

## Prioritization of Exigency Services in Multi-Agent Transportation Systems

Awais Qasim<sup>1</sup>, Saira Hussain<sup>2</sup>, Shuja ur Rehman Baig<sup>3</sup>, Muhammad Ilyas Fakhir<sup>4</sup>, Muhammad Bilal<sup>5</sup>, and Adeel Munawar<sup>6</sup>

<sup>1,2,4</sup>Department of Computer Science, GC University, Lahore, Pakistan

<sup>3</sup>Punjab University College of Information Technology, Lahore, Pakistan

<sup>5</sup>Department of Computer Science, University of Management and Technology, Lahore, Pakistan

<sup>6</sup>Sirindhorn international institute of technology, Thammasat University, Bangkok, Thailand

<sup>6</sup>Department of Computer Science, Lahore Garrison University, Lahore, Pakistan

### ABSTRACT

*Multi-agent system helps in achieving a single global goal by working on different tasks in a distributed environment. This research presents a new framework for handling the prioritization of exigency services in an urban transportation system. Prioritization in exigency services needs to be handled for vehicles like Ambulance, Police Mobile, Fire Brigade, Bomb Disposal Squads, Search and Rescue vehicles, Turntable Ladder, and other similar vehicles. In the proposed framework a single ARTIS agent with four In-agents is deployed at each signal node. In-Agents at different nodes share information about the exigency with the ARTIS Agent that analyses the input data and determines the types of conflict that might cause delays in emergency services provision. We operate the signal based on their priorities for exigency vehicles with different priorities. In case they have the same priorities then we use the lane congestion and vehicle wait time to operate a signal. We demonstrate the application of our proposed approach using different cases of a traffic case study.*

### KEYWORDS

Multi-Agent System, Exigency Services, ARTIS Agent, transportation system.

### JOURNAL INFO

HISTORY: Received: January 08, 2023

Accepted: February 26, 2023

Published: March 5, 2023

### INTRODUCTION

An agent is an entity or a program that acts on the behalf of a user. In the background, agents perform many tasks by gathering and processing information. A Multi-Agent System (MAS) is a collection of agents and their environment. In the modeling and simulation of road traffic management MASs play a vital role.

Problem-solving techniques are being evolved in a distributed environment due to the introduction of agent technology and due to its immense capabilities in solving real-world complex problems, agent technology has gained vital importance [1]. In a distributed environment, multiple agents work together to achieve a single and common goal. Multi-agent System works on the principle of combining and coordinating the knowledge of every single agent to find the optimal solution for the problem [2]. By using MAS, the analysis of the problems of the distributed problem domain, dynamic environment containing subsystems, and their communication with each other can be resolved. And as the transportation system carries the same problems, MAS can solve them [3]. Work is being done for traffic signal automation using the MAS, but the focus remained on traffic regulation, thus ignoring the prioritization of emergency vehicles which includes ambulances, fire trucks, police

mobility, bomb disposal squad, Rescue 1122 vehicles, Turn Table Ladder (TTL), etc. This paper presents a solution for handling the prioritization of these emergency vehicles by using MAS-based architecture. The agents in this architecture perceive the vehicle data through a video on the runtime from the external world and decide based on recorded and collected information. The quality of service in these exigency vehicles depends on extremely delicate factors, MAS provides a mechanism for efficient transportation and communication through its powerful decision-making mechanism. This paper proposes an architecture that changes the priority of different vehicles to handle the emergency services vehicles problem. The architecture works by using a priority-switching mechanism at the conjunction of the roads and also measures the vehicle's distance from the destination to resolve the conflicts imposed due to the opposite direction vehicles.

Different types of conflicts that could occur throughout the day and could be resolved by this MAS-based transportation architecture are discussed below.

1. DEV: Different Exigency Vehicle (DEV) conflict occurs when two or more exigency vehicles emerge from two or more different directions.

2. CVT: Congestion Versus Time (CVT) conflict occurs when vehicles with the same priority appear at the same time.



This work is licensed under a [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/).

3. RR: Re-Route(RR) occurs when vehicles with the same priority emerge simultaneously with the same congestion value.

4. P or C req: A parallel or concurrent request from neighboring agents occurs when the neighboring agents encountering any of the above conflicts send a request for traffic acceptance towards itself.

A lot of research has been done in MAS to handle autonomous traffic signals, but priority switching for emergency vehicles hasn't been discussed in any research. The main concern of this paper is how to ensure the prioritization of emergency vehicles in an automated traffic environment. Also, what parameters are used to resolve the conflicts between exigency vehicles. For this, we have proposed a framework that along with traffic regulation decides between different vehicles for their priority.

The rest of the paper is structured as follows. In section 2, a comprehensive literature review of related work is presented. The proposed approach for formal modeling of social laws is presented in section 3. Section 4 demonstrates the application of the proposed approach using a case study. In section 5 we conclude the paper.

#### RELATED WORK:

Multi-agent systems have been used in the past for various areas of road traffic management, including modeling and simulation, but using them to solve problems in managing emergency services is a new idea. A study [1] presented an overview of the creation and use of multi-agent systems. This article describes the behavior and basic structure of multi-agent systems. To save processing costs, it also includes the technique of eliminating specific agents at the beginning of the global optimization problem-solving process. This article [2] provides a detailed analysis of reinforcement learning (RL) applications for traffic management systems. It also presented a detailed case study that illustrates the multi-agent architecture. Three reinforcement algorithms were investigated for testing purposes. It explores multi-agent systems' reinforcement learning challenges. Parallel methods of traffic control are discussed in the paper [3]. By comparing them, all the components of conventional traffic management systems and their parallel traffic control strategies are demonstrated.

In-depth research on parallel traffic control systems is presented in [4] and each module of the traditional traffic management approaches is contrasted and compared with its parallel alternatives. In [5], highlighted the development of a novel algorithm called distributed team trigger that integrates many concepts from self-stimulating control with the concept of "promise". However, since agents are often unaware of their neighbors, these assurances enable them to make wise selections. A study [6] presents cellular automata and multi-agent systems based on microscopic simulation. The simulation is crucial for giving current information on road capacities and supporting research into the complexity of traffic systems. Two fundamental issues

with emergency convoys are philosophically addressed in [7]. The first issue focuses on resolving the traffic condition, while the 2nd focuses on convoy orientation, with the basic emphasis on a dispersed and supportive answer to the key challenge. Three steps are suggested by the framework in [8] the creation of the e-questioner, the procedure of social grouping, and the process of tactical adaptation. Firstly, interviews are performed at several colleges with a range of individuals from a variety of backgrounds and with a variety of issues. After processing the abovementioned inputs, a multi-agent system assesses people's preferences in the next stage. In the last step, the facilitator agent publishes the data that has been examined above on the website. This technique of understanding people's preferences leads to a self-adaptive website that presents content depending on their choices.

The objective of [9] is to handle problems associated with equipment repair and maintenance while adhering to financial restrictions. It has given rise to a concept that covers all the operations necessary for maintenance services. The idea involves a system based on agents for controlling the MRO process, which was designed to maintain and repair equipment. The study [10] has adapted the idea of multi-agents to protein synthesis. This method combines the proteins automatically by using existing DNA sources. These proteins are automatically added to a database as long as the DNA source is present. Such a database may be used by biologists in many other fields, such as medicine and pathology, etc. This [11] project's main goal is to construct and deploy ISS module-based MAS in the applications and LAMS-based security. The IIS-based server plugin is called the IIISM monitors and for HTTP protocol requests coming from web applications and Moodle's LMS. MAS architecture was developed using the Gaia approach and implemented using the JADE framework. Currently, the MAS system can detect cross-site scripting and SQL injection attacks on any web application. Without having to completely redesign the MAS system, this system may be enhanced by the addition of attack module detection components for every kind of attack.

The lightweight multi-agent-based test tool for analyzing interconnected automobile systems is presented in [12]. Miniature smart vehicles and quadrotor vehicles equipped with WIFI, ZigBee radios, and integrated CPUs are utilized as multi-agents. Consequently, a multi-agent system is capable of transmitting data through the ground as well as from cars to the side of the road. The future development of intelligently interconnected vehicles will benefit from this research. [13] concentrates on a customized multi-agent system that works in a distributed and parallel environment. This research focuses on the issue of the usage of web crawlers and scrapers. The first portion of the article assesses customized systems, while the second phase describes an external tool for web scraping. [14] suggests using a multi-agent system to improve communication between power grid stations. The electricity grid station can communicate and make decisions more quickly and effectively by using several

agents. Java and the JADE programming language were used to construct the system. Agents can communicate intelligently with several renewable sources due to the technology demonstrated in this study. They also demonstrated simulation to show the results of the proposed system. [15] proposed a wireless, high-speed microprocessor that may give emergency vehicles with clear and seamless traffic management, enabling them to reach their destination quickly. When an emergency arises, an RFID tag is attached to an emergency vehicle, and data is sent to the traffic system via an RF transmitter and a receiver. Additionally, this system can identify stolen vehicles as they travel along the route. When the information is identified, it is sent through GSM to the control room so that immediate action may be taken. Another study [16] suggested using the extended Junction Tree Algorithm (JTA) to find the best Joint Action for the traffic network. JTA can handle graphs with a loop in comparison to the max plus method. JTA was developed to deal with problems that mostly relate to the graphical model. A JTA-based RL algorithm has the benefit of being able to solve problems requiring coordinated signal management.

In [17] A complex method, called cuckoo search optimization is proposed for optimizing the settings of intelligent controllers. The adaptive neuro-fuzzy inference system (ANFIS) and neural networks are employed. The comparison also included Q learning and fixed-time controllers. Multiple scenarios were created and tested for the nine "four-way" junctions that make up the traffic networks. The Cuckoo technique of search was used to achieve the results. [18] provides a reinforcement learning method for controlling traffic signals that enhances exploration and shortens suspension and queuing times. The method by which this article operates is as follows. (A) A brand-new parallel and reinforcement learning technique for traffic light management. (b) A recommended technique's ability to raise a problem's quality has been shown in practice. (C) Sort the difficulties that have come up while employing parallel learning. A low latency program is developed by [19]. In-depth descriptions have been given for connected vehicles, smart grids, wireless sensor actuator networks, smart traffic lights, wind farm use cases, and smart traffic lights. Applications for gaming, streaming videos, and augmented reality have also been mentioned. With intelligent control technology known as Fuzzy intelligent traffic signals control, [20] provides a better traffic light system. Hardware elements that can accept messages from signal control hardware were used to build this system. The findings were achieved using a simulation of microscopic transportation.

[21] Uses innovative traffic management to make it easy for emergency vehicles to pass through. RFID tags have been installed in every vehicle. The development of RFID readers has made it easier to find the tags attached to vehicles using RFID technology. The green light goes on if an emergency vehicle drives past it, and it also detects traffic congestion. This allows wireless communication between ambulances and controllers using Zigbee CC2500 and

PIC1678778 systems. [22] suggests a two-level hierarchical traffic management system based on Q learning. The network is separated into various sections, each of which is supervised by a single agent. Tile code is employed as a linear function at the highest level. The experiment shows that hierarchical control improves Q-learning efficiency even for bottom-level agents. The author [23] suggests using a multi-agent simulation approach for transportation planning. It is possible to thoroughly examine adjustments to transportation policy, such as timetable adjustments and infrastructure growth. In this study, the passengers are shown as a distinct entity. This method has been used in a variety of ways to improve traffic signal management. Reinforcement learning is encouraged since it is regarded to be the most successful because of its self-learning capabilities [24]. Additionally, Q-learning and SARSA, two algorithms, were examined and put to the test on a four-legged intersection.

A model that may be utilized with an autonomous vehicle and replace intersection controls is presented in [25]. The findings show that this system can regulate stop signals and reduce travel time and delay for vehicles by up to 89%. [26] suggests that the intersection be resolved by enhancing real-time coordination of the nearby vehicles that are speaking to each other and also in line. The intersection uses a mixed-integer linear programming method to interact with nearby vehicles and make decisions. [27] offers a Framework for traffic signal set-up difficulties. This article addresses level-1 and level-2 difficulties. Level-1 determines the driver's average journey duration, whereas level-2 needs level-1 to achieve network equilibrium. This study used the evolutionary algorithm to break a bi-level difficult issue into sub-levels to address them separately and on a fast basis. [28] proposes the SIMBA architecture which is a MAS-based framework consisting of different agents including an ARTIS agent and also other real-time agents and these agents-based modules. This architecture enables the communication between agents thus maintaining their constraints. In [29], a case study and architecture are proposed that are capable of functioning in a difficult real-time situation. To effectively deal with Timed Communicating Sequential Processes, [30] offers the idea of Resource Time Process Algebra (RTPA) (TCSP). The method in this study, which was implemented, was a real-time task. Then the directions were entered into the event's resource. As a consequence, the findings that may be utilized to increase efficiency in Real-Time systems are more accurate and efficient when resource calculations are performed using this approach. An ant-inspired negotiating method is proposed in [31], in which a collection of agents freely bargains to establish acceptable contract rules. Also presented is a multi-party, multi-phase negotiating procedure. This approach uses two-stage negotiating procedures. Most automated negotiating research has been in the early stages up until now. The difficulty of allocating and reusing software due to the absence of unified knowledge excellence is one of the main factors in completing these kinds of tasks [32]. It has been suggested in [33-35] that MAS provides an efficient

approach for handling temporal tasks with minimal human intervention.

### **PRELIMINARIES:**

This section demonstrates some preliminaries that are important for this research paper.

#### ***ARCHITECTURE OF ARTIS:***

ARTIS architecture is designed to work in a real-time environment which is considered an extended structure of blackboard. It includes Artificial Intelligence (AI) and covers various aspects of real-time systems and a response-based system within its environment. The architecture includes In-Agents and ARTIS agents where the ARTIS agent has in it a few extensions to deal with the real-time environment. ARTIS consists of a hybrid structure that includes effectors, an intelligent server, a sensor set, a control module, and a level for measuring reflexes. Sensors and effectors are used to perceive data from the environment and to give output in the form of action to the environment respectively and both the sensors and the effectors are time-bounded. The control module executes the components of ARTIS in an environment. The reflex level handles the time-bounded event of that environment whereas Intelligent Server (SI) is used to handle events that are not time-bounded. It is also used to handle In-agents at a cognitive level that is considered an internal entity for the ARTIS. In-agent uses reflex as well as cognitive to solve calculations that are time-bounded as well as those that are not time-bounded by first dividing the bigger problems into small ones [36].

#### ***ARTIS AGENT'S ARCHITECTURE:***

The ARTIS Agent's architecture is the extension of the Blackboard model. [31] proposes the modified version of the blackboard to work under a time-sensitive environment. A dedicated agent is deployed to make sure that the time constraints are fulfilled through offline schedule analysis. Sensors are used to perceive information from the environment while effectors are used to respond to the environment. Reflex and deliberative are two classifications that an agent holds and it also provides some domain functionalities that are provided to the ARTIS agents. In-agent resolves two kinds of problems which are critical and uncritical. The critical problem must be solved within a specific timeframe whereas the uncritical problem does not have any time limit but contains layers such as deliberative and reflex. On encountering any real-time task, In-agent responds through reflex or deliberative layer because of the time constraint. The ARTIS agent uses a control module to control In-agent and coordinate its execution by using two sub-modules which are Deliberative Server (DS) and Reflex Server (RS) to control both the critical tasks as well as deliberative ones.

#### ***FIPA PERFORMATIVES:***

FIPA performatives have been used to carry out communications between agents. Performatives are the speech-act theory that denotes the measures used within the message [19]. Some of the performatives that are considered

the standard for FIPA messages are, agree (in case of agreement), accept-proposal (for proposal acceptance), cancel (for agreement cancellation), call for proposal (for proposal action), inform (to inform some other agent), confirm (to confirm about any proposal), disconfirm (confirm negatively about any proposal), inform-if (inform any other agent through a condition), inform-ref (to know about the value as a reference), failure (in case of failure of agreement), refuse (to refuse for any offer), propose (reply to call for proposal), reject-proposal (rejecting any proposal), not understood (when the agent did not understand any information), proxy (message sending according to description), request-whenever (in case of a true proposition), propagate (for the propagation of information), request-when (in case of a true condition), etc.

#### ***Z-NOTATION:***

Z-notation is the language used as a formal specification to model a computer system. It defines the computer system or the program precisely and uses standards that are used by lambda calculus, axiomatic set theory, and linear predicate logic. A math tool kit is also used by this Z-notation and is used in formal languages too.

### **PROPOSED PESAT FRAMEWORK:**

Our proposed framework is Priority of Exigency Service in Agent-based Transportation system (PESAT). It makes use of two different kinds of agents the first one is called the In-agent which is usually called the internal agent of the system and the second one is called the ARTIS agent which is installed at the traffic signals in the real world environment to detect and perceive information from the outer world. These agents work together and coordinate with each other as the In-agent specifies for any exigency vehicle to the ARTIS while the signals are operated by the ARTIS agent. Our framework, shown in Fig. 4.1, consists of five main components each having agents to control its operations. One component includes the ARTIS agent while the rest four contains in it the In-agents. Every component works in a real-time environment and is time-bounded. The Signal controller deals with the performance and efficiency of the signals that work on the designed algorithm and in a case when an exigency vehicle appears, it follows the instruction received from the ARTIS agent. In-agent deployed at every four signals perceives data and notifies the ARTIS agent for any exigency vehicle and in return, the ARTIS agent replies with emergency detection as true when encountering any ambulance or other exigency vehicles.

#### ***IN-AGENT:***

In-Agent perceives real-time traffic environment through sensors and sends this information to ARTIS agent to handle the exigency case by resolving conflicts in exigency services. In-agent communication with ARTIS agent through FIPA messages and getting specific instructions for signal controlling to handle prioritization of emergency services. Due to the concurrent and time-dependent systems, the in-agent works in a strict environment, and the ARTIS agent, on

the other hand, works in a time-bounded environment, therefore, our framework uses an ARTIS agent in a time-restricted environment as it can handle such tasks perfectly.

**ARTIS AGENT:**

ARTIS agents are installed on all the signals on a 4-way road intersection to control each signal independently. In-Agents send all types of data related to exigency services to ARTIS agents with a request for an opening signal. It receives emergency services detection data from In-agents and processes it according to the conflict resolution algorithm

(Algorithm 1). With the help of this algorithm, ARTIS agents perform different mathematical operations by using different parameters like distance, congestion value, and wait time of exigency vehicle at a signal and resolve different types of conflicts that affect the prioritization of exigency vehicles. ARTIS agent responds to In-Agent by using FIPA messages like agree, refuse, or not-understand which are the results extracted through mathematical calculations. ARTIS agent is deployed and working in real-time and under a hard environment therefore it fits best in our architecture.

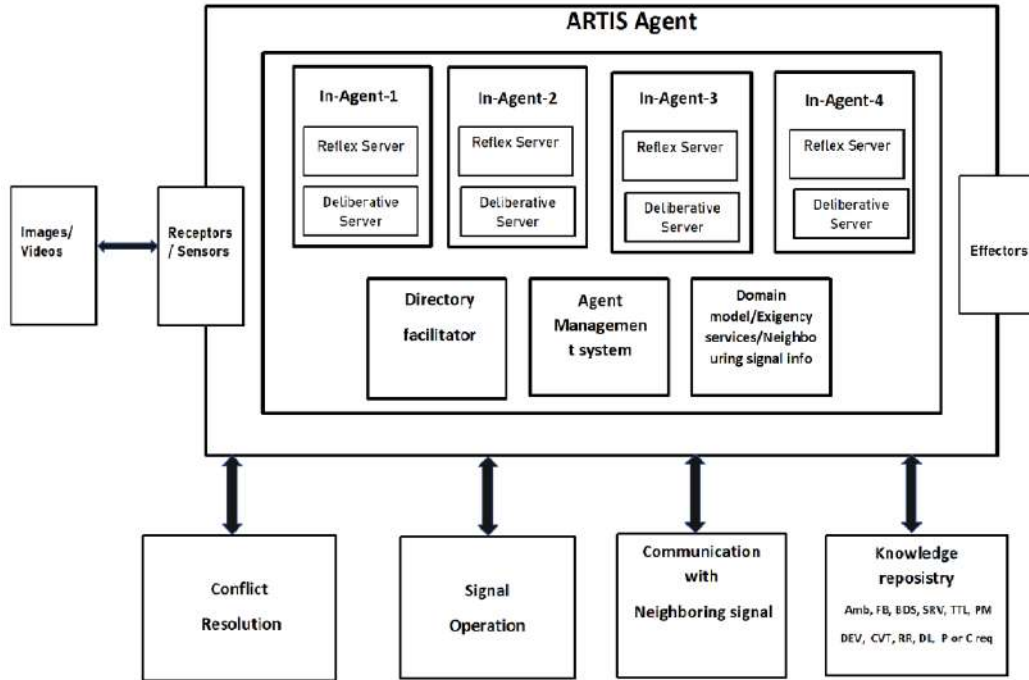


Figure 1 PESAT Framework

**CALCULATING THE EFFICIENCY OF AGENTS IN THE PESAT FRAMEWORK:**

The efficiency and performance of a single agent must be calculated and measured in a routine for checking the

overall efficiency of a system. In eq 1 we proposed a utility function for checking the efficiency of an agent that works in our framework.

$$Utility\ of\ an\ agent = \frac{Total\ No.\ of\ exigency\ services\ passed\ through\ the\ signal}{Total\ No.\ of\ exigency\ services\ reached\ at\ the\ signal}$$

$$U_a = \frac{\sum_{i=1}^m P_i}{\sum_{j=1}^n P_j} \tag{1}$$

$\sum_{i=1}^m$  indicates  $i$  indicates the total no. of emergency services passed through the signal.

$\sum_{j=1}^n P_j$  indicates the total no. of emergency services that arrived at the signal

**FORMAL SPECIFICATION OF AN ARTIS AGENT:**

The working of the ARTIS agent formally describes here deployed at each signal. We have a precise notation for the representation of the working of an agent.

Timed communicating object-Z(TCOZ) language is used for formal specification. InputData represents the type of data that will be processed by the agent. ExigencyType represents one of the predefined exigency vehicles. ExigencyPriority

represents the priority of the identified vehicle. The four In-agents represent the agents that will be responsible for four paths of an intersection. The methods GetInputData and MakeDecision depicts the process of data perception by an agent and then taking an action respectively. Finally, the method OpenSignal depicts the action of opening a signal by an agent in case of an exigency.

*Signal – ARTIS – Agent*

```
inputData : video/image
ExigencyType : {AMB, FB, BDS, SARV, TTL, PM}
ExigencyPriority ← {0, 1, 2, 3, 4, 5}
In – agent : In – agent1, In – agent2, In – agent3, In – agent4
```

```
GetInputData
senseAction :  $\mathbb{P}$  Subsystem  $\times$   $\mathbb{P}$  State  $\rightarrow$   $\mathbb{P}$  State
state' = senseAction(subsystem?, state)
```

```
MakeDecision
execute : {FB  $\times$  AMB  $\rightarrow$  AMB,
SARV  $\times$  FB  $\rightarrow$  FB,
FB  $\times$  TLT  $\rightarrow$  FB,
FB  $\times$  PM  $\rightarrow$  FB,
FB  $\times$  BDS  $\rightarrow$  FB,
FB  $\times$  FB  $\rightarrow$  FollowAlgorithmProcedure,
AMB  $\times$  PM  $\rightarrow$  AMB,
AMB  $\times$  BDS  $\rightarrow$  AMB,
AMB  $\times$  SARV  $\rightarrow$  AMB,
AMB  $\times$  TLT  $\rightarrow$  AMB,
AMB  $\times$  AMB  $\rightarrow$  FollowAlgorithmprocedure,
BDS  $\times$  PM  $\rightarrow$  AMB,
BDS  $\times$  SARV  $\rightarrow$  AMB,
BDS  $\times$  TLT  $\rightarrow$  AMB,
BDS  $\times$  BDS  $\rightarrow$  FollowAlgorithmprocedure,
SARV  $\times$  TLT  $\rightarrow$  SARV,
SARV  $\times$  PM  $\rightarrow$  SARV,
SARV  $\times$  SARV  $\rightarrow$  FollowAlgorithmprocedure,
TLT  $\times$  PM  $\rightarrow$  TLT,
TLT  $\times$  TLT  $\rightarrow$  FollowAlgorithmprocedure,
PM  $\times$  PM  $\rightarrow$  FollowAlgorithmprocedure, }
```

```
rModel' = execute(state, rModel)
```

```
OpenSignal
triggerAction : Open  $\leftarrow$  In – agent & Close  $\leftarrow$  In – agent
In – agent' = triggerAction(state, In – agent)
In – agent' = triggerAction(state, In – agent)
```

MAIN  $\hat{=} \mu T \bullet$  GetInputData; MakeDecision; OpenSignal; T

**Figure 2 Formal Model of an ARTIS Agent**

### ***PESAT ARCHITECTURE WORKING:***

The working of PESAT framework is divided into two phases (i) conflict detection and (ii) conflict resolution. PESAT framework provides the solution to different conflicts that become the cause of delay in exigency services. There are five different conflicts that we have identified in our research. DEV conflict is resolved by assigning the priorities to exigency services on an emergency basis like ambulance having 1st highest priority, fire brigade 2nd highest, and

so on. CVT conflict is resolved by solving the mathematical equation, in which congestion and wait time are used as parameters. The RR conflict is resolved by checking the request of an exigency vehicle for opening a signal, a vehicle requested first will be entertained first. For the DL conflict solution, the ARTIS agent asks for information from the neighboring agents about the route having a low congestion rate. The parallel and concurrent request conflict is resolved through the prioritization of vehicles that we have assigned to each exigency vehicle, a request is accepted only by the neighbor who has the highest priority exigency vehicle. If they have the same exigency vehicles we solve it by using the formula mentioned in equation 2.

$$= \min \left( \sum_{i=1}^m V_i P_i, P_{ni=1} V_j P_j \right) \quad (2)$$

1. Vehicles that have different priorities.
2. Vehicles that have the same priorities.

The vehicle with the highest priority would be allowed to pass through the intersection in case of different priorities while in case of two or more exigency vehicles with the same priority, the vehicle which has been detected earlier would be allowed to pass first.

### ***COMMUNICATION MECHANISM OF MULTI-AGENTS:***

For the internal and external communication of agents, we use FIPA ACL in this proposed framework. For sharing different information between agents Multiple performatives are used. ARTIS agents handle the working of internal agents deployed at each road signal of a 4-way intersection. Audio and video data of traffic perceived by In-agents deployed at each signal. If an exigency vehicle detects through collected data the further processed for the type of exigency service. After this In-agents inform the ARTIS agent about the detection of the exigency vehicle for further processing and ARTIS opens the signal for that particular vehicle on the request. ARTIS agents use congestion and time as a parameter to evaluate mathematical calculations to resolve any conflict that occurs in assigning priorities to different emergency vehicles. The result evolved through these calculations and equations will help to respond with agree, refuse, or not-understand. The ARTIS agent can work under severe and hard circumstances therefore it's used as the base component of our PESAT framework. ARTIS instructs In-agents about opening and closing signals. These instructions help in handling the prioritization of exigency vehicles. All these operations are performed in a real-time environment therefore every agent must work concurrently and dependent on the other. ARTIS agents are time-bounded and this system is also time-dependent therefore both the agents that are ARTIS and as well as In-Agent are included in our framework.

### ***PROPOSED ALGORITHM::***

Proposed algorithm that is used to identify the type of conflict from our identified conflicts and resolve it is shown in Algorithm 1.

**PESAT ARCHITECTURE DEMONSTRATION:**

Our proposed architecture is tested here against different scenarios. Four In-Agents have been deployed on all the signals in a 4-way road conjunction and along with these four agents, one ARTIS is used. In-agents through its sensors acquire information on traffic conditions through video and audio and communicate with ARTIS agents by using FIPA performative “inform” to update the ARTIS agent about the arrival of an emergency vehicle towards that specific road. The ARTIS on receiving a message responds with either the “detection or rejection” status of the exigency vehicle. In-agent on receiving detected status information from the ARTIS agent responds with a request to open the signal of the priority exigency vehicle. The ARTIS agent use algorithm 1 to respond against the opening signal request of In-agents. The ARTIS agent handles the signal accordingly and also informs the ARTIS agent about the arrival of an emergency case. We have discussed five different cases in which we handle the priority of emergency services. In the first case, we discussed two vehicles an ambulance and a fire truck with different priorities appearing at the same intersection at the same time. PESAT resolve it by letting to go ambulance first because it has higher priority as our priority mechanism and after that fire brigade. In the second case, two same types of exigency vehicles arrive at the same time at the same road

intersection on two opposite roads. PESAT solves this conflict by using wait time and congestion value as a parameter in the mathematical formula. In the third case, we resolve conflicts of the same emergency vehicle arriving at the same time with the same congestion value and intersection. The PESAT resolves it by the re-route method. In the fourth case, we have discussed a case when all available roads are checked or congestion on all four sides. PESAT solves it by getting information from neighboring ARTIS agents about the less congested route. In the fifth and last case, we have discussed a conflict of parallel and concurrent requests from ARTIS agents of neighboring signals that they send traffic toward it to handle exigency cases. For succinctness, only one case we have presented below.

**CASE: CVT (CONGESTION VERSUS TIME)**

Here we are going to provide a solution to CVT conflict for handling the prioritization of emergency services. This type of conflict occurred when simultaneously the same exigency vehicles arrived at the same time at the same road intersection on two different roads. You can see in figure 3, two ambulances arrived simultaneously let’s say ambulance V1 arrives on road R1 and ambulance V2 arrives on road R4. In-agent-1 notifies the ARTIS about the ambulance V1 as soon as V1 gets in the range of the agent and the arrival of ambulance V2 is notified to the ARTIS agent by the In-agent-4.

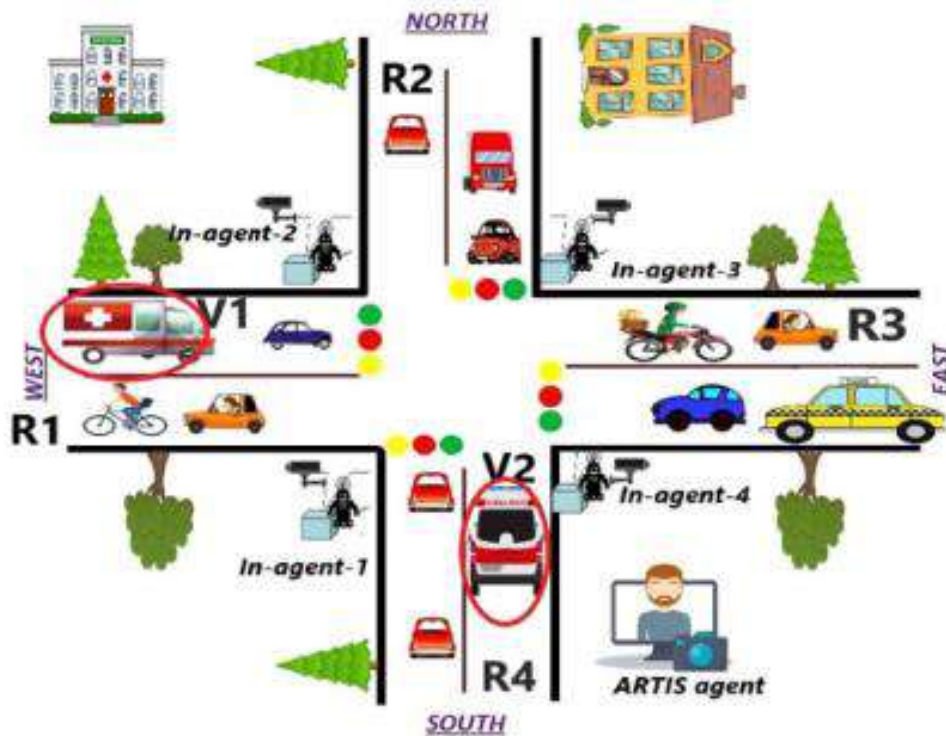


Figure 3 . Ambulance V1 arriving from R4 and Ambulance V2 arriving from R4

**Algorithm 1:** Conflict Identification

---

$ConTypes \leftarrow \{ DL, CVT, P-OR-C-Req, RR, DEV \}$   
 $ExigCategory \leftarrow \{ TTL, FB, SARV, PM, BDS, AMB \}$   
 $Priorityassigner \leftarrow [PM \leftarrow 5, TTL \leftarrow 4, SRV \leftarrow 3, BDS \leftarrow 2, FB \leftarrow 1, AMB \leftarrow 0]$

$conType = \text{conflict types, ExigCategory} = \text{ExigencyCategory,}$   
 $d = \text{data, } v = \text{vehicle, IData} = \text{ImageData, VData} = \text{VideoData}$

**Process** EvaluatingVehicleConflict  
 $DataGetter(d) \leftarrow [VData, IData]$   
 $RecognizedVehicle \leftarrow ProcessData(d)$   
 $IsVehicleExigency(V, ExigCategory)?Ignore:$   
 $CheckExigencyType (V)$

**if** (Number of ExigVehicles == 1) **then**  
    openSignal

**else if**  $Conflicts \leftarrow CheckConflicts(ExigencyVehiclesContypes)$  **then**  
    **for** (each conflict  $\in$  conflicts) **do**  
      **if** (ConType == DEV) **then**  
        Open signal for high priority vehicle.  
      **else if** ( ConType == CVT) **then**  
        Open Signal for having less route value which is calculated by using Congestion and detection time  
      **else if** ( ConType == RR) **then**  
        Open signal whose vehicle's agent requested first.  
      **else if** (( ConType == DL) **then**  
        The ARTIS agent request the neighbouring ARTIS agent for checking a route having less congestion.  
      **else if** ( ConType == P-OR-C-Req ) **then**  
        Resolve by using priority value of exigency vehicles.  
      **else**  
        contype not recognized.

**else**  
    No conflict found.

---

For the resolution of two ambulances conflict, the proposed scenario has been depicted. 0.1, 0.2, and 0.3 taking as congestion values. 0.1 represent minimum congestion, 0.3 for maximum, and for representation of normal congestion we use 0.2. The total 30sec time divide into intervals like 0-10 sec wait time will be assigned 0.3 congestion value, 10-20 sec wait time will be assigned 0.2 similarly 21-30 sec assign 0.1.

Let's assume the wait time for the vehicle is 30 sec. waiting time of vehicle V1 is presented as:

$$V1 \leftarrow \{t11, t12, t13\}$$

Here we use threshold timing first interval of 0-10sec 0.3 waiting time, 2nd interval 11-20sec 0.2 wait time, and 3rd interval of 21-30 sec 0.1 wait time

$$T \leftarrow \{0.3,0.2,0.1\}$$

Congestion value of routes are for V1:

$$V1 \leftarrow \{c11, c12, c13\}$$

$$\text{min} \rightarrow 0.1$$

$$\text{norm} \rightarrow 0.2$$

$$\text{max} \rightarrow 0.3$$

$$C \leftarrow \{\text{min, norm, max}\}$$

$$C \leftarrow \{0.1,0.2,0.3\}$$

WaitTime of V2 is:

$$V2 \leftarrow \{t21, t22, t23\}$$

WaitTime of V2 is:

$$V2 \leftarrow \{c21, c22, c23\}$$

$$\text{min} \leftarrow \{t11, t12, t13\} \cup \{t11, t22, t23\}$$

Total emergency vehicles in this case = 2

$$\text{Vehicle } V = \{v1, v2\}$$

Possible routes for each vehicle = 3

$$\text{Route for vehicle } V1 = \{R2, R3, R4\}$$

$$\text{Unique routes for vehicles } V1 \& V2 \text{ respectively} = \{R1, R4\}$$

$$\text{Common routes for vehicles } V1 \& V2 = \{R2, R3\}$$

Route calculated shown in eq 3:

$$R = \min \left( \sum_{i=1}^3 \sum_{j=1}^3 c_{ij} \right) \quad (3)$$

All the possible combinations of V1 and V2 for the routes are:

$$R = \min ( (0.1 * 0.1) (0.2 * 0.1) (0.3 * 0.1) (0.1 * 0.3) (0.1 * 0.2) (0.2 * 0.3)(0.2 * 0.2) (0.3 * 0.3)(0.3 * 0.2) )$$

$$V1 = \min (R2, R4, R3)$$

$$V2 = \min (R3, R2, R1)$$

Assume the V1 vehicle wait time is 10 sec and the V2 20 sec, calculated route for V1 is given below:

$$= \{(0.3*0.3)(0.2*0.2)(0.1*0.3)\}$$

$$= \{(0.09)(0.04)(0.03)\}$$

$$= 0.000108$$

$$V2 = \{(0.3*0.2)(0.2*0.2)(0.1*0.2)\}$$

$$= \{(0.06) (0.04) (0.02)\}$$

$$= 0.000048$$

The signal is open for v2.

**FIPA MESSAGES FOR CASE STUDY:**

**Table 1 FIPA messages summary exchange between the agents.**

Sr.	Performative (FIPA)	Sending Agent	Receiving Agent	Information
1.	Inform	IA-1	ARTIS	Detection of exigency vehicle.
2.	Request	ARTIS	IA-1	Evaluation of the type of vehicle is requested.
3.	Agree	IA-1	ARTIS	Request agreed.
4.	Inform	IA-1	ARTIS	Ambulance detected through response.
5.	Request	ARTIS	IA-1	Congestion value and wait time are requested.
6.	Agree	IA-1	ARTIS	Request agreed.
7.	Inform	IA-1	ARTIS	The requested congestion value and wait time are sent through inform.

8.	Inform	IA-4	ARTIS	Detection of exigency vehicle.
9.	Request	ARTIS	IA-4	Evaluation of the type of service is requested.
10.	Agree	IA-4	ARTIS	Request agreed.
11.	Inform	IA-4	ARTIS	Ambulance detected through response.
12.	Request	ARTIS	IA-4	Congestion value and wait time are requested.
13.	Agree	IA-4	ARTIS	Request agreed.
14.	Inform	IA-4	ARTIS	The requested congestion value and wait time are sent through inform.
15.	Request	IA-1	ARTIS	Request for signal opening.
16.	Agree	ARTIS	IA-1	The request agreed upon by the agent.
17.	Request	IA-4	ARTIS	Request for signal opening.
18.	Refuse	ARTIS	IA-4	Request refused.

### **SIMULATION OF OUR PROPOSED PESAT FRAMEWORK:**

The working of our proposed PESAT framework is demonstrated with the help of some case studies in this section. Internal agents perceive data from a real-time environment through video and in case any exigency vehicle arrives on any specific road, the ARTIS agent is informed by these internal agents through inform FIPA performative, and in reply, the ARTIS agent responds with the status whether or not the exigency vehicle is detected or not. In-Agent opens that specific signal of the road where the emergency vehicle arrived after receiving the detection status message from the ARTIS agent, who uses algorithm 3.1. for its response. API for Google Maps locates the nearest hospital, the shortest distance to the fire location, and the nearest exit for the police mobile in case of an emergency vehicle. ARTIS agent not only handles one specific signal but intimates all the corresponding signals' agents on the route of the exigency vehicle about the emergency. There had been various scenarios that we have proposed solutions to through our framework. The first scenario is in which two vehicles with different exigency value appears at the same time, let's say an ambulance and a fire truck, our framework will let the ambulance move first before the fire truck because of the prioritization value of the ambulance. The second scenario is in which two emergency vehicles with the same exigency value appear at the same road intersection and at the same time but there is more congestion and less distance or the other way. The third scenario is in which two emergency vehicle with the same exigency appears on the same opposite road at the same time having the same congestion rate and distance from the destination. The fourth scenario deals with the congestion on all available paths in case an ambulance

arrives. The fifth scenario discusses a case where two vehicles say a fire truck and an ambulance appear at the same time in the same conjunction on opposite roads and their priorities are switched through the retrieval of information from the neighbor's agent and changes the priority of the fire truck. This category also includes any contradictions against the functional requirements of any agent. The neighboring agents send P or C requests to the agents allowing them to redirect their traffic toward those requesting agents. Explanations regarding every scenario are briefly described below.

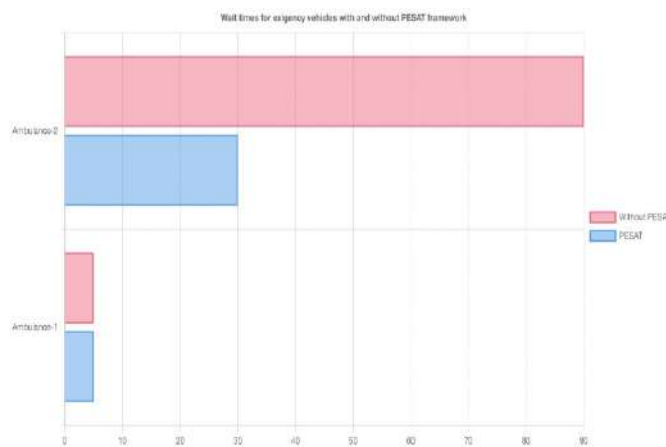
### **DISCUSSION AND RESULTS:**

In agent-based traffic management systems, emergency services handling is a very big problem in urban cities. Some research like with the help of siren detection systems we detect ambulances has been done. But to handle prioritization of exigency vehicles by resolving different types of conflicts has not been discussed in any of the research. PESAT architecture overcomes this problem. It is an agent-based transportation system consisting of an ARTIS agent and four internal agents deployed at each signal node at a 4-way road intersection. In-agents send collected data from each signal node and pass it to the ARTIS agent. The ARTIS agent calculates it with help of Algorithm1 and provides the solution for the occurred conflict to handle an emergency case. Multi-agent decides the calculation of mathematical equations through the processing of information received by the ARTIS agent.

We performed different experiments by varying the exigency vehicles. We perform experiments with and without the PESAT framework.

- PESAT framework: By using this approach, the exigency vehicles let to cross the signal in a very short period of waiting time.
- Normal Approach: There is no method used in this approach for opening signals, for more than one exigency vehicle. Hence, the exigency vehicles cross the signal in a non-prioritized manner just like the other vehicles in traffic.

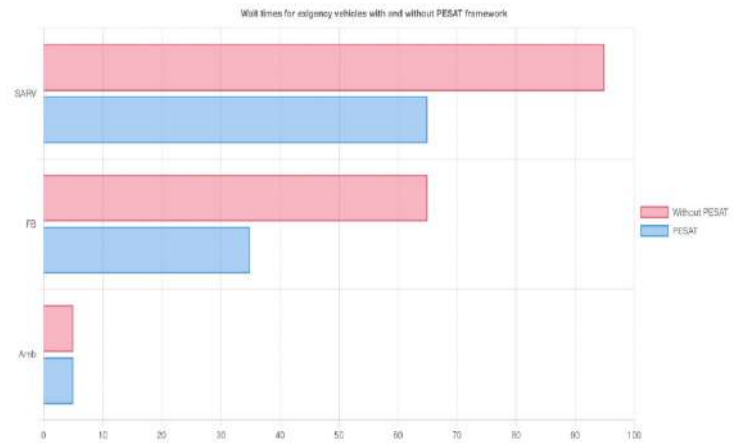
The experiments were performed using different exigency vehicles loaded at a signal at a time. The higher number of exigency vehicles, the higher the traffic load. The waiting time for each exigency vehicle also increased and the risk of loss of lives and property also increased. The graph in Figure 6.1 shows the comparison between the PESAT framework and the Normal approach.



**Figure 4. Difference in wait times with and without the PESAT framework for two vehicles.**

Exigency vehicles are plotted at the y-axis and the wait time of each exigency vehicle is plotted at the x-axis. The horizontal red bar graph shows the waiting time of the exigency vehicle at a signal by using the Normal approach and the blue bars by using the PESAT framework. Ambulance-1 shows 5 sec wait time that is used for detection so, both approaches show equal results for Ambulance-1. For Ambulance-2 the waiting time with the PESAT framework is 30 and without PESAT is 90. The graphs clearly show the effectiveness of the proposed framework for handling the prioritization of exigency vehicles. The graphs in Figure 6.2 shows three different exigency vehicles Ambulance (Amb), Fire brigade (FB), and Search and rescue vehicle (SARV) depicted at a time on the same signal. The red bars show the waiting time with the Normal approach and the blue bar shows the result with the PESAT framework. The graph shows the average waiting times of exigency vehicles with varying exigency vehicles under different traffic loads. Wait time is defined as the total time of exigency vehicle has to wait to cross that signal with and without the PESAT framework. The waiting time for Amb is 5 sec with both approaches, for FB with the PESAT approach is 35 sec and without PESAT approach is 65 sec wait time and for SARV

wait time with PESAT approach is 65 sec and for Normal approach is 95 sec. The waiting time is always high in both cases with two same exigency vehicles and with three different exigency vehicles as shown in figures, figure 4 and figure 5 without the PESAT framework because in the normal approach there is no priority given to resolving different conflicts. PESAT framework is completely autonomous and distributed with less risk of human lives and property loss.



**Figure 5. Difference in wait times with and without the PESAT framework for three vehicles.**

**CONCLUSION:**

In our research, we present how multi-agent systems effectively are used for handling the prioritization of emergency vehicles in automated transportation systems in urban cities. Our method demonstrated how to specify the functional requirements of agents in terms of the traffic signal's optimal common global goal. We measure the performance of PESAT agents with the help of our devised utility formula to achieve a common goal. Exigency vehicles categories according to their priority value. In case of conflict, we resolve it by using wait time and congestion values. Agents' communication is presented in the form of FIPA messages to operate a signal operation in a case of handling prioritization of exigency vehicles. Lastly, To demonstrate how our PESAT framework will function to optimize the functioning of the signal, we gave many scenarios of a signal in operation. In our current approach, we ignored an agent system already installed in a vehicle. If an exigency vehicle might consider having an agent system installed, it intimates in advance to In-agent about its arrival at that signal. Due to advance intimation, real-time scheduling can be done to handle the priority of exigency services and resulting in optimized signal operations.

**FUTURE WORK:**

In the future, we plan to devise methods that allow the agents in a platoon to make decisions in contradictory situations. As if implementing a social law contradicts an

agent's functional requirements. Normally functional requirements are above all social laws. We will look for new ways of finding such contradictions at the formal specification level.

## REFERENCES

- [1] Byrski, R. Dreżewski, L. Siwik and M. Kisiel-Dorohinicki, "Evolutionary multi-agent systems," *The Knowledge Engineering Review*, vol. 30, pp. 171-186, 2015.
- [2] M. Fiosins, B. Friedrich, J. Görmer, D. Mattfeld, J. P. Müller and H. Tchouankem, "A multiagent approach to modeling autonomic road transport support systems," *Autonomic road transport support systems*, pp. 67-85, 2016.
- [3] L. Li and D. Wen, "Parallel systems for traffic control: A rethinking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 1179-1182, 2015.
- [4] K. H. N. Bui and J. J. Jung, "Internet of agents framework for connected vehicles: A case study on distributed traffic control system," *Journal of Parallel and Distributed Computing*, vol.116, pp. 89-95, 2018.
- [5] S. L. Bowman, C. Nowzari and G. J. Pappas, "Coordination of multi-agent systems via asynchronous cloud communication," In 2016 IEEE 55th conference on decision and control (CDC), pp. 2215-2220, December 2016.
- [6] M. Chmielewska, M. Kotlarz and J. Waś, "Computer simulation of traffic flow based on cellular automata and multi-agent system," In *Parallel Processing and Applied Mathematics: 11th International Conference, PPAM 2015*, Krakow, Poland, pp. 517-527, September 6, 2016.
- [7] M. Benalla, B. Achchab and H. Hrimech, "A distributed intelligent system for emergency convoy," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 4, 2016.
- [8] M. A. Mahmoud and M. S. Ahmad, "A self-adaptive customer-oriented framework for intelligent strategic marketing: A multi-agent system approach to website development for learning institutions," In 2015 International Symposium on Agents, Multi-Agent Systems and Robotics (ISAMSR) pp. 1-5, August, 2015.
- [9] V. V. Pantelev, V. A. Kamaev, A. V. Kizim and A. V. Matokhina, "Using of multi-agent system to model a process of maintenance service and repair of equipment of a service company," In 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), pp. 1-3, October, 2016.
- [10] M. H. Kermani and Z. Boufaïda, "A modeling of a multi-agent system for the protein synthesis," In 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pp. 1-7, November, 2015.
- [11] F. El Hajj, A. El Hajj, and R. A. Chehade, "Multi-agent system vulnerability detector for a secured e-learning environment," in *Digital Information Processing and Communications (ICDIPC)*, 2016 Sixth International Conference on, pp. 113-118, IEEE, 2016.
- [12] Y. Zhou, J. Wang, H. Du, H. Li, P. Hu, G. Wang, and R. Shao, "The multi-agent based evaluation of connected vehicle systems," in *Vehicular Networking Conference (VNC)*, pp. 131-132, IEEE, 2014.
- [13] Trifa, A. H. Sbaï, and W. L. Chaari, "Evaluate a personalized multi agent system through social networks: Web scraping," in *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2017 IEEE 26th International Conference on, pp. 18-20, IEEE, 2017.
- [14] W. Li, T. Logenthiran, V. T. Phan, and W. L. Woo, "Intelligent multi-agent system for power grid communication," in *Region 10 Conference (TENCON)*, 2016 IEEE, pp. 3386-3389, IEEE, 2016.
- [15] P. Priyanka, V. Sharmila, V. C. Sindhu, and P. Sangeetha, "Intelligent traffic control system for ambulance clearance and stolen vehicle detection," *International Journal of Research and Engineering*, vol. 3, no. 3, pp. 13-17, 2016.
- [16] F. Zhu, H. M. Abdul Aziz, X. Qian, and S. V. Ukkusuri, "A junction tree-based learning algorithm to optimize network-wide traffic control: A coordinated multi-agent framework," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 487-501, 2015.
- [17] S. Araghi, A. Khosravi, and D. Creighton, "Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network," *Expert Systems with Applications*, vol. 42, no. 9, pp. 4422-4431, 2015.
- [18] P. Mannion, J. Duggan, and E. Howley, "Parallel reinforcement learning for traffic signal control," *Procedia Computer Science*, vol. 52, pp. 956-961, 2015.
- [19] S. Araghi, A. Khosravi, and D. Creighton, "A review on computational intelligence methods for controlling traffic signal timing," *Expert systems with applications*, vol. 42, no. 3, pp. 1538-1550, 2015.
- [20] J. Jin, X. Ma, and I. Kosonen, "An intelligent control system for traffic lights with simulation-based evaluation," *Control Engineering Practice*, vol. 58, pp. 24-33, 2017.
- [21] R. Sundar, S. Hebbar, and V. Golla, "Implementing intelligent traffic control system for congestion control, ambulance clearance, and stolen vehicle detection," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 1109-1113, 2015.
- [22] M. Abdoos, N. Mozayani, and A. L. C. Bazzan, "Hierarchical control of traffic signals using q-learning with tile coding," *Applied Intelligence*, vol. 40, no. 2, pp. 201-213, 2014.
- [23] D. S. Grether, "Extension of a multi-agent transport simulation for traffic signal control and air transport systems," in *Proc. IEEE Intelligent Transportation Systems Conf.*, 2011.
- [24] J. Jin and X. Ma, "Adaptive group-based signal control by reinforcement learning," *Transportation Research Procedia*, vol. 10, pp. 207-216, 2015.
- [25] M. Elhenawy, A. A. Elbery, A. A. Hassan, and H. A. Rakha, "An intersection game-theory-based traffic control algorithm in a connected vehicle environment," in *Proc. IEEE 18th International Conf. on Intelligent Transportation Systems (ITSC)*, pp. 343-347, 2015., J. Raphael, S. Maskell, and E. Sklar, "From goods to traffic: first steps toward an auction-based traffic signal controller," in *Proc. International Conf. on Practical Applications of Agents and Multi-Agent Systems*, 2015, pp. 187-198.
- [26] R. Florin and S. Olariu, "A survey of vehicular communications for traffic signal optimization," *Vehicular Communications*, vol. 2, no. 2, pp. 70-79, 2015.
- [27] F. Ashtiani, S. A. Fayazi, and A. Vahidi, "Multi-intersection traffic management for autonomous vehicles via distributed mixed integer linear programming," in *Proc. 2018 Annual American Control Conf. (ACC)*, 2018, pp. 6341-6346.
- [28] Z. Li, M. Shahidepour, S. Bahramirad, and A. Khodaei, "Optimizing traffic signal settings in smart cities," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2382-2393, 2017.

- [29] V. Julian, C. Carrascosa, M. Rebollo, J. Soler, and V. Botti, "Simba: an approach for real-time multi-agent systems," in Proc. Topics in Artificial Intelligence, 2002, pp. 282-293.
- [30] V. Botti, C. Carrascosa, V. Julian, and J. Soler, "Modelling agents in hard real-time environments," in Proc. European Workshop on Modelling Autonomous Agents in a Multi-Agent World, pp. 63-76, 1999.
- [31] Y. Zhu, K. Chen, J. Xia, and G. Zhang, "Resource modeling and analysis of real-time software based on process algebra," in Proc. 2012 Fourth International Conf. on Computational and Information Sciences, pp. 247-250, 2012.
- [32] Qasim et al., "Intelligent agent for formal modelling of temporal multi-agent systems," International Journal on Smart Sensing and Intelligent Systems, vol. 13, no. 1, pp. 1-13, 2020.
- [33] Qasim et al., "Efficient performative actions for e-commerce agents," Applied Computer Systems, vol. 25, no. 1, pp. 19-32, 2020.
- [34] Qasim et al., "Handling temporal constraints in interaction protocols for intelligent multi-agent systems," International Journal on Smart Sensing and Intelligent Systems, vol. 13, no. 1, pp. 1-15, 2020.
- [35] P. K. Haleema and N. Ch. S. N. Iyengar, "Jadetraderpal-jade mobile agent embedded with flexible negotiation strategies for dynamic e-commerce environments," Journal of King Saud University-Computer and Information Sciences, vol. 29, no. 4, pp. 551-560, Oct. 2017. doi: 10.1016/j.jksuci.2017.02.001.