e-SMARTS: A System to Simulate Intelligent Traffic Management Solutions (Demo Paper)

Udesh Gunarathna
pgunarathna@student.unimelb.edu.au
The University of Melbourne
Australia

Renata Borovica-Gajic
renata.borovica@unimelb.edu.au
The University of Melbourne
Australia

Shanika Karunasekera
karus@unimelb.edu.au
The University of Melbourne
Australia

Egemen Tanin
etanin@unimelb.edu.au
The University of Melbourne
Australia

ABSTRACT

Intelligent traffic management solutions that leverage machine learning have gained a lot of interest in recent years. These techniques, however, cannot be deployed in real-world settings at a desirable pace due to technological barriers. Thus, easily customizable, realistic simulation environments are needed to train and verify the effectiveness of machine learning algorithms for traffic control. We propose an easily extendable traffic simulation system named e-SMARTS to allow researchers to experiment with novel data-driven traffic management algorithms in a setup that mimics real-world traffic conditions. We demonstrate the flexibility of e-SMARTS using widely researched traffic management solutions for Autonomous Intersection Management (AIM). In the demonstration, we present several pluggable algorithms for AIM and show that these computationally efficient algorithms can achieve effective and safe results.

CCS CONCEPTS

• Networks → Traffic engineering algorithms; **• Computing methodologies → Reinforcement learning**.

KEYWORDS

Simulations, Spatial-temporal data, Reinforcement learning

ACM Reference Format:

Udesh Gunarathna, Renata Borovica-Gajic, Shanika Karunasekera, and Egemen Tanin. 2022. e-SMARTS: A System to Simulate Intelligent Traffic Management Solutions (Demo Paper). In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3557915.3560955

1 INTRODUCTION

With the emergence of Connected Autonomous Vehicles (CAVs) and intelligent traffic infrastructures, AI techniques and research have found a fertile application domain in traffic engineering [7, 9, 13]. Related traffic management solutions often require making predictions or mining patterns to learn how to control the traffic elements. In recent years, Machine Learning (ML), and in particular Deep Learning and Reinforcement Learning (RL), have become a popular choice for many data-driven solutions for traffic optimization [1, 5, 14]. These new algorithms, however, need to be simulated in an environment that resembles real-world conditions before being deployed to actual road networks. To address this issue, we propose an easily extendable microscopic traffic simulation system where researchers can implement and verify different algorithms for traffic management and observe their impact instantaneously.

We build upon Scalable Microscopic Adaptive Road Traffic Simulator (SMARTS) [11], which can simulate large traffic networks compared to other existing simulators [3, 8, 16]. We extend SMARTS with the capability to plug and play various traffic optimization solutions, naming the new system extendable-SMARTS (or e-SMARTS for short). In e-SMARTS, simulation objects such as vehicles, intersections, and road segments are treated as *intelligent agents* that can communicate with each other. All simulation objects contain a general interface where researchers can plug in new algorithms to control the behaviour. Since each object is able to communicate with others, coordination among different algorithms is a possibility. Furthermore, machine learning algorithms can interact with these objects to learn dynamics or hidden patterns without relying on explicit models. We showcase the capabilities of our system using a novel distributed RL solution for Autonomous Intersection Management (AIM) [2].

AIM solutions are developed to replace the traditional traffic signal control systems with intelligent solutions. In AIM, each CAV arriving towards an intersection coordinates their arrival time to the intersection via an intersection controller. In AIM, the set of vehicles and the intersection controller can be thought of as two types of intelligent agents, and different algorithms can be used for both of these agents.

Using e-SMARTS for AIM, one can implement different scheduling algorithms for the intersection controller to coordinate the arrival times of vehicles. Then, different trajectory control algorithms for vehicles can be explored in order to adjust speeds and adhere to the arrival schedule of the intersection controller. Decoupling these two components allows researchers to try different combinations of algorithms, as both components communicate with each other, while algorithms can receive real-time traffic data from

SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9529-8/22/11.
https://doi.org/10.1145/3557915.3560955

the simulation if they require it. As AIM needs to coordinate vehicles in real-time, the computation times are of critical importance for the overall solution's safety and performance. Unfortunately, previous efforts exhibit high computational time as they rely on Linear Programming (LP) [10] and are not designed to learn the simulation dynamics.

In this work we demonstrate two computationally efficient algorithms for intersection and trajectory control using e-SMARTS that can learn the simulation dynamics. We refer to this combination by the name Intelligent Autonomous Intersection Management (I-AIM) (refer to [4] for the detailed version of I-AIM). The first solution is a polling-based coordinating agent (*a.k.a.* intersection controller) for the intersection. The second component consists of a set of distributed Reinforcement Learning agents, which are assigned to each vehicle. Whenever vehicles are within a certain distance from the intersection (called control region), the coordinating agent communicates with the RL agents to schedule the arrival times of those vehicles to the intersection. The coordinating agent uses a novel polling-based algorithm to handle multi-lane intersections with multiple turning directions. Once the coordinating agent determines a time schedule, an RL agent controls each vehicle's trajectory to adhere to the coordinating agent's time schedule. As we show in our demonstrations, the RL agent can control trajectories much faster than the state-of-the-art methods. Also, RL agents are able to learn vehicle dynamics through e-SMARTS, enabling safe trajectories as a result, unlike other methods we compared against.

Our microscopic simulations in a complex intersection serve four key purposes: (1) We show that e-SMARTS can be successfully integrated with futuristic traffic management solutions by allowing users to try their own implementations, even during the demonstration. (2) We demonstrate that I-AIM can avoid the stop-and-go nature of traditional dynamic traffic signals and reduce heavy traffic congestion. (3) Our demonstrations show that CAVs achieve platooning-like behavior with I-AIM, which further eases the congestion. (4) We demonstrate that conventional approaches may pose safety risks due to high computational time and the inability to learn simulation dynamics. We then present how RL-based methods mitigate these problems. Leveraging the proposed interfaces, other machine learning algorithms can learn dynamics of the simulation environment to accelerate the development and deployment of solutions to real-world road networks.

2 EXTENDABLE SCALABLE MICROSCOPIC ADAPTIVE ROAD TRAFFIC SIMULATOR

SMARTS is a microscopic traffic simulator with a distributed architecture developed in Java that can simulate very large traffic networks. The simulator uses a custom-made distributed architecture that allows simulation acceleration with multiple processors. The architecture consists of a server processor and an arbitrary number of worker processors. When setting up a simulation, the server can partition the entire simulation area into multiple sub-areas as a grid-based or sub-graph-based approach. Grid-based deployment is useful when there are many subdivisions and processors, while the subgraph-based deployment is preferable with a fewer number of nodes and subdivisions. Each sub-area is simulated by a different worker. All the workers run in parallel to shorten the simulation time. The distributed architecture enables simulations of large cities with dozens of processors. The distributed system has been tested on Nectar Research Cloud with up to 100 workers.

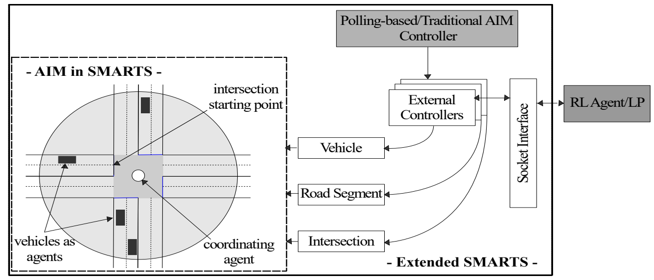

Figure 1: The overall architecture of e-SMARTS. A Coordinating Agent is virtually located in the center of the intersection and provides time schedules for vehicles within the control region (gray circle). The vehicles are considered as a set of distributed RL Agents.

We build upon SMARTS and provide interfaces to each simulation component (e.g. intersections, vehicles, or road segments) to be instructed through external inputs (thus named e-SMARTS), where each component in the simulator can be thought of as an intelligent agent. e-SMARTS enables researchers to experiment with novel traffic management algorithms through a simple plug and play mechanism and immediately observe the algorithm's impact.

Figure 1 shows the overall e-SMARTS architecture. The *AIM in SMARTS* box showcases one simulation environment. e-SMARTS can access the simulation objects from that environment, such as vehicles, road segments, and intersection simulation objects. We designed a set of interfaces named *External Controllers* which contain a general interface for each simulation object. An interface allows access to the simulation object attributes and permits to send control instructions back to the simulation object. For example, one can experiment with and test an intelligent driver model by connecting to the vehicle interface. The grey-colored boxes show such external algorithms that can be connected to these interfaces in the AIM setting. The algorithms can be connected to an interface as a Java object itself in the simulator or as an external algorithm through socket programming implemented using ZeroMQ (https://zeromq.org/socket-api). The use of socket programming allows researchers to develop algorithms in a language of their choice. In addition, machine learning predictive algorithms can first be plugged into the e-SMARTS interface to acquire data for training. Data can be collected over a longer time interval, such as a day or a week. Once trained, the algorithm can be connected to the same interface for testing over different time intervals (short or long), e.g., traffic prediction algorithms can measure the accuracy of predictions by connecting to e-SMARTS [6]. Also, RL-based algorithms can be connected to the same interface where researchers can take actions on e-SMARTS and immediately observe the impact in the next simulation cycle. The ability to connect various algorithms allows researchers to verify and deploy their solutions faster.

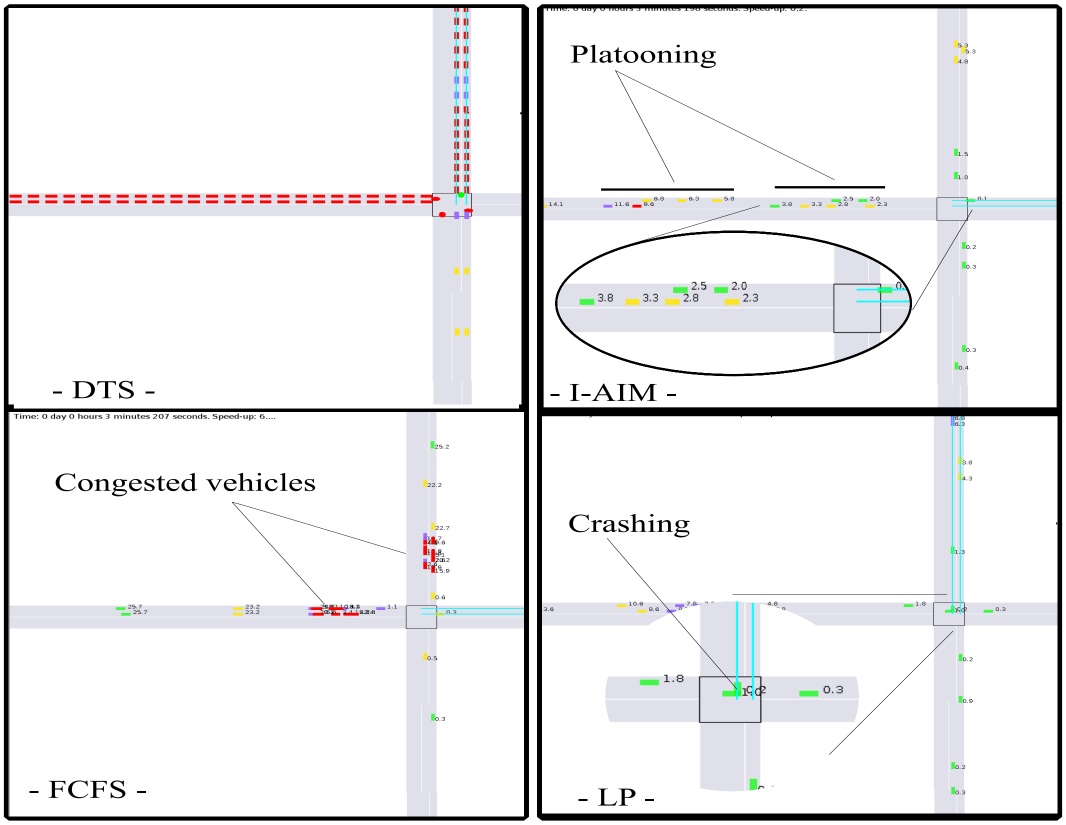

Figure 2: Four intersection scenarios in e-SMARTS are shown. The first snapshot shows the intersection controlled by a traditional dynamic traffic signal system. The second snapshot shows the same intersection controlled by I-AIM. The time to reach the intersection is shown at the top of each vehicle. The third snapshot shows an intersection controlled by an FCFS-based controller which is unable to handle high traffic load. The fourth snapshot shows an LP-based trajectory with crashing due to not reaching the intersection at the scheduled time.

3 INTELLIGENT AUTONOMOUS INTERSECTION MANAGEMENT

We first provide a brief overview of a computationally efficient solution for AIM that is used in the demonstration. We demonstrate two computationally efficient algorithms for vehicles and intersections that improve traffic flows and adhere to safety requirements. We can reduce the computational complexity of the overall optimization problem for AIM by decoupling it into two separate sets of agents, which act cooperatively through e-SMARTS. Two optimization problems are: (a) scheduling optimization and (b) trajectory optimization. Our solution, named **Intelligent Autonomous Intersection Management (I-AIM)** [4], consists of a *Polling-based Coordinating* agent, and a set of distributed *RL-based Trajectory Control Agents*. The Coordinating agent schedules arrival time at the intersection for every vehicle in the control region, while Trajectory Control Agents control the trajectory of vehicles so that each vehicle reaches the intersection precisely at the scheduled time given by the Coordinating Agent at the maximum possible speed [15].

Polling-based Coordinating Agent: The Polling-based Coordinating Agent models the intersection as a polling system. A polling system consists of a set of queues with elements stored on a First-Come-First-Serve (FCFS) basis. The polling system can select an element from one of the queues to process, and the processing time is known as *service time*. Once an element has been processed, then the polling system can select another element from the same queue or from a different queue. The switching time between queues is known as *switch-over time*. Each incoming lane of an intersection is modeled as a queue and the service time is the time that a vehicle takes to cross the intersection square. The switch-over time is the time that a vehicle needs to wait when a vehicle from a different queue is crossing the intersection. Using the above formulation, past work [10] has modeled a simplistic intersection with one lane without turning. In this demonstration, we model a more complex intersection with a modified polling system where each lane is modeled as multiple queues, which allows handling multiple turning directions. A queue-dependent switch-over time is used to allow multiple lanes in a road segment.

RL-based Trajectory Control Agent: Once the Coordinating Agent schedules an arrival time for each vehicle, we use Q-learning to control the vehicle's speed so that the vehicle reaches the intersection region precisely at the scheduled time (*trajectory control* objective). While doing so, we need to ensure that the vehicle is not going to crash with the former vehicle in the same lane (*cruise* *control* objective). The first objective is a long-term objective as this is our end goal, and the second is a short term because we need to consider crashing with the former vehicle in the near future. Combining these two objectives by combining two reward functions from each task [12] is a non-trivial task. This is because the existing Q-learning algorithm can be set either as a short-term or as a long-term objective task using the *discount factor*. In our work, we adaptively change the discount factor using the rewards from the trajectory and cruise control tasks to achieve both objectives with one single RL agent.

e-SMARTS can be easily executed and evaluated on other traffic management algorithms similar to the above AIM solution. For example, a dynamic lane allocation system can be connected to road segments in the simulation (via external interfaces) and, based on the observed traffic, the number of lanes for each direction can be changed through the external interface. The travel time changes due to lane changes can be obtained from the simulation output.

4 DEMONSTRATION

Our demonstration setup[1] in e-SMARTS consists of a four-legged intersection with 2-lane road segments similar to Figure 1. Snapshots from e-SMARTS are shown in Figure 2.

1) Plug and play APIs. In the first scenario, apart from the default algorithms, we will allow users to program and try their own algorithms for both the vehicles and intersection. The users can use either the Python or Java interface to write algorithms, and a basic set of examples will be given as building blocks. This allows users to understand how to design AIM algorithms and observe their effectiveness.

2) Stop and go nature. The second demonstration scenario (in Figure 2) shows the superiority of I-AIM over dynamic traffic signals (DTS). We use the same traffic conditions in both scenarios and demonstrate the detrimental effect of the stop-and-go nature of the vehicles with DTS which results in long queues and overall longer travel time.

3) Platooning behaviour. The third demonstration scenario shows that compared to the traditional AIM, I-AIM can improve the traffic flow by platooning vehicles. In the traditional AIM, vehicles traverse the intersection on a FCFS basis, which is inefficient. The simulations demonstrate that the Polling-based intersection controllers allow platooning behavior which substantially reduces the traffic congestion at intersections (in Figure 2). The Polling-based controller's computational complexity is linear to the number of vehicles, which allows applying the algorithm in real-time.

4) Linear Programming. The last demonstration scenario compares the RL-based trajectory control against the LP-based trajectory control. The LP-based agent is programmed through the Python interface, the same as the RL-based agent. The LP-based modelling relies on analytical equations that cannot precisely capture the intricacies of real-time traffic dynamics. This is because typically vehicles tend to have slight deviations from the intended path computed by LP due to changes in traffic conditions. Consequently, LP-based methods tend to violate the switch-over time, thus posing a safety risk as shown in Figure 2 (far right). In contrast, the RL-based agent reaches the intersection at the exact scheduled time within a one-second interval throughout the entire simulation. This added safety level for the RL-based Agents is attributed to the fact they are trained in the same simulation environment, allowing it to learn the vehicle and simulation dynamics. Furthermore, since RL works iteratively, the RL-based agent can adjust the vehicle acceleration even when vehicles are slightly deviating from the intended trajectory.

[1]The demo video is available here: https://go.unimelb.edu.au/3oje

5 CONCLUSION

This paper presents a demonstration of a flexible traffic simulator called e-SMARTS that allows researchers to implement and verify their novel traffic management algorithms in a setup that mimics real-life traffic conditions. e-SMARTS considers all simulation components to be intelligent entities, enabling users to implement and test AI-based traffic management approaches for emerging autonomous vehicles, thereby paving the way for intelligent data-driven traffic control.

REFERENCES

[1] Hua Wei Chach Che, Nan Xu, et al. 2019. Toward A Thousand Lights: Decentralized Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *AAAI Conference on Artificial Intelligence* (2019).

[2] Kurt Dresner and Peter Stone. 2008. A Multiagent Approach to Autonomous Intersection Management. *Journal of Artificial Intelligence Research* 31 (2008), 591–656.

[3] PTV Group. 2020. Traffic Simulation Software (VISSIM). www.myptv.com/en/mobility-software/ptv-vissim. [Online; accessed 10-June-2022].

[4] Udesh Gunarathna, Shanika Karunasekera, Renata Borovica-Gajic, and Egemen Tanin. 2022. Real-Time Intelligent Autonomous Intersection Management Using Reinforcement Learning. In *IEEE Intelligent Vehicle Symposium*. 135–144.

[5] Udesh Gunarathna, Hairuo Xie, Egemen Tanin, Shanika Karunasekara, and Renata Borovica-Gajic. 2021. Real-Time Lane Configuration with Coordinated Reinforcement Learning. In *Machine Learning and Knowledge Discovery in Databases*. 291–307.

[6] Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai, and Weiyang Kong. 2020. LSGCN: Long Short-Term Traffic Prediction with Graph Convolutional Networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. 2355–2361.

[7] Qize Jiang, Jingze Li, Weiwei Sun, and Baihua Zheng. 2021. Dynamic Lane Traffic Signal Control with Group Attention and Multi-Timescale Reinforcement Learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. 3642–3648.

[8] Daniel Krajzewicz, Georg Hertkorn, Christian Rössel, and Peter Wagner. 2002. SUMO (Simulation of Urban MObility)-an open-source traffic simulation. In *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*. 183–187.

[9] Siyi Li, Tianbo Liu, Chi Zhang, Dit-Yan Yeung, and Shaojie Shen. 2018. Learning Unmanned Aerial Vehicle Control for Autonomous Target Following. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. 4936–4942.

[10] D. Miculescu and S. Karaman. 2020. Polling-Systems-Based Autonomous Vehicle Coordination in Traffic Intersections With No Traffic Signals. *IEEE Trans. Automat. Control* 65, 2 (2020), 680–694.

[11] Kotagiri Ramamohanarao *et al.* 2016. SMARTS: Scalable Microscopic Adaptive Road Traffic Simulator. *ACM TIST*. 8, 2, Article 26 (2016), 22 pages.

[12] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. 2013. A Survey of Multi-Objective Sequential Decision-Making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113,.

[13] Guni Sharon. 2021. Alleviating Road Traffic Congestion with Artificial Intelligence. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. 4965–4969.

[14] Yuanyuan Wu, Haipeng Chen, and Feng Zhu. 2019. DCL-AIM: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles. *Transportation Research Part C: Emerging Technologies* 103 (2019), 246–260.

[15] Y. Zhang, A. A. Malikopoulos, and C. G. Cassandras. 2017. Decentralized Optimal Control for Connected Automated Vehicles at Intersections Including Left and Right Turns. In *IEEE CDC*. 4428–4433.

[16] Ming Zhou, Jun Luo, Julian Villella, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, , et al. 2020. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. *arXiv preprint arXiv:2010.09776* (2020).