

Real-Time Traffic Controlling System Using Multi-Agent Technology

Randi Thakshila Wakkumbura, Budditha Hettige*, Ananda Edirisuriya

Department of Computer Engineering, Faculty of Computing, General Sir John Kotelawala Defence University, Rathmalana 10390, Sri Lanka

Corresponding Author Email: budditha@kdu.ac.lk



<https://doi.org/10.18280/jesa.540413>

ABSTRACT

Received: 8 June 2021

Accepted: 20 July 2021

Keywords:

traffic congestion, multi-agent systems, dynamic environment, agents, junction

With the development, almost all the sectors, countries tend to grow adapting to latest technologies. The transport sector also has a huge impact on this development sphere. When it comes to traffic, it is a huge problem in the world. In Sri Lanka traffic is a problem that exists for a long period. Annually there is a loss of Rs. 400 billion due to traffic congestions. Over the past years various solutions have been proposed using different methods for traffic control. These solutions are based on different trending technologies such as machine learning, image processing, fuzzy logic and the Internet of Things. However, those systems are not able to handle traffic congestions problems in complex environments. Therefore, a real-time traffic controlling system with the ability to handle traffic congestions in dynamic environment is highly valued. This study aims to design and build a real-time traffic management system using Multi Agent technology. Simulations for both existing system and Agent-Based system are implemented using NetLogo simulation tool and compared for various traffic situations. According to the results obtained from the two simulations, Agent-Based systems provide more accuracy and efficiency than the existing fixed scheduling systems.

1. INTRODUCTION

Road transport is one sector that evolves with numerous complex variables to handle, such as vehicle density, climate change, infrastructure, etc. With these problems, traffic congestion is a rising transport problem. Most employees who are traveling to their workplaces are on the road about 8.00 a.m., according to the Urban Mobility Study 2019 [1]. Traffic jams cause disruptions for those who use public and private transports during these hours. Colombo is Sri Lanka's economic hub and largest city. On daily basis, 7.8 million passengers travel into and out of Colombo city [2]. Therefore, in areas of Colombo, the average travel speed decreases to 10 km per hour. By 2035, the number of passengers entering the Colombo area is expected to rise by 4.4 million [3]. Sri Lanka has 7.4 million vehicles, which contributes to 70% of traffic congestion on the roads. As a result of the heavy traffic jam, Sri Lanka lost 1.5 percent of its GDP. The majority of these traffic jams occur at intersections where traffic is controlled by traffic light systems.

Traffic congestion is not only a problem for Sri Lanka but also to entire globe [4]. Increased traffic congestion is an inevitable condition in large and growing urban areas in each continent. Peak hours road congestion is unavoidable in the way industrial cities operate. Everyone hates traffic congestion, and despite attempted solutions, it keeps getting worse. According to the past world statistics, the cities with the highest traffic congestion are in South East Asian countries, where India is leading [5]. The main reason for this traffic congestion in India is the complete inadequacy of alternatives for the public to get around.

In Sri Lanka, as in most other nations, the current traffic

control system is pre-programmed and timed. The cycles are split into two groups: peak hours and off-peak hours. Mornings, evenings, and nights are usually considered peak hours, while early mornings and midnight are considered off peak hours. Despite the fact that it is listed as such, the process is still pre-programmed and cannot be modified to accommodate the situation. This causes significant issues, particularly at intersections, because the system cannot adapt to the real traffic level, resulting in severe traffic jams. When two junctions are close by, a lot of time and fuel are lost. Police officers are also deployed to intersections in addition to the current traffic control systems. These traffic jams not only waste time but also trigger tension, injuries and the urge to violate the traffic rules, among other factors. Figure 1 shows the vehicle growth for the past 8 years and passengers travel daily into Colombo city.

The need for real time traffic controlling for a country is essential due to this increasing traffic jams. Most countries used the preprogrammed traffic controlling systems for junctions as well as for other roads, while some countries used CCTV monitoring systems separately to monitor the road system. Since various traffic situations can occur, a preprogrammed system is not efficient and productive for current traffic congestion. With the advancement of technology, we can now move into better real time traffic controlling systems that provide several advantages. When moving into such systems the technology selection is a more important factor. Among them, Multi Agent system technology can be defined as an intelligent agent technology that can implement real time solutions to handle this type of complex problem [6]. A complex system is difficult to analyze and understand, thereby difficult to monitor. With the above

concerns in traffic controlling systems in Sri Lanka, the research presents the development of an intelligent traffic control system through Multi Agent system technology. The system was developed with the use of NetLogo simulation software as two simulations for fixed-scheduling system and Agent-Based system. Then the results are being evaluated in terms of efficiency, accuracy and performance by comparing two systems.

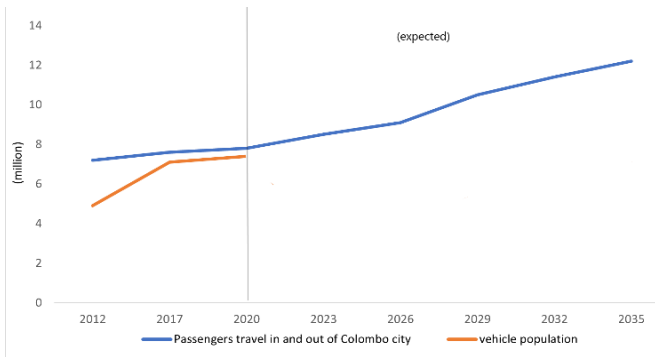


Figure 1. Vehicle growth and daily passenger travel to Colombo city

The rest of the paper is organized as follows. The second section provides an overview of works related to Multi Agent technology-based frameworks, simulations and existing traffic control systems. The third section describes the methodology of the system developed with design diagrams and simulation interfaces. The fourth section presents the results obtained from two simulation setups and briefly discussed the performance of the system including. The last section concludes by highlighting the entire system efficiency and further works of the project.

2. RELATED WORK

Many researchers have applied the Multi Agent technology to address complex problems. The concept of complexity can be described with seven features how it can be measured and tuned namely autonomy, emergency, nonequilibrium, non-linearity, self-organization, co evolution and connectivity [7]. It is important to understand the concept when addressing complex problems. There are several agent programming languages, frameworks, and simulations developed for the Multi Agent technology. The agent programming languages can be categorized into two sections as declarative and imperative. Declarative languages are based on logic while imperative languages have uncommon language literature. Most of the developments have been involved declarative language since imperative languages are not suitable for high level designs. FLUX, Minerva, Dali, CLAIM, MDLPC are some of the common declarative languages [8]. CLAIM is a language which support to design mobile Multi Agent system through its organized agent hierarchy. FLUX is a specially designed programming language for cognitive agents.

Though imperative language usage is not high as declarative there are some languages implemented based on this concept, such as JACK, 3APL and Jason [9]. Agent-oriented software company developed the JACK programming language as an extension of Java. It is comprised of properties like the ability to use agent organizations, agent teams and valuable supporting tools. Various industrial applications have been

used JACK for their systems. 3APL is also known as abstract agent programming language which is designed for cognitive agents that have special characteristics. Some software engineering and programming principles like reusability, abstraction, modularity have been followed by the 3APL language. The special application of 3APL is the SONY AIBO robot behaviour implementation.

The difference in agent languages and agent frameworks is most of the languages have some underneath support platform while frameworks are more supportive in agent communication. JADE, TuCSoN, Jadex, MaSMT and DESIRE are some of Multi Agent frameworks developed with a variety of features [8]. TuCSoN can be described as Multi Agent system coordination framework. A run time platform, API's and supporting tools are available in this Java based TuCSoN technology. JADE is one of the most popular Java based framework that can use to develop the distributed Multi Agent applications. JADE is easy for debugging and testing because of the available services and tools. It consists of a good structure and follows FIPA specifications. Thus, it achieves the framework's main objective, interoperability. MaSMT is another framework developed for Agent-Based programming that saves a lot of programming time [10]. Various applications have been developed with this framework including English to Sinhala Machine Translation, Octopus and Ricemart. In Multi Agent solution for English to Sinhala Machine Translation, the system was implemented with six agent swarms on the MaSMT framework [11]. This complex system handles more than 500 agents. The behaviour of the agents has exposed them to language specific rules.

Multi Agent based simulation development software provides more benefits than traditional software. NetLogo, SeSAM, MASS, StarLogo, GAMM, FLAME are some of the existing simulation software [12]. NetLogo is the most widely used software developed at Northwestern University. Users can provide a set of rules to follow for the model population and NetLogo software can simulate the possible scenarios for a given situation. Advanced programmers can build their model with custom Java extensions. They can develop the prototype, debug and algorithm visualization through NetLogo. For beginners also it is easy to install, use, extend and publish. NetLogo provides great supporting materials and a web environment for testing and rapid publishments [13].

Various studies have been conducted to find solutions for this traffic congestion problem. Arel and others implemented a traffic light control system with the combination of reinforcement learning and Multi Agent technology [14]. The system methodology involves a Feedforward Neural Network and Q learning algorithm. The simulation is developed through MATLAB software for a five-way junction and measured the performance. The system showed improved results in average vehicle delay and some heavy traffic jam situations. Further, Jamal and others implemented an Intelligent system that optimizes the delays in road intersections with a search algorithm [15]. The system is proposed for a specific city in Arabia. The methodology involves the use of a Genetic Algorithm and differential evolution. The algorithms follow a series of procedures to optimize the delays as encoding the situation, population generation, evaluation, best solution update and cross-over. The methodology is validated via a simulation tool called TRNASYT7F. The result of the model showed a significant diminish in delays compared to the existing system circumstances. An Intelligent traffic information system was developed based on a combination of

the Internet of Things (IoT) and Agent technology [16]. The system collects traffic information and provides a solution based on IoT and Agent-Based technologies. The system aims to achieve high compatibility, scalability, and low cost features. RFID technology had been used to collect data and the data analysis was done through the server. Each vehicle has a separate RFID tag to collect data. The system mainly consists of three layers as application, network and acquisition. However, the large number of sensors and data includes in the system cause to have high security measures for the system.

A Smart automated system was developed to manage the traffic using Image Processing technology [17]. The system was implemented to control overtaking and law-breaking as it affects manage the road traffic. The Image Processing techniques cascade classifier and the grayscale algorithm are used to detect a vehicle in live streaming video. Apart from that the system counts the number of vehicles passing some specially assigned areas. A large-scale traffic signal control system was implemented with deep reinforcement learning technology [18]. Theories like pressure-based coordination and other transportation theories were applied for the model. The base model is adapted from FRAP architecture and a deep Q learning algorithm. The model experimented for various scenarios especially for real world environments with more than 2500 traffic lights in New York city. A traffic management scheme was present for Kottawa city to handle the present traffic situation and predict the future possibilities [19]. The process consists of data collection, capacity analyses and data analyses related to different traffic and road conditions. The traffic scheme is analyzed by dividing the junction into several subdivisions as two-way two-lane rural road, roundabout, sidewalk, crosswalk and parking areas. With the analyzed results the system proposed areas for parking areas for vehicles and buses separately. Further, a layout plan was designed to achieve minimum traffic congestions. Road traffic and congestion management system developed based on the vehicular Ad Hoc Networks techniques [20]. The color light system is implemented with Ad hoc network platform and traffic environment simulated through Veins framework. Omnet++ software that is deployed on top of the platform is used to control the traffic and the results depict vehicle communication cause to traffic reduction. To protect the law and driver behaviours a Machine Learning lane detection feature is implemented in the system. Traffic congestion management in junction involves vehicle prioritization which handles through vehicle type and first in first out methods.

Another system was implemented for traffic monitoring based on priority using the Internet of Things and sensor technology [21]. The hardware system consists of UV sensors, ARM 7 controller, RF Transceiver, LED unit, PIC microcontroller and Wi-Fi modules. Vehicle traffic levels are detected by the UV sensors and transmit the data to the server via a Wi-Fi module. In the server, the data is analyzed according to the traffic levels. The higher priority is given to the lane with high traffic levels. The use of RF transceiver is mainly for communication. The real time traffic monitoring addresses problems such as traveling time, air pollution and fuel wastage. Walad and Shetty proposed the potential to manage traffic lights via Image Processing [22]. Hardware, software, and connecting components make up the system. The presence of vehicles was declared using edge detection, and roads with high vehicle density were given priority. Time was also used as a component in the operation process. Four edge detection operators, Robert, Sobel, Prewitt and Canny

have been analyzed and compared to perform the traffic density function. The systems prioritize avoiding time wasted in the existing fixed scheduling system; however, the correctness of both systems is not fully guaranteed. Fuzzy logic based smart traffic light system is another idea proposed to prevent traffic congestion [23]. The system consists of a simulator to optimize the traffic using a Fuzzy Mamdani Logic model that functions based on several criteria such as queue length, vehicle count and road width. The results were evaluated with the fuzzifications, vehicle count and road width. The parameters used to optimize the green length of traffic lights and the traffic congestion reduce with green light application. The system showed efficient results, especially for intersections. Traffic density estimation and traffic control system are proposed based on a Convolutional Neural Network to address the traffic problem in Sri Lanka [24]. The system was implemented to convert static traffic signal system into dynamic. The traffic density is measured through Machine Learning algorithms. The system is implemented especially for a four-way junction and to reduce the traffic congestion system used three methods as signal handling, estimating crowd density and density estimation algorithm. The traffic light simulation is implemented with a Q learning algorithm to obtain the optimal result. Apart from the Convolution Neural Network model system used the Recurrent Neural Network as an alternative model. The system was simulated through SUMO software and density estimation through YOLOV3 software. The system is trained for two classifiers and evaluated the results. Overall, many traffic management systems were proposed or built using numerous technologies as shown in Table 1.

Table 1. Summary of related works

References	Technology	System type/ Features
[14]	Reinforcement Learning, Agent Technology	MATLAB Simulation for junction, Algorithm based
[15]	Meta-Heuristic Search Algorithms	Simulation with actual data in Dhahran city, traffic guidance system.
[16, 21]	Internet of Things, RFID, Sensor network	Simulation and hardware system, Controller-based
[17, 22]	Image Processing	Real time, for all road network types
[18]	Machine Learning	Model for real world scenario in New York city
[19]	Statistical Analysis	For Kottawa city, Sri Lanka, give predictions
[20]	Vehicular Ad Hoc Networks	Simulation, for junctions in Sri Lanka, real time
[23]	Fuzzy Logic,	Software system, real time and for intersections
[24]	Convolutional Neural Network	Simulation and app, for four-way junctions in Sri Lanka, traffic minimizing approach

These systems are built using machine simulation or modeling solutions. To deal with system complexity, many systems employ traditional techniques such as constructed algorithms, mathematical functions and models. These schemes fix traffic congestion in urban areas, with a small percentage of them focused on traffic at four-way intersections. To handle the Sri Lankan traffic conditions, there is a practical need for effective, simple yet powerful traffic control system.

3. METHODOLOGY

Using the Multi Agent system technology, research has been undertaken to create an Intelligent traffic control system [25]. The input for both the software simulation and the hardware framework would be the number of vehicles in each lane. The car count in machine simulation can be manually adjusted and input. The Multi Agent model will access the database and take decisions according to the data.

The Multi Agent technology ontology provides the metadata and the relations associated with the agents [26]. The Figure 2 represents the ontology design for the system. Ontological information may be used to construct the system's knowledge base. The Junction agent has the Vehicle and Pedestrian agents. The Controller agent controls the Vehicle, Junction and Pedestrian agent according to the requests.

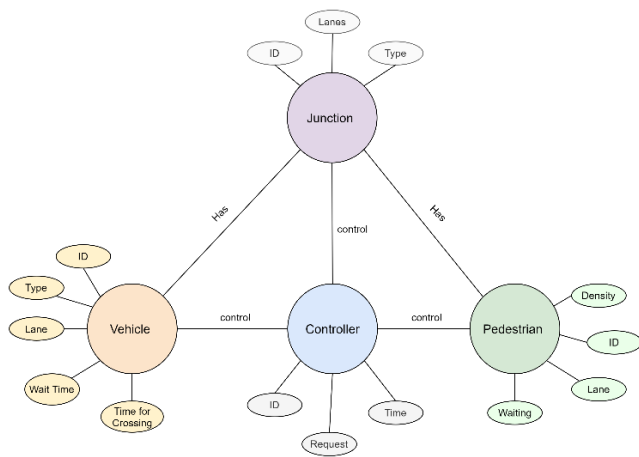


Figure 2. Ontology of the MAS

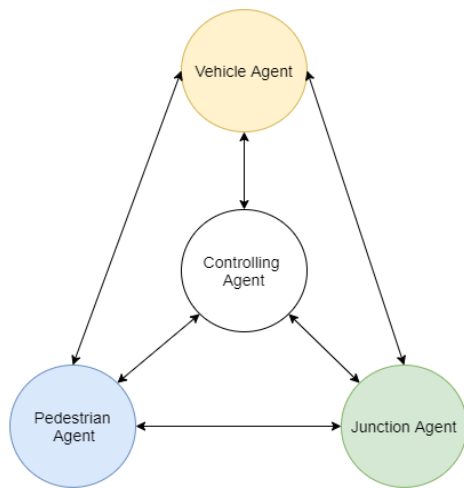


Figure 3. Experimental design of the MAS

This traffic management scheme is made up of four distinct types of agents. They are Vehicle, Intersection, Pedestrian, and Controlling agents. With the support of both car and pedestrian officers, the Junction agent operates the traffic lights in each direction of the intersection. Junction agent specifically aims to link vehicles and pedestrians by offering options for both without creating confrontation. The Vehicle agent will consult the other agents at the intersection to assess the situation. Priorities will be assigned to Vehicle agents based on different traffic conditions. Vehicle agents must also give Pedestrian

agents a chance. The Controller agent is in charge of making those decisions. The Pedestrian agent may speak with the Junction agent, but he or she will have to wait for permission to cross before other agents have dealt with the situation. The Controlling agent is in charge of all Vehicle, Pedestrian and Junction agents. Figure 3 shows the agent design of the system.

The simulations are developed for two systems the existing fixed scheduling system and the proposed Agent-Based system. The existing fixed scheduling system depicts the traditional preprogrammed traffic controlling in a four-way junction, while the Agent-Based system depicts the Intelligent based solution for traffic control in terms of vehicle and pedestrian control. The machine in Multi Agent simulation makes decisions based on previously established principles. The lanes that do not have any vehicles will not become operational until one appears.

The agent communication scheme diagram is shown in Figure 4. There are three rules in the system, lanes without any vehicle will be in an inactive state, lanes with high vehicle density will gain more turns and a maximum waiting time assigned for each lane in the junction.

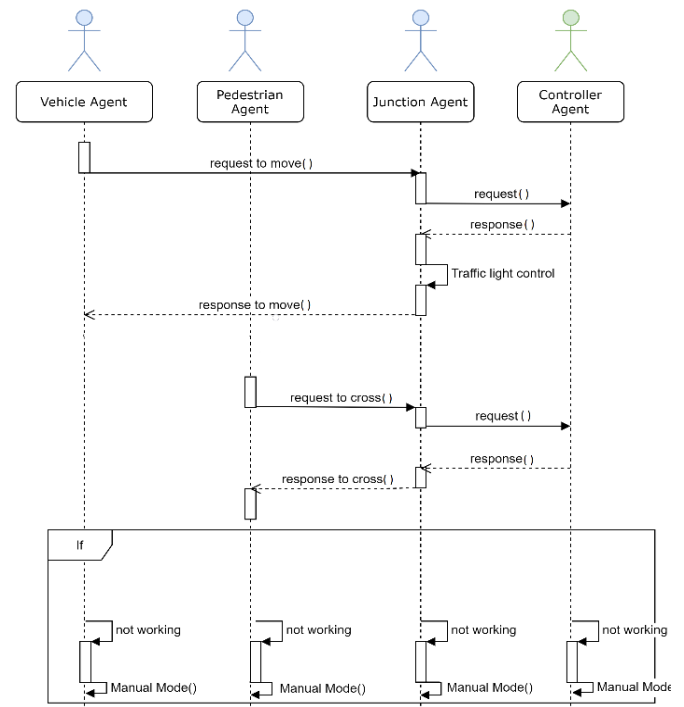


Figure 4. Communication diagram

All the requests and response messages receiving from the Vehicle agent, Junction agent and Pedestrians agent is controlled by the Controller agent. A request from the Vehicle agent will communicate to the Controller agent via the Junction agent. A request from the Pedestrian agent will pass to the controlling agent via the Junction agent. The Controller agent will analyze all the requests coming from the agents and respond according to the system model. In case the agents are not operating the agents will communicate and switch to the manual mode which is the existing fixed scheduling system. This is a safety precaution of the system since the whole system will affect by any agent collapse.

With the defined ontology and experimental design, the working simulation is developed through NetLogo software. In the simulation environment, four-way junction is implemented with three lanes for each road. And each road

contains a pedestrian crossing (white color). The Traffic lights for roads and pedestrians are located in red and green circles. Pedestrians' walk path is initialized in brown color. Figure 5 represents the designed environment and agent placement in simulation. The first lane is for turn left. The second lane is to go forward and the third lane is to turn right.

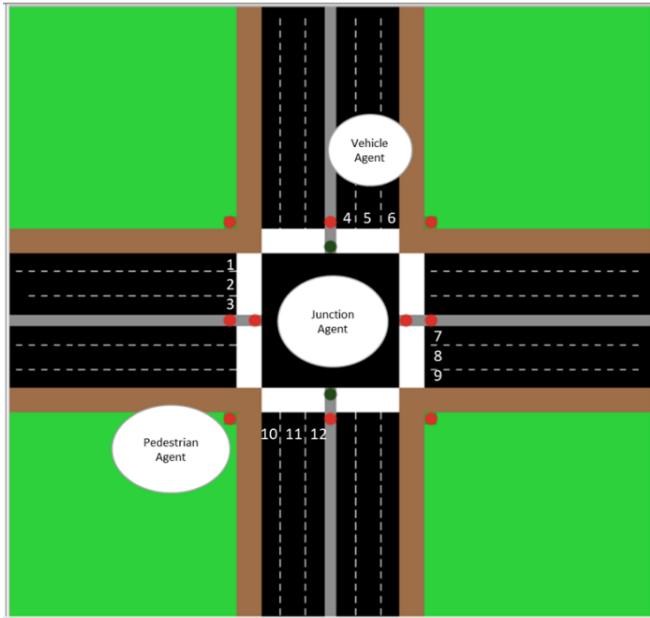


Figure 5. Fixed scheduling simulation

Cars or vehicles are one of the main agents in the system. In the simulation, each car is initialized by its speed and given a top speed that cannot be exceeded. The car count in each lane can be differentiated manually for various simulation situations. The traffic lights belong to the next main agent type Junction. The two simulations differ as normal and Agent-Based by this function. From the pair of lights in the middle of each road, front lights are for turn left and move forward. Rear lights for turn right. The four lights in four corners are for the pedestrian crossings.

The simulations are created for the existing system and the proposed Agent-Based system. Both Simulations consist of several features to simulate different scenarios. The first feature is vehicle density control which allows users to manually change the number of vehicles in each road at once or change the vehicle density lane by lane in 12 lanes in the junction. Here the roads are labeled as north, east, south, and west as per directions. In the second feature, the vehicle control user can change the parameters of a vehicle. The parameters are acceleration, deceleration, maximum speed, and maximum patience. Here maximum patience indicates obeying the traffic light. If patience is low the vehicle collisions can be increased. Here that feature is initiated because in real time environment not all the drivers are perfect. Therefore, the simulation should be subjected to those scenarios. Then the pedestrian control, the user can change the total pedestrian density arrive for the junction and the turning frequency. Turning frequency initiates how many times, a pedestrian can cross by turning to the given directions. Finally, the Graphical visualization of results. This is one of the

important features in both simulations. Results of the simulations are represented in terms of graphs so that it is easy to compare both simulations for discussion. In this graphical representation, three parameters have been graphed such as the total number of vehicles crossed the junction, Average waiting time and Vehicle density bear by each road in the junction, both in and out vehicle count.

Figure 6 shows the simulation environment for the fixed scheduling system and Figure 7 shows the simulation for Agent-Based system. The existing system depicts the traditional preprogrammed traffic controlling in a four-way junction, while the Agent-Based system depicts the Intelligent based solution for traffic control in terms of vehicle and pedestrian control. In a Multi Agent simulation, the system makes decisions based on previously established principles.



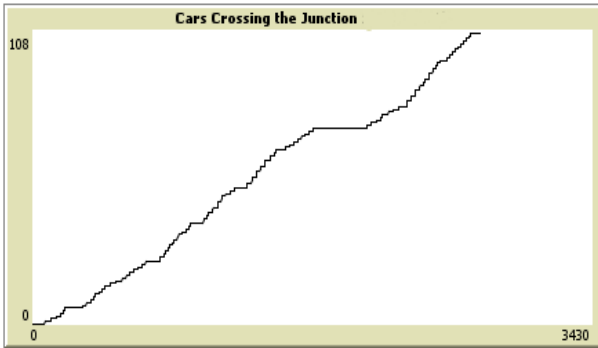
Figure 6. Fixed scheduling simulation



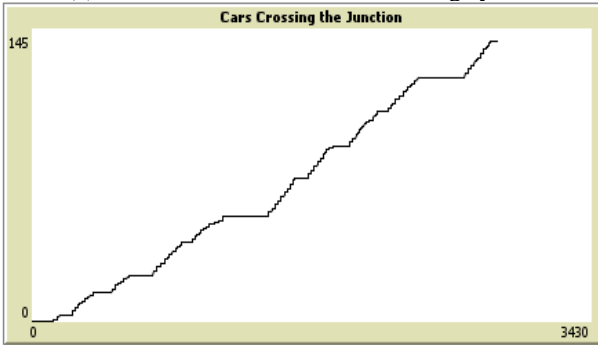
Figure 7. Agent-based simulation

4. RESULTS AND DISCUSSION

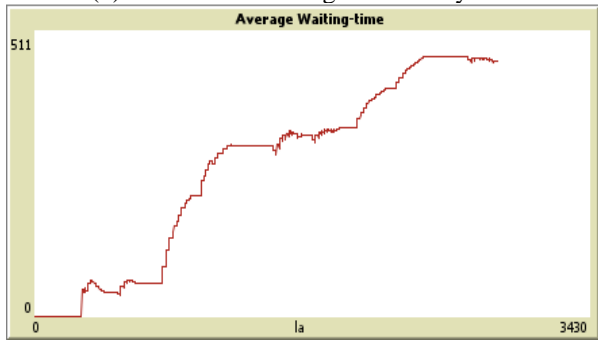
Several traffic status results are presented here separately in terms of the number of vehicles crossed the junction and average waiting time by comparing time-based simulation and Agent-Based simulation. The simulations graph values represent by ticks. All the simulations run to equal tick time which is 3430 and compared the results. In addition to the above parameters change of vehicle density in each road is graphed in both simulations. Figure 8 and Figure 9 present the results of two scenarios obtained from these simulations.



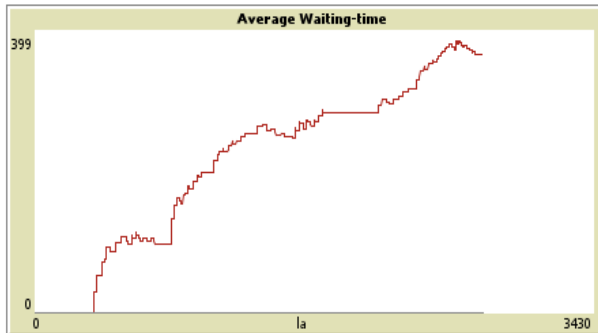
(a) Vehicle count in fixed scheduling system



(b) Vehicle count in agent-based system

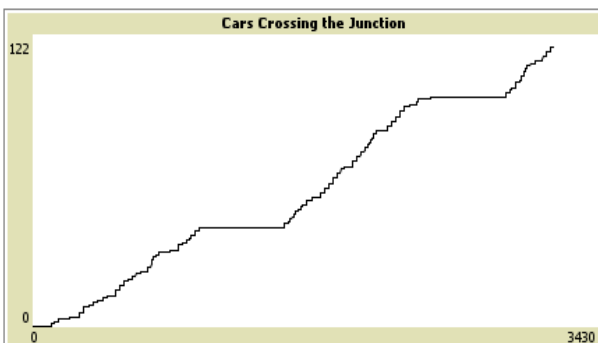


(c) Average waiting time in fixed scheduling system

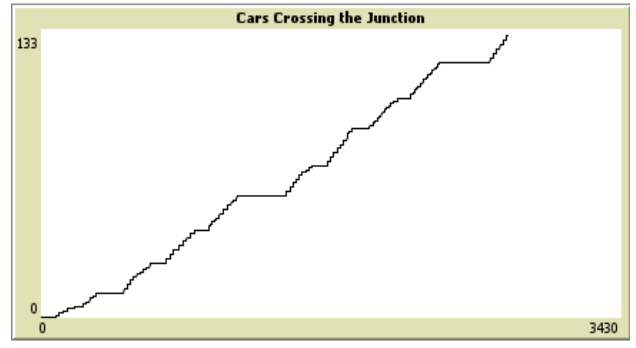


(d) Average waiting time in an agent-based system

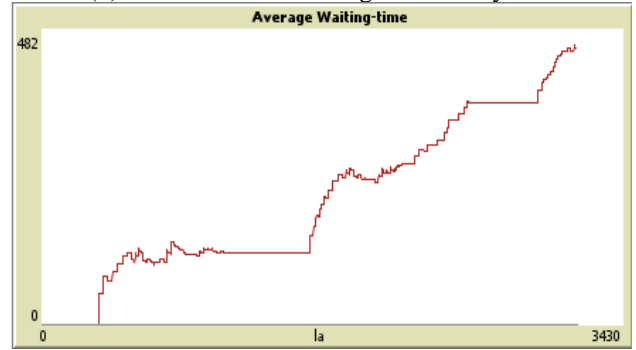
Figure 8. Scenario 1- Less vehicles in one road



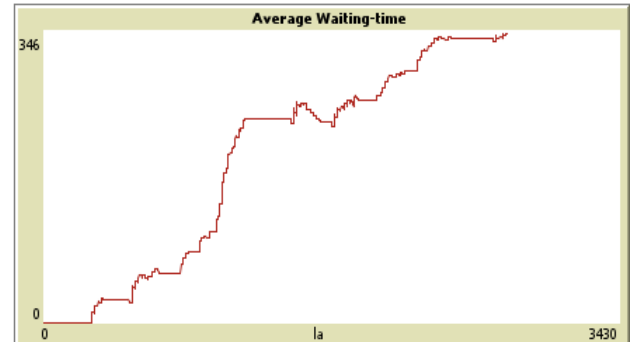
(a) Vehicle count in fixed scheduling system



(b) Vehicle count in the agent-based system



(c) Average waiting time in the fixed scheduling system



(d) Average waiting time in an agent-based system

Figure 9. Scenario 2 - More vehicles on one road

Scenario 1: In a situation, where one road with fewer vehicles, (a), (b), (c) and (d) of Figure 8 shows the two simulation results. When comparing graphs (a) and (b) in Figure 8 it is clear more vehicles passed in the Agent-Based system. Though values gradually increased in both graphs the Agent-Based system performance is higher than the fixed scheduling system. As for the waiting time when comparing graphs (c) and (d) the value difference shows the Agent-Based system vehicles cross the junction with less vehicle time. This kind of situation can be occurred due to accidents on one road. The Agent-Based traffic light controlling system brings more benefits in a situation, where one road occupies less vehicle density rather than in normal condition.

Scenario 2: In Figure 9, (a), (b), (c) and (d) shows the current system simulation and Agent-Based system simulation results for the situation where one road with more vehicles. While the Agent-Based system shows a rapid increase in vehicle count as in graph (a) the scheduling system vehicle count increment is slow as depicted in graph (b) in Figure 9. Though the traffic situation is different from scenario 1, this scenario also shows efficient performance in an Agent-Based system when comparing the waiting time of vehicles as depicted in (c) and (d) graphs of Figure 8. This kind of situation can be occurred due to road connection with highway

road etc. Agent-Based systems in four-way junctions take less waiting time and pass more vehicles across the junction. Here, also we can see the Agent-Based traffic light system is more efficient in a situation, where one road occupies a high vehicle density rather than in normal condition.

Scenario 3: This scenario depicts the situation where traffic with fewer pedestrians on the road. With the obtained NetLogo simulation values, the average results have been presented. Mainly the waiting time of vehicles reduces in both systems when there are few pedestrians. But Agent-Based system presents more performance than a scheduling system. Less waiting time means more vehicle crossings as shown in results 110 and 141. Since there are 4 pedestrian crossings in a four-way junction, the two simulations depict a significant difference with the above efficient results for an Agent-Based system.

Scenario 4: This is a general traffic situation without any special traffic conditions. All the parameter values were adjusted to normal conditions. The NetLogo simulation results were obtained and compared for the general traffic situation without increasing or decreasing any parameters such as vehicle count and pedestrian count. The values obtained for vehicle count are 118 and 135 for the scheduling system and Agent-Based system respectively. Therefore, the result evaluation shows high performance efficiency in an Agent-Based system.

The system has tested 10 times for each traffic situation and obtained average values for average waiting time and average vehicle count crossed the junction. Each simulation has been run for 10 test cases for one traffic situation to get the average value. Apart from the above-mentioned scenarios, the system tested for traffic situations such as more vehicles in one road, no vehicles in one road, no vehicles to turn right in one lane, vehicles only in the right turn lane of one road, no Pedestrians to cross the junction and no vehicle in two roads of the junction. Below Table 2 represents the average values obtained for the above discussed situations.

Table 2. Four traffic scenario results obtained from NetLogo simulation

Traffic Situation	System	Number of vehicles crossed junction
Scenario 1	Fixed Scheduling system simulation	108
	Agent-Based system simulation	145
Scenario 2	Fixed Scheduling system simulation	122
	Agent-Based system simulation	133
Scenario 3	Fixed Scheduling system simulation	110
	Agent-Based system simulation	141
Scenario 4	Fixed Scheduling system simulation	118
	Agent-Based system simulation	135

Notes: 1. The values are obtained through NetLogo simulation. Units of the values are ticks.

Obtained results can be graphically shown as in Figure 10. We can clearly identify the difference between the two methods.

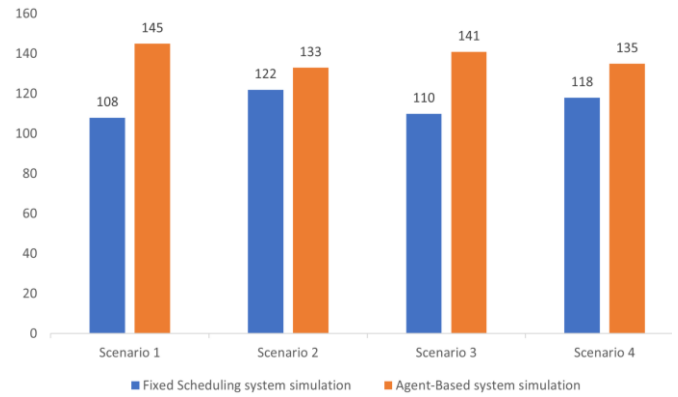


Figure 10. Results comparison

5. CONCLUSIONS

This research work intended to model real time traffic controlling through simulations. The performance of the system was tested and verified for various scenarios at the testing stage. In the simulation, it is notable that how agent technology can contribute to the system performance. The majority of traffic control systems, use Fuzzy Logic, Learning Classifier and Reinforcement Learning technologies. These systems were created with standard functions in mind, such as real-time monitoring, center controller-based algorithms, and so on. The majority of traffic control systems do not gain dynamic control. Dynamic scheduling is critical for controlling junctional traffic congestion, particularly when transitioning from a fixed time-based system to an autonomous Intelligent system. In addition, technologies such as image recognition and the Internet of Things have also been used for automatic traffic control. When it comes to traffic management, only a few methods have been used Multi Agent technologies. Those methods, on the other hand, are often limited to theories and automated programs. A traffic control system involves agents, who may operate individually and have control of their internal state. The environment is what determines how agents deal with the outside world. In the world traffic control system is complex, meaning it varies when an agent is deliberating. To allow each agent to interpret messages from other agents for traffic control, the ontology must be settled upon and accepted by the agent class.

The main benefit of using Multi Agent system technology for traffic management is that it can handle changing environmental conditions. This can track each lane in the intersection and make changes based on the situation. This meets the adaptability criterion. The system environment can be changed from static to dynamic using the agent communication feature. Choosing the right agent for device development can be difficult. According to the research, traffic management using Multi Agent device technologies in accordance with software and hardware systems may be more reliable and effective than the existing fixed scheduling system.

The study primarily focused on traffic management through agent interaction in an intersection does not cover the entire scenarios of four-way junctions in Sri Lanka like breakdown effects, sudden road plans, police officers, etc. The comparative results obtained from simulation operated in the computer can differ when testing in a large network like Sri Lankan traffic environment is also a limitation in the study. Another identified flaw is most of the time simulation

comparison provides positive feedbacks but some instances received negative outputs with the testing times. Therefore, the system fault tolerance should be improved in future developments.

In the next step, we will build a completely functional hardware module to validate the device in a real-time environment. The proposed system's actual success will be measured after it has been applied in a real world setting. Furthermore, a device for two consecutive junctions could be built to allow coordination between two agents and to manage traffic demands.

REFERENCES

- [1] Schrank, D., Eisele, B., Lomax, T. (2019). 2019 Urban Mobility Report 50.
- [2] Traffic Congestion Problems in Colombo City. <https://www.ukessays.com/essays/tourism/analysing-traffic-congestion-problems-in-colombo-city-tourism-essay.php>.
- [3] 'Jammed!|SundayObserver'. <http://www.sundayobserver.lk/2017/10/15/features/jammed>.
- [4] Ye, S. (2012). Research on urban road traffic congestion charging based on sustainable development. *Physics Procedia*, 24: 1567-1572. <https://doi.org/10.1016/j.phpro.2012.02.231>
- [5] 'Traffic congestion ranking | TomTom Traffic Index'. https://www.tomtom.com/en_gb/traffic-index/ranking/.
- [6] Xie, J., Liu, C.C. (2017). Multi-agent systems and their applications. *Journal of International Council on Electrical Engineering*, 7(1): 188-197. <https://doi.org/10.1080/22348972.2017.1348890>
- [7] Rzevski, G., Skobelev, P. (2014). *Managing Complexity*. Wit Press.
- [8] Bordini, R.H., Braubach, L., Dastani, M., Seghrouchni, A.E.F., Gomez-Sanz, J.J., Leite, J., Ricci, A. (2006). A survey of programming languages and platforms for multi-agent systems. *Informatica*, 30(1): 33-44.
- [9] Dastani, M., Meyer, J.J.C. (2007). A practical agent programming language. In *International Workshop on Programming Multi-Agent Systems*, 107-123. https://doi.org/10.1007/978-3-540-79043-3_7
- [10] Hettige, B., Karunananda, A.S., Rzevski, G. (2021). MaSMT4: The AGR organizational model-based multi-agent system development framework for machine translation. In *Inventive Computation and Information Technologies*, 691-702. https://doi.org/10.1007/978-981-33-4305-4_50
- [11] Hettige, B., Karunananda, A.S., Rzevski, G. (2016). A multi-agent solution for managing complexity in English to Sinhala machine translation. *Complex Systems: Fundamentals & Applications*, 90: 251. <https://doi.org/10.2495/DNE-V11-N2-88-96>
- [12] 'Agent-Based Models, Simulators'. <http://www.agent-based-models.com/blog/resources/simulators/>.
- [13] Sklar, E. (2007). NetLogo, a multi-agent simulation environment. *Artificial Life*, 13(3): 303-311. <https://doi.org/10.1162/artl.2007.13.3.303>
- [14] Arel, I., Liu, C., Urbanik, T., Kohls, A.G. (2010). Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intel. Transp. Syst.*, 4(2): 128. <https://doi.org/10.1049/iet-its.2009.0070>
- [15] Jamal, A., Tauhidur Rahman, M., Al-Ahmadi, H.M., Ullah, I., Zahid, M. (2020). Intelligent intersection control for delay optimization: using meta-heuristic search algorithms. *Sustainability*, 12(5): 1896. <https://doi.org/10.3390/su12051896>
- [16] Al-Sakran, H.O. (2015). Intelligent traffic information system based on integration of internet of things and agent technology. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 6(2): 37-43. <https://doi.org/10.14569/IJACSA.2015.060206>
- [17] Shamrat, F.J.M., Mahmud, I., Rahman, A.S., Majumder, A., Tasnim, Z., Nobel, N.I. (2020). A smart automated system model for vehicles detection to maintain traffic by image processing. *International Journal of Scientific & Technology Research*, 9(2): 2921-2928.
- [18] Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Li, Z. (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4): 3414-3421. <https://doi.org/10.1609/aaai.v34i04.5744>
- [19] Maduwanthi, C.D.S., Gunarathna, T.M.N., Weerasekera, K.S. (2016). A Traffic Management Scheme to ease Vehicle & Pedestrian Movements in the Kottawa Town. *Engineer: Journal of the Institution of Engineers, Sri Lanka*, 49: 35-50. <https://doi.org/10.4038/engineer.v49i2.7007>
- [20] Lakindu, Y., Jayawardena, J., Bandara, S. (2020). Road traffic and congestion management system for intersections based on VANET. *International Conference on Advances in Computing and Technology*, p. 3.
- [21] Nagmode, V.S., Rajbhoj, S.M. (2017). An IoT platform for vehicle traffic monitoring system and controlling system based on priority. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1-5. <https://doi.org/10.1109/ICCUBEA.2017.8463825>
- [22] Walad, K.P., Shetty, J. (2014). Traffic light control System using image processing. *International Journal of Innovative Research in Computer and Communication Engineering*, 2(5): 289. <https://doi.org/10.35741/issn.0258-2724.55.1.31>
- [23] Hartanti, D., Aziza, R.N., Siswipraptini, P.C. (2019). Optimization of smart traffic lights to prevent traffic congestion using fuzzy logic. *TELKOMNIKA Telecommunication Computing Electronics and Control*, 17(1): 320-327. <https://doi.org/10.12928/telkomnika.v17i1.10129>
- [24] Ikiriwatte, A.K., Perera, D.D.R., Samarakoon, S.M.M.C., Dissanayake, D.M.W.C.B., Rupasinghe, P.L. (2019). Traffic density estimation and traffic control using convolutional neural network. In *2019 International Conference on Advancements in Computing (ICAC)*, pp. 323-328. <https://doi.org/10.1109/ICAC49085.2019.9103369>
- [25] Bellifemine, F.L., Caire, G., Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE (Vol. 7)*. John Wiley & Sons.
- [26] Kim, S., Son, Y.J., Tian, Y., Chiu, Y.C., Yang, C.D. (2017). Cognition-based hierarchical EN route planning for multi-agent traffic simulation. *Expert Systems with Applications*, 85(3): 335-347. <https://doi.org/10.1016/j.eswa.2017.05.045>