comparison of Eco_PI index observed from DGMARL and baseline vehicle actuated signal timing plans. The overall Eco_PI improved by 44.27%, with improvements ranging from 11.23% to 81.47% over the 11 intersections. Due to the time limit, one trial was performed with a 24-hour scenario.

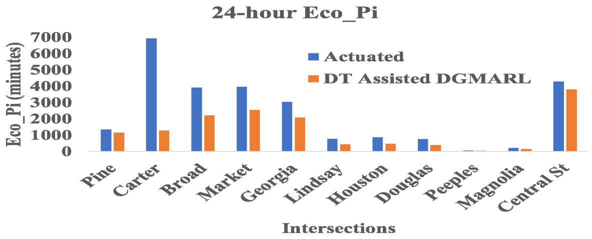


Fig. 6: 24-hour in MLK Smart Corridor: Overall Eco_PI improved by 44.27%

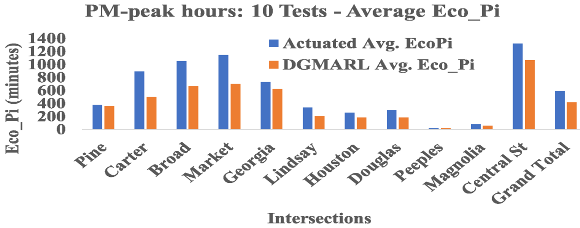


Fig. 7: PM-peak hour scenarios in the MLK Smart Corridor: an average Eco_PI of 29.88% observed from 10 tests using 10 random seeds

PM-peak hour scenarios: Results from ten replicate trials with different random seeds for PM-peak hour were compared from DGMARL and vehicle actuated signal timing plan implementation. Figure 7 shows that the average Eco_PI over

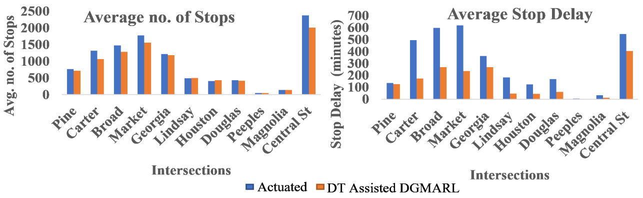


Fig. 8: PM-peak hour scenarios in the MLK Smart Corridor: an average of 10.48% stops reduced and 49.68% of stop delay reduced from 10 tests using 10 random seeds.

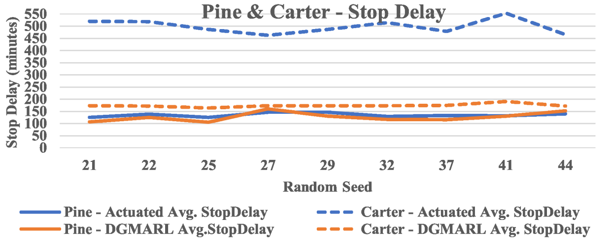


Fig. 9: PM-peak hour scenarios: best performing intersection Carter, with an average stop delay reduced by 65.01%, and at the Pine intersection, the average stop delay reduced by 5.96%

ten replicate trials, improved by 29.88%, with improvements ranging from 6.07% to 43.90% over the 11 intersections.

PM-peak hour scenarios shows an average reduction in stops by 10.48% and in stop delay by 49.68% compared to the baseline actuated signal timing scenario from the ten random seed replicate trials, as shown in Figure 8. Among all intersections, the stop delay at Carter intersection has larger improvement of 65.01%, while the least improvement of 5.96% is observed at Pine intersection, as shown in Figure 9.

Conclusions

This paper presents the use of a Digital Twin assisted graph-based, decentralized multi-agent reinforcement learning algorithm for optimizing traffic signal timing by observing traffic state in real time. By enabling multiple agents to exchange knowledge with the environment, this approach shows improvements in learning efficiency, enables performance with lower latency, and demonstrates its' ability to optimize signal timing for reduced Eco_PI and average stop delays.

In future, further development of this algorithm for varying optimization frequencies, large-scale networks, and its deployment in real-world environments will be investigated.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy under the Award Number DE-EE0009208.

References

[1] PTV.Group, "Ptv vissim," 2022. [Online]. Available: https://www.ptvgroup.com/en/solutionsproducts/ptv-vissim/

[2] ARCADIS, "Creating an intelligent transportation systems for atlanta's first smart corridor." [Online]. Available: https://www.arcadis.com/en-us/projects/north-america/united-states/north-ave-corridor

[3] J. Wu, X. Wang, Y. Dang, and Z. Lv, "Digital twins and artificial intelligence in transportation infrastructure: classification, application, and future research directions," *Computers and Electrical Engineering*, vol. 101, p. 107983, 2022.

[4] A. J. Saroj, A. Guin, and M. Hunter, "Deep lstm recurrent neural networks for arterial traffic volume imputation," *Journal of big data analytics in transportation*, vol. 3, no. 2, pp. 95–108, 2021.

[5] N. P. Farazi, B. Zou, T. Ahamed, and L. Barua, "Deep reinforcement learning in transportation research: A review," *Transportation research interdisciplinary perspectives*, vol. 11, p. 100425, 2021.

[6] A. Stevanovic, S. A. Shayeb, and S. S. Patra, "Fuel consumption intersection control performance index," *Transportation research record*, vol. 2675, no. 9, pp. 690–702, 2021.

[7] S. A. et al., "Investigating impacts of various operational conditions on fuel consumption and stop penalty at signalized intersections. international journal of transportation science and technology," 2022.

[8] A. Stevanovic and N. Dobrota, "Impact of various operating conditions on simulated emissions-based stop penalty at signalized intersections," 2021.

[9] S. Alshayeb, A. Stevanovic, and B. B. Park, "Field-based prediction models for stop penalty in traffic signal timing optimization. energies," 2021.

[10] F. Tao, H. Zhang, A. Liu, and A. Y. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on industrial informatics*, vol. 15, no. 4, pp. 2405–2415, 2018.

[11] M. Farsi, A. Daneshkhah, A. Hosseinian-Far, H. Jahankhani *et al.*, *Digital twin technologies and smart cities*. Springer, 2020.

[12] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020.

[13] A. J. Saroj, S. Roy, A. Guin, and M. Hunter, "Development of a connected corridor real-time data-driven traffic digital twin simulation model," *Journal of Transportation Engineering, Part A: Systems*, vol. 147, no. 12, p. 04021096, 2021.

[14] A. Rudskoy, I. Ilin, and A. Prokhorov, "Digital twins in the intelligent transport systems," *Transportation Research Procedia*, vol. 54, pp. 927–935, 2021.

[15] K. Zhang, J. Cao, and Y. Zhang, "Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1405–1413, 2021.

[16] S. Dasgupta, M. Rahman, A. D. Lidbe, W. Lu, and S. Jones, "A transportation digital-twin approach for adaptive traffic control systems," *arXiv e-prints*, pp. arXiv–2109, 2021.

[17] J. Maroto, E. Delso, J. Felez, and J. M. Cabanellas, "Real-time traffic simulation with a microscopic model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 513–527, 2006.

[18] J. Li, H. Ma, Z. Zhang, J. Li, and M. Tomizuka, "Spatio-temporal graph dual-attention network for multi-agent prediction and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 556–10 569, 2021.

[19] J. R. Palit, "Application of machine learning and deep learning approaches for traffic operation and safety assessment at signalized intersections," *University of Tennessee at Chattanooga*, 2022.

[20] M. A. Basmassi, S. Boudaakat, J. A. Chentoufi, L. Benameur, A. Rebbani, and O. Bouattane, "Evolutionary reinforcement learning multi-agents system for intelligent traffic light control: new approach and case of study," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 5, p. 5519, 2022.

[21] T. Chu, S. Chinchali, and S. Katti, "Multi-agent reinforcement learning for networked system control," in *International Conference on Learning Representations*.

[22] Y. Wang, T. Xu, X. Niu, C. Tan, E. Chen, and H. Xiong, "Stmarl: A spatio-temporal multi-agent reinforcement learning approach for cooperative traffic light control," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 2228–2242, 2020.

[23] A. Harris, J. Stovall, and M. Sartipi, "Mlk smart corridor: An urban testbed for smart city applications," in *IEEE International Conference on Big Data*, 2019, pp. 3506–3511.

[24] GDOT, "Vissim simulation guidance," 2021-06-01, FHWA-GA-21-1833, 18-33. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/60642