Proceedings of 7th Transport Research Arena TRA 2018, April 16-19, 2018, Vienna, Austria

Developing an open-source platform for the evaluation of intelligent traffic control algorithms

Bo Gao [a,1], Bani Anvari [a], Christos Tsotskas [b], Patrizia Franco [b], Simon Box [a]

[a] *Transportation Research Group, Boldrewood Innovation Centre, University of Southampton, Southampton, SO16 7QF, UK*
[b] *Transport Systems Catapult, The Pinnacle, Milton Keynes, MK19 1BP, UK*

Abstract

Intersection management is a key component of road transport systems. Envisaging a new age of road transport systems accommodating intelligent, connected, and autonomous vehicles, many novel intersection control algorithms have been proposed in the literature. These algorithms are often implemented using bespoke software and tested over custom built network models because of their complexity and the lack of freely accessible software tools. This in turn makes them difficult to evaluate and benchmark.

To solve this issue, in this paper, we present the Traffic Control Test Bed project, the objective of which is to develop an open source microsimulation platform for the evaluation of intersection control algorithms. The platform provides a library of road network models together with an intuitive synthetic road network generator for user-defined layouts. It facilitates and streamlines the parallel execution of simulations. Outputs and performance indicators are monitored and visualised by the platform both during runtime and at post processing stage. We demonstrate the usage of the platform with a case study evaluating two simple signal optimisation methods. As well as being an arena for traffic control algorithms, the open source property of the platform also invites contributions from the wider research community to improve execution validity and efficiency of traffic control systems.

Keywords: microscopic simulation; open source; intersection management; traffic control

Project Repository: https://github.com/intelaligent/tctb

[1] * Corresponding author. Tel.: +44-23-8059-5869; fax: +44-23-8059-3152.
E-mail address: b.gao@southampton.ac.uk

1. Introduction

Efficiency of road traffic control systems has significant implications to the society. Recent developments in sensors, wireless communication, and artificial intelligence algorithms inspired a wave of innovative traffic control systems. Because of the complexity of these control systems researchers often opt to build their own simulation software to best suit the requirements of their work. The implementations of these systems are also constrained by the availability of commercial traffic simulation software and the technical limitations of open-source simulation platforms. For these reasons, traffic control systems proposed in the literature are often difficult to compare beyond theoretical merits.

In order to solve this issue, the Traffic Control Test Bed (TCTB) project has been developed to provide an open source benchmarking platform on which different traffic control systems can be evaluated side by side. As shown in Fig. 1, the objective of TCTB is to bridge the gap between high level questions posed by policymakers of road transport systems on behalf of the society, such as selecting the best traffic control system for a given location (which we demonstrate with a use case in section 4), and the underlying technical capabilities available to the development of traffic control systems, such as that of the open source microsimulation software SUMO by Krajzewicz et al. (2012). Through TCTB, researchers of traffic control systems are also able to improve their work in iterations of implementation and testing.

Furthermore, as shown in Fig. 1's Data layer, as well as facilitating the process of evaluating control systems, TCTB also provides a library of road network models to standardise the process which ensures that the evaluation results are comparable outside the context of each evaluation process. Performance statistics collected from each simulation allows control systems to be ranked in terms of different performance metrics such as overall road network throughput, waiting time, harmful gas emission, and fuel consumption etc..

Additionally, the TCTB platform also reduces the steep learning curve posed to researchers who are new to microsimulation software. With TCTB, instead of learning how to build a road network model, the user already has access to the model library included in TCTB. The default settings embedded in TCTB means that the user is able to start a couple microsimulation sessions in parallel and observe the evaluation process by executing one script without any manual configuration. The configuration file used by TCTB to customise a benchmarking session is intuitive to understand (as shown in Fig. 5), and because it is in XML format, users of SUMO will be familiar with its structure.

Above all, as SUMO, the TCTB platform is open source and freely accessible to the research community via our repository (https://github.com/intelaligent/tctb). By doing so, we hope to build a community of researchers from all over the world to keep improving the efficiency of traffic control systems.

In the next section, we discuss the background of TCTB and how it was inspired by similar platforms from other fields of research. In section 3, we present the architecture of TCTB and give details of its software components. We demonstrate the usage of TCTB with a case study in section 4. Conclusion and future work are discussed in section 5.

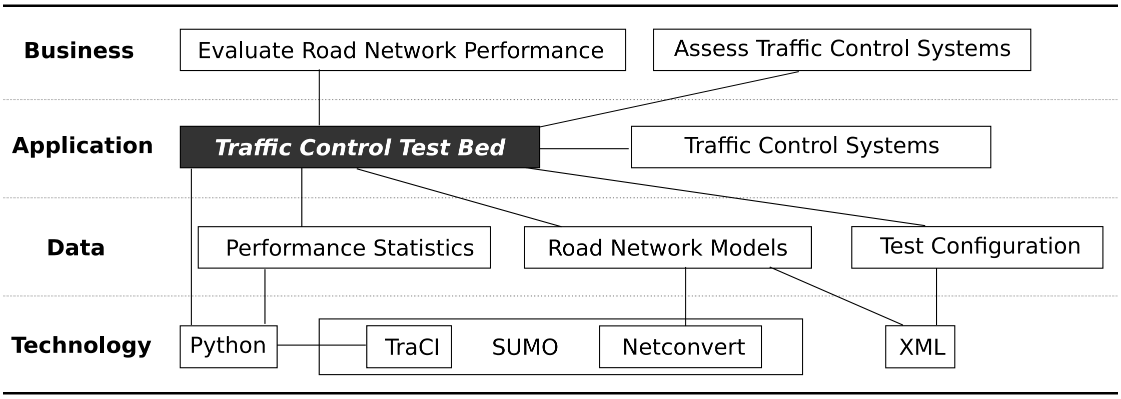


Fig. 1 Traffic Control Test Bed bridging business requirements with underlying data and technology

2. Background and Related Work

Papadimitriou and Tsitsiklis (1999) has proved that network wide traffic signal control is one of the most computationally complex problems known. For other complex problems such as the Travelling Salesman Problem (TSP) which is known to be NP-hard, good heuristics have been developed by the research community via platforms such as the National TSP Collection and the World TSP hosted by the University of Waterloo (2013). At the core of these platforms are openly accessible libraries of standardised TSP problems. The most recent copy of the World TSP dataset includes 1,904,711 cities of the world with given latitude and longitude information with a standard cost function, all freely available to the research community of TSP. Other examples of such research community platforms include the USC-SIPI image database hosted by the University of Southern California (since 1977), and the Trading Agent Competition by Wellman et al. (2001).

In contrast, evaluations of traffic control systems are usually carried out on ad-hoc traffic simulation models built by the researcher, modeling a section of road network near their university. For instance, the multi-agent reinforcement learning adaptive traffic signal control (MARLIN-ATSC) system proposed by El-Tantawy et al. (2013) was evaluated on a model of downtown Toronto. The multi-objective control algorithm based on reinforcement learning (Multi-RL) for traffic signal control proposed by Houli et al. (2010) was tested on a urban road model of Beijing’s Second Ring road. Box and Waterson (2013) applied temporal difference reinforcement learning method to train traffic controllers and tested the system with a synthetic T-junction and a road model of Southampton. A synthetic four junction network with three signalised junctions were used in Box et al. (2012). Camponogara and Kraus (2003) tested reinforcement-learning agents over a synthetic road network model including two four-approach intersections. Dresner and Stone (2008) implemented their multi-agent approach to autonomous intersection management using custom software and demonstrated their result using a road network including a single intersection.

Without a common evaluation platform, results of these work are difficult to compare. As Vlahogianni et al. (2014) concluded at the end of their review of short-term traffic forecasting studies, that an openly accessible test bed, is essential for future developments of advanced traffic control systems using modern technologies such as new sensors, communications, and data processing algorithms, which is the objective of TCTB.

To the best of our knowledge, TCTB is the first work to meet this need of a commonly accessible test bed for traffic control systems. Fok et al. (2012) developed a hardware test bed including one intersection and multiple robotic vehicles equipped with sensors and wireless communication components to evaluate autonomous intersection control systems in a physical environment. The iTETRIS project by Rondinone et al. (2013) is built on top of SUMO and ns3. Its primary objective is to build a platform for the development of cooperative control systems using wireless communication technologies. The objective of the Traffic Control Testbed platform is to evaluate controls systems side by side, with the same set of performance parameters, hence providing a global overview on the performance of different algorithms.

3. Platform

3.1. Benchmarking / Testing Workflow

Fig. 2 illustrates the workflow of TCTB. From the library of network models and traffic control systems / agents, the user begins with selecting the models and systems in interest and specify this in a configuration file (see example shown in Fig. 5). A list of performance indicator monitors is also specified in the test configuration.

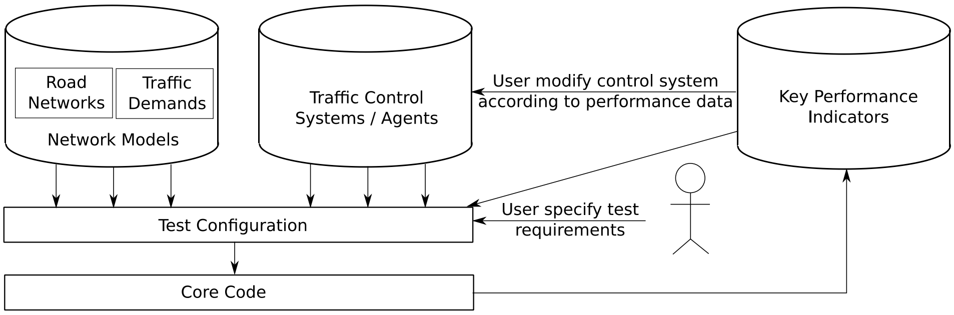

Road Networks
Traffic Demands
Network Models
Traffic Control Systems / Agents
User modify control system according to performance data
Key Performance Indicators
User specify test requirements
Test Configuration
Core Code

Fig. 2 Traffic Control Test Bed workflow

The core code of TCTB then initialises the simulation process by reading the test configuration prepared by the user. Key performance indicators such as network throughput, vehicle waiting time, and harmful gas emission are collected and saved in a database. Note that some performance indicators such as network throughput are visualised during runtime alongside the SUMO GUI to help the user understand the relation between the control system's performance to the microsimulation actions that took place in the road network. On the other hand, some other performance indicators such as vehicle waiting time along road section is better understood when statistics are drawn over a period of simulation time, and are thus visualised at the end of the simulation process. This also reduces delay in runtime GUI updates.

At the end of the workflow, the user is able to make adjustments to the algorithm applied in the agent representing the traffic control system according to the results produced by the workflow. Comparisons may be made to other control systems or anticipated numerical values of the system under development. The user group include both researchers of traffic control systems developing new systems and a policymakers who is trying to select an appropriate control system for a given location.

3.2. Network Model Library

In order to test new traffic control systems, one common approach applied in literature is to use synthetic / abstract road network layouts such as a simple cross intersection as in Dresner and Stone (2008), a corridor of intersections as in Camponogara and Kraus (2003), or a collection of such network models as in Rafter, Anvari, and Box (2017). We refer the interested reader to Chen and Englund (2016) for a more comprehensive review of traffic control systems.

To researchers new to this field, building a simple T junction model compatible with a simulation software such as SUMO is a time consuming task. Therefore, in TCTB, we provide a library of synthetic network models so that the user can get started straightaway. An online browser of these models together with their dimensions is available online (https://intelaligent.github.io/tctb/t1_model_browser.html). Each model is also represented with an interactive logo in SVG format so that the user can identify the name of the intersections without reading the network's code. In addition to this library of synthetic road models, we also provide a network generation functionality so users may generate a road network model by specifying a network definition matrix. This process is documented online (https://intelaligent.github.io/tctb/t1_network_definition_matrix.html) and we refer the interested user to our repository for more details of this functionality.

We also include real world network models such as the one used in section 4 in the model library. Because of the complexity of building these models, the majority of models included in the library are synthetic. We welcome contributions from the research community to enrich this library.

3.3. Core Code Architecture

Fig. 3 gives the UML diagram of Traffic Control Test Bed's core code which is responsible for executing the simulations specified by the user in parallel. The software has five key functional components:

- A **scenario** gives description of a simulation instance specified by the user. Properties of a scenario includes:
    - The road network this scenario is based on
    - The traffic demand definition to be used in this scenario
    - The control system/agent applied in the scenario to manage traffic

  Each test, as defined by one configuration file (XML format, see example shown in Fig. 5), includes multiple scenarios. Scenarios within the same test may share many properties, and differ only in the details of one of the properties such as the traffic control system used (see use case presented in section 4). The user can also choose to apply the same control system to different networks, or the same network but with different traffic demand.

- A **connection** object corresponds to one SUMO instance established between the TCTB and SUMO. A TraCI (Wegener et al. (2008)) instance is contained within a connection object. Using this instance, TCTB is able to synchronise the execution of all scenarios included in the test, and also collect performance data during runtime. This TraCI instance is also exposed to the traffic control system for runtime dynamic traffic management actions.

- A **demand** object generates traffic demand according to user configuration at the beginning of the test. The TCTB platform currently provide the interface of two types of demand generations:
    - Random. This is done by executing the work of Krajzewicz, Erdmann and Behrisch (2010). We set the fringe-factor to be 100 to enforce traffic to originate and end nearing the edges of the network. The purpose of this functionality is to get a simulation running without having to specify traffic zones and flows.
    - Origin-Destination defined. This requires the definition of Traffic Assignment Zones (TAZs) corresponding to the network model, and origin-destination traffic flow definition in either V-, O-, or Amitran formatted OD definition. SUMO's OD2TRIPS binary is used to generate traffic demand using TAZ and OD definitions. The use of Amitran format together with a TAZ definition file can be observed in Fig. 5.

- An **agent** represents the algorithm of a traffic control system in a simulation. TCTB currently provides interfaces to two traffic light optimisation algorithms:
    - tlsCoordinator by Taraz and Erdmann (2015) which improve the traffic flow by adjusting the offsets between traffic lights along the same route.
    - tlsCycleAdaptation by Floetteroed (2017) which improves the traffic flow by adjusting the length of the traffic light phases.

  Both algorithms are part of SUMO's toolbox. We demonstrate the effects of both algorithms using results of our case study in section 4.3. Future work of TCTB will include implementations of other control systems.

- A **monitor** is responsible for collecting performance data both during and after the test. Performance metrics provided by SUMO are all accessible via TCTB including vehicle waiting time, harmful gas emission, fuel consumption etc. TCTB also visualise performance data during runtime.

Each of these five components is managed by a controller class to manage objects of that component's class. For instance, the ConnectionManager class holds a list of connections. This architecture provides a good degree of encapsulation within the code, ensuring good maintainability of the code base.

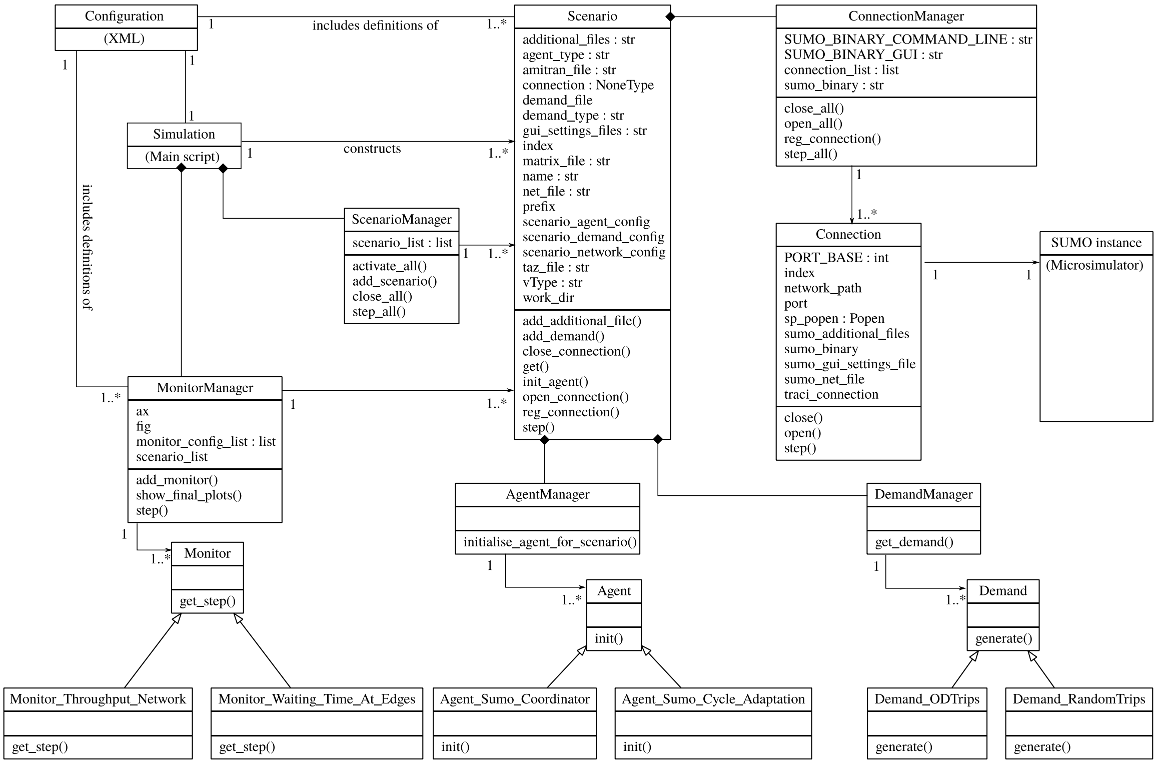


Fig. 3 UML diagram of the core code component of Traffic Control Test Bed

4. Experiment

In order to demonstrate the usage of TCTB, in this section, we present a use case of the platform the objective of which is to reduce the waiting time between Junction 14 of the M1 motorway and the Northfield roundabout travelling towards the city centre of Milton Keynes during morning rush hours. This stretch of the road is a bottleneck position because of its location. Using the methods presented in this paper, in this experiment, we apply two traffic control algorithms to take full control of the traffic lights immediately after this bottleneck location, and present benchmarking results produced by our platform.

Note that our objective for this experiment is to assess if the platform is able to produce benchmarking results for different traffic control systems under identical circumstances. Rough estimations were made to produce a scenario similar to that of a morning rush hour at this location. Both the traffic light plan and traffic demand used in the road network models in this experiment are not calibrated according to real world data. Results from this experiment validates the platform rather than the traffic control system in place at this location in the real world.

4.1. Problem Definition

First opened in 1959, the M1 is a motorway connecting London in the south to Leeds in the north of England. Junction 14 of the M1 leads to the city of Milton Keynes. As shown in Fig. 4, vehicles heading to Milton Keynes city centre originate from four main source locations including M1 southbound, M1 northbound, Cranfield and Broughton. All of these vehicles have to go pass Northfield roundabout which is located immediately after Junction 14 in the direction to Milton Keynes city centre. Because of this road layout, the section of road going from Junction 14 to the Northfield roundabout becomes a bottleneck during morning rush hours. Therefore the traffic light plan applied at the Northfield roundabout is critical in producing a smooth traffic flow towards Milton Keynes city centre.

Four sets of traffic lights control the traffic flow at each of the four entry points of the Northfield roundabout as highlighted in Fig. 4. Each set controls the traffic flow towards this roundabout from one of four directions. Each set must choose to let through either vehicles that are already on the roundabout or those arrived from the direction controlled by the set to entre the roundabout. Giving priority to those already on the roundabout will help clearing traffic that came from the other three entry points of the roundabout, but doing so will inevitably cause delay to vehicles that wish to enter the roundabout from the direction controlled by this set of traffic lights, and vice versa. Therefore, all valid traffic light plans at this location would require collaboration between the four sets of traffic lights. In order to create a smooth traffic flow, collaboration between these four sets of traffic lights is crucial. Different collaboration strategy would produce different traffic flow performances at this roundabout.

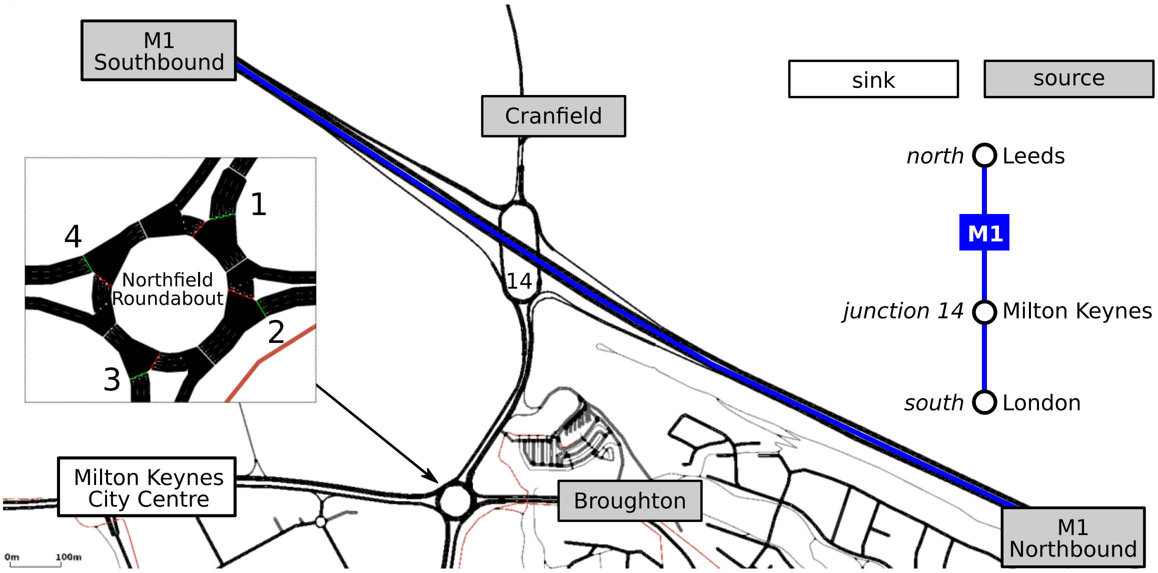


Fig. 4 Road network layout at Junction 14 of the M1 motorway including the Northfield roundabout (zoomed in) leading to Milton Keynes city centre.

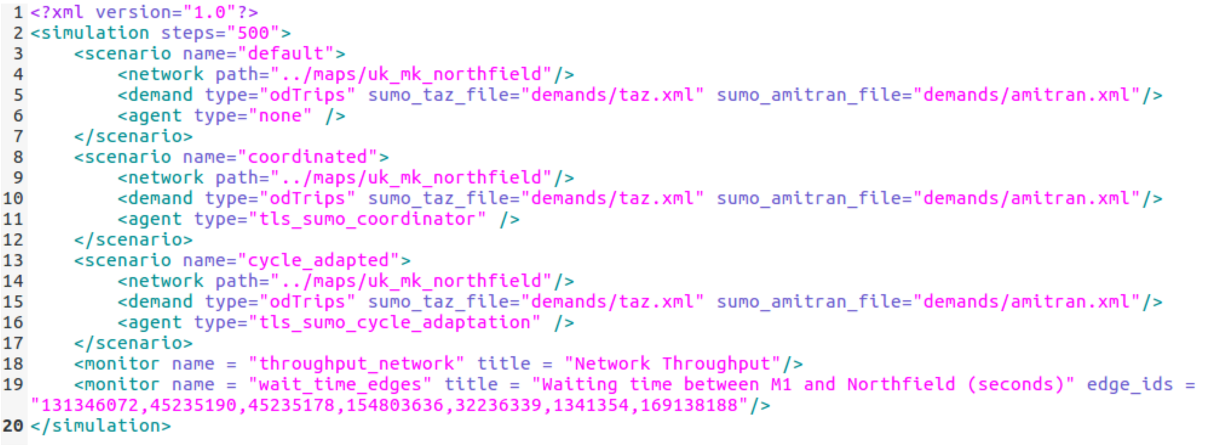
To produce a scenario similar to that of the morning rush hours going into Milton Keynes's city centre via this road network, we define the traffic flow between Milton Keynes Central and other four traffic zones as shown in Table 1.

Table 1. Traffic demand definition. (Number of Vehicles in 2000 seconds, uniform distribution)

	M1 Southbound	M1 Northbound	Cranfield	Broughton
To Milton Keynes Central	500	500	50	50
From Milton Keynes Central	200	200	20	20

4.2. Configuration

To start this experiment on the TCTB platform, we prepare a configuration file (in XML format) as shown in Fig. 5. We first specify the simulation to run for 500 seconds at line 2. Next, we specify that three simulation scenarios are to be run in parallel. Details of each scenario are given in line 3-7, line 8-12 and line 13-17 respectively. All three scenarios share the same network and demand definition (as described in 4.1). Whilst leaving the first scenario running default settings without any additional agency intervention (line 6), we apply two traffic control systems tls_sumo_coordinator by Taraz and Erdmann (2015), and tls_sumo_cycle_adaptation by Floetteroed (2017) to the other two scenarios (line 11 and line 16). At the end of the configuration file, at line 18 and 19, we specify the attachment of two performance monitors on overall network throughput and vehicle waiting time at the road from Junction 14 to the Northfield roundabout respectively for evaluation purposes.



```
 1 <?xml version="1.0"?>
 2 <simulation steps="500">
 3     <scenario name="default">
 4         <network path="../maps/uk_mk_northfield"/>
 5         <demand type="odTrips" sumo_taz_file="demands/taz.xml" sumo_amitran_file="demands/amitran.xml"/>
 6         <agent type="none" />
 7     </scenario>
 8     <scenario name="coordinated">
 9         <network path="../maps/uk_mk_northfield"/>
10         <demand type="odTrips" sumo_taz_file="demands/taz.xml" sumo_amitran_file="demands/amitran.xml"/>
11         <agent type="tls_sumo_coordinator" />
12     </scenario>
13     <scenario name="cycle_adapted">
14         <network path="../maps/uk_mk_northfield"/>
15         <demand type="odTrips" sumo_taz_file="demands/taz.xml" sumo_amitran_file="demands/amitran.xml"/>
16         <agent type="tls_sumo_cycle_adaptation" />
17     </scenario>
18     <monitor name = "throughput_network" title = "Network Throughput"/>
19     <monitor name = "wait_time_edges" title = "Waiting time between M1 and Northfield (seconds)" edge_ids =
   "131346072,45235190,45235178,154803636,32236339,1341354,169138188"/>
20 </simulation>
```

Fig. 5 Configuration file of the Northfield experiment.

4.3. Results

We first look at the overall performances of the three traffic control systems. The network throughput (vehicles arrived at their destinations) and accumulated vehicle waiting time (within road section from Junction 14 to the Northfield roundabout) of the three simulation scenarios are as shown in Table 2. Both control systems improved the performance of the road system significantly as compared to default settings. tls_sumo_coordinator gives the best overall performance almost doubling the network throughput.

Table 2. Overall performances of three traffic control systems

	Default	tls_sumo_coordinator	tls_sumo_cycle_adaptation
Network Throughput (number of vehicles)	112	**220**	168
Accumulated Waiting Time (seconds)	435404	**24014**	28085

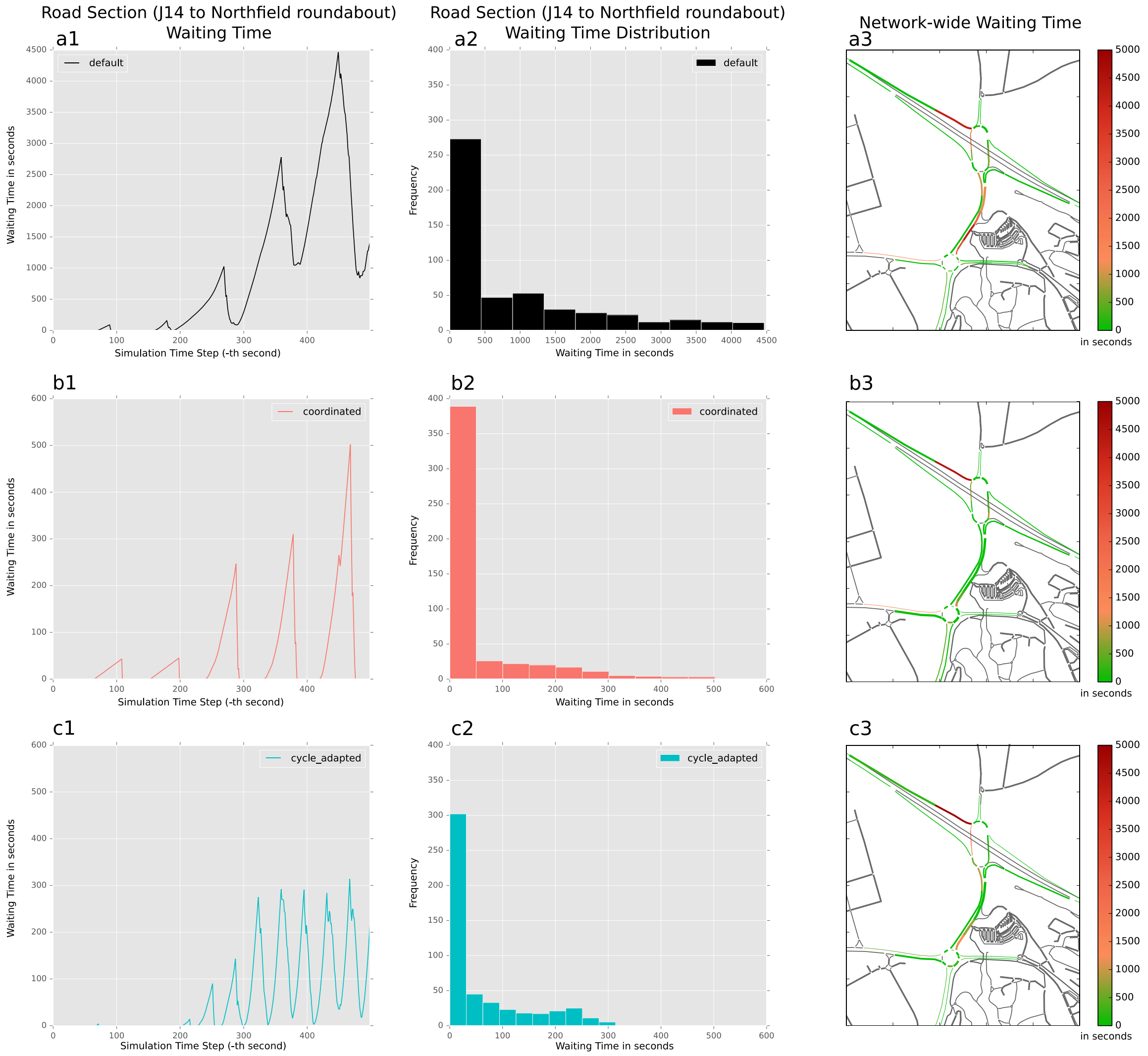


Fig. 6 Vehicle waiting time analysis. (a1-a3) Results from using default traffic light plan. (b1-b3) Results from using the tls_sumo_coordinator agent. (c1-c3) Results from using the tls_sumo_cycle_adaptation agent. Note that in the first column, (a1) does not share the y axis with (a2) and (a3); and that in the second column, (a2) does not share the x axis with (b2) and (c2).

Next, we take a more detailed look at how these performances were achieved by observing the waiting time data over the course of the simulation as shown in Fig. 6. More specifically, in the first two columns of Fig. 6, we analyse the waiting time of vehicles entering the Northfield roundabout from the M1 motorway, i.e. entry point 1 of the roundabout as shown in Fig. 4. Note that in this subsection, we omit the “Fig. 6” prefix to figure references, and all such references are from Fig. 6.

Increase in vehicle waiting time is caused when traffic is stopped by red lights, while decrease in vehicle waiting time is caused when traffic is dispersed by green lights. In the case of the vehicle waiting time along the section of road from Junction 14 to the Northfield roundabout, its value is controlled by the red/green light phases of the traffic light controlling entry point 1 of the Northfield roundabout. From the first column of Fig. 6, we see that (a1) and (b1) both have 5 peaks each corresponding to a red light phase (upward) followed by a green light phase (downward) of entry point 1 of the Northfield roundabout. This indicates that tls_sumo_coordinator, which is in charge of the traffic control system in the second simulation scenario (i.e. row b of Fig. 6), has the same number of light phases as the default setting. This indicates that tls_sumo_coordinator excels over the default setting in the traffic dispersion rate within each light phase.

This comparison is most observable at the 400th second mark along the x axis. It is very clear to see that at this point, when the traffic light at entry point 1 of the roundabout switched from green to red, the traffic build up in (b1) started from 0 whereas the traffic build up in (a1) had to start from 1000 because of residual traffic waiting at entry point 1 of the Northfield roundabout.

Comparing (b1) and (c1), we see that as well as making sure to completely disperse the waiting traffic at entry point 1 within each green phase in contrast to (a1), the number of light changes has also been doubled at entry point 1 of the roundabout by tls_sumo_cycle_adaptation. In this network, this more frequent light change produced a more stable peak value in (c1) as compared to (b1). The same can be observed by comparing the tails shown in (b2) and (c2).

In (a3), (b3) and (c3), we visualise the performance of the entire road network rather than just at entry point 1 of the Northfield roundabout. We see that the waiting times at the section of road between Junction 14 of the M1 and the Northfield roundabout have been reduced in both (b3) and (c3) in comparison to (a1). Matching our previous observations, this reduction is shown to be more significant in (b3) than in (c3). Therefore in terms of waiting time reduction at entry point 1 of the Northfield roundabout, tls_sumo_coordinator is the preferred choice over tls_sumo_cycle_adaptation.

Lastly, contrary to previous observations, expanding our comparison between (b3) and (c3) from entry point 1 of the roundabout to the entire network, we see that in (c3), the reduction in waiting time at entry point 4 of the Northfield roundabout is more significant than that produced by (b3). Without further analysis, this suggests that tls_sumo_cycle_adaptation has produced a fairer traffic control system at this location than that of tls_sumo_coordinator. However, at the same time, the quicker traffic flow via entry point 4 of the Northfield roundabout seem to also put more loads approaching Junction 14 of the M1.

5. Conclusion and Future Work

In this paper, we presented Traffic Control Test Bed (TCTB), an open source benchmarking platform for traffic control systems. The objective of this platform is to facilitate and streamline the process of evaluating new and innovative ideas of traffic control systems before they can be considered for real world implementations. TCTB achieves this objective by constructing a software platform providing agency between user-defined intelligent traffic control systems and a cohort of static assets required for the evaluation of traffic control systems which is also included in the TCTB. These assets include a library of road network models associated with traffic demands, and managed connections to the popular traffic microsimulation software SUMO.

Users of our platform are able to seamlessly apply different traffic control systems to different road network models and run simulation sessions in parallel. This not only enables the comparison between traffic control systems at the post-processing stage, but also gives the user an opportunity to visually observe the performance of each control system during runtime in parallel. In addition to post-processing output data at the end of the simulations, this platform provides the user a more intuitive and direct access to the microsimulation action.

The open source nature of this platform not only provides free and fair access to all researchers of traffic control systems in the world, but also encourages participation in further developments of the platform improving its mechanism and enriching its contents. To the best of our knowledge, this is the first work of its kind.

In future developments of this platform, we will focus on implementing both traditional and advanced traffic control systems and including them inside the platform's database of control systems for user reference and performance comparison. The accessibility and openness of control algorithms included in this platform will provide a source of inspiration and encourages future development in this field.

The end goal of this platform is to find and develop the best traffic control systems under given circumstances. We aim to build a community of researchers and developers around this open source platform and encourages participation in not only the usage of this platform, but also in its development.

Acknowledgements

This work is sponsored by the University Partnership Programme of the Transport Systems Catapult. The authors would like to particularly thank the SUMO community for the development and maintenance of the SUMO microsimulation software package.

References

Box, S. (2014). Supervised learning from human performance at the computationally hard problem of optimal traffic signal control on a network of junctions. Royal Society Open Science, 1(4). https://doi.org/10.1098/rsos.140211

Box, S. and Waterson, B. (2013) 'An automated signalized junction controller that learns strategies by temporal difference reinforcement learning', Engineering Applications of Artificial Intelligence, 26(1), pp. 652–659. doi: 10.1016/j.engappai.2012.02.013.

Box, S., Snell, I., Waterson, B. and Hamilton, A. (2012) 'A methodology for traffic state estimation and signal control utilizing high wireless device penetration.', 19th ITS World Congress, pp. 1–8.

Camponogara, E. and Kraus, W. (2003) 'Distributed Learning Agents in Urban Traffic Control', in Progress in Artificial Intelligence, volume 2902 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 324–335. doi: 10.1007/978-3-540-24580-3_38.

Chen, L. and Englund, C. (2016) 'Cooperative Intersection Management: A Survey', IEEE Transactions on Intelligent Transportation Systems, 17(2), pp. 570–586. doi: 10.1109/TITS.2015.2471812.

Dresner, K. and Stone, P. (2008) 'A Multiagent Approach to Autonomous Intersection Management', Journal of Artificial Intelligence Research, 31, pp. 591–656. doi: 10.1613/jair.2502.

El-Tantawy, S., Abdulhai, B. and Abdelgawad, H. (2013) 'Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto', IEEE Transactions on Intelligent Transportation Systems, 14(3), pp. 1140–1150. doi: 10.1109/TITS.2013.2255286.

Floetteroed, Y. (2017), tlsCycleAdaptation.py, SUMO traffic light system tools, http://sumo.dlr.de/userdoc/Tools/tls.html

Fok, C.-L., Hanna, M., Gee, S., Au, T.-C., Stone, P., Julien, C. and Vishwanath, S. (2012) 'A Platform for Evaluating Autonomous Intersection Management Policies', in 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems. IEEE, pp. 87–96.

Houli, D., Zhiheng, L. and Yi, Z. (2010) 'Multiobjective Reinforcement Learning for Traffic Signal Control Using Vehicular Ad Hoc Network', EURASIP Journal on Advances in Signal Processing. Springer International Publishing, 2010(1), p. 724035. doi: 10.1155/2010/724035.

Krajzewicz, D., Erdmann, J. and Behrisch, M. (2010) randomTrips.py, SUMO Userdoc, http://sumo.dlr.de/userdoc/Tools/Trip.html

Krajzewicz, D., Erdmann, J., Behrisch, M. and Bieker, L. (2012) 'Recent Development and Applications of SUMO - Simulation of Urban MObility', International Journal On Advances in Systems and Measurements, 5(3), pp. 128–138. doi: 10.1016/j.trc.2016.04.002.

Papadimitriou, C. H., and Tsitsiklis, J. N. (1999). The Complexity of Optimal Queuing Network Control. Mathematics of Operations Research, 24(2), 293–305. https://doi.org/10.1287/moor.24.2.293

Rafter, C., Anvari, B. and Box, S. (2017) 'Traffic responsive intersection control algorithm using GPS data', In Proceedings of the 20th International IEEE Conference on Intelligent Transportation Systems (ITSC)

Rondinone, M., Maneros, J., Krajzewicz, D., Bauza, R., Cataldi, P., Hrizi, F., Gozalvez, J., Kumar, V., Röckl, M., Lin, L., Lazaro, O., Leguay, J., Härri, J., Vaz, S., Lopez, Y., Sepulcre, M., Wetterwald, M., Blokpoel, R. and Cartolano, F. (2013) 'iTETRIS: A modular simulation platform for the large scale evaluation of cooperative ITS applications', Simulation Modelling Practice and Theory, 34, pp. 99–125. doi: 10.1016/j.simpat.2013.01.007.

Taraz, M., Erdmann, J. (2015), tlsCoordinator.py, SUMO traffic light system tools, http://sumo.dlr.de/userdoc/Tools/tls.html

University of Waterloo (2013), 'TSP Test Data', The Travelling Salesman Problem, http://www.math.uwaterloo.ca/tsp/data/index.html

University of Southern California (since 1977), 'USC-SIPI Image Database', http://sipi.usc.edu/database/

Vlahogianni, E. I., Karlaftis, M. G. and Golias, J. C. (2014) 'Short-term traffic forecasting: Where we are and where we're going', Transportation Research Part C: Emerging Technologies, 43, pp. 3–19. doi: 10.1016/j.trc.2014.01.005.

Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S. and Hubaux, J.-P. (2008) 'TraCI: An Interface for Coupling Road Traffic and Network Simulators', in Proceedings of the 11th Communications and Networking Simulation Symposium, CNS'08, pp. 155--163.

Wellman, M., Wurman, P., O'Malley, K., Bangera, R., Lin, S., Reeves, D. and E.Walsh, W. (2001) 'Designing the market game for a trading agent competition', IEEE Internet Computing, 5(2), pp. 43–51. doi: 10.1109/4236.914647