



## Improvement of Multi-agent Routing Guidance with an Intelligent Traffic Light Scheduling and the Ability to Select Intermediate Destinations

M. Habibi<sup>a</sup>, A. Broumandnia<sup>\*b</sup>, A. Harounabadi<sup>c</sup>

<sup>a</sup> Department of Computer Engineering, Qom Branch, Islamic Azad University, Qom, Iran

<sup>b</sup> Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran

<sup>c</sup> Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran

### PAPER INFO

#### Paper history:

Received 29 November 2020

Received in revised form 19 January 2021

Accepted 20 February 2021

#### Keywords:

Multi-agent Routing

Urban Traffic

Intelligent Traffic Light Scheduling

Intermediate Destinations Selection

### ABSTRACT

Controlling traffic congestion and path routing are integral parts of urban development in large cities. Car routing and traffic flow have a direct impact on each other. Therefore, the first step is to determine a criterion for assessing the traffic situation. The type of vehicles should also be considered in routing. Emergency vehicles must arrive at their mission site as soon as possible. Public transportations must also travel according to their plans. Ordinary vehicle drivers can either choose a road as an intermediate destination (place of interest, picking up someone, etc.). In this paper, two new algorithms are proposed to 1) route with intermediate destination selection for ordinary vehicles, and 2) schedule traffic lights to decrease traffic density and routing delay. The first algorithm proposes an agent-based routing model that in addition to finding the least expected travel time (LET) routes, drivers could select a part of the route as intermediate destinations according to their interests, to raise their satisfaction level. The second algorithm considers the density of traffic flow and the presence of emergency vehicles. This algorithm evaluates the status of the traffic flow by the fuzzy logic. The evaluation is conducted by considering traffic flow speed and density. The output of fuzzy logic is used by the Gradational Search Algorithm (GSA). The GSA regards the status of the flow, the priority of the traffic flow, and the distance of the emergency vehicles to the traffic light. The simulation results indicate that the proposed algorithms have better performance.

doi: 10.5829/ije.2021.34.04a.11

### NOMENCLATURE

|           |  |                      |   |
|-----------|--|----------------------|---|
| $u(x)$    | A vehicle can enter a street or not                | $e$                  | Eligibility of members                  |
| $L(x)$    | Vehicle length limitation of a street              | $st_i$               | The status value of the flow            |
| $w(x)$    | Vehicle weight limitation of a street              | <b>Greek Symbols</b> |   |
| $P_i$     | Probability value for each road                    | $\rho$               | The amount of pheromone on each path    |
| $\bar{v}$ | The flow speed                                     | $\alpha_i$           | The green interval assigned to the flow |
| $d_i$     | Distance between the vehicle and the traffic light |                      |   |

### 1. INTRODUCTION

Today, traffic is one of the most challenging issues around the world. Heavy traffic is a major problem for many drivers since it causes delay and stress. The economic impact of traffic management is also important, as it increases fuel consumption per capita, loss of time, and so forth. These problems lead to a challenging issue, namely the Vehicle Routing Problem

(VRP). Classic VRP consists of assigning routes to a fleet of vehicles to provide services to customers [1]. The classical traveling salesman problem (TSP) can be regarded as VRP. This problem is considered deterministic and static; in other words, all information is previously determined, and during the execution of the routing plan, it remains static [2]. VRPs are usually modeled based on graphs. These graphs are composed of

\* Corresponding Author Institutional Email: [broumandnia@gmail.com](mailto:broumandnia@gmail.com)  
(A. Broumandnia)

vertices (representing intersections) and edges (representing streets). Edges generally contain costs that often specify distances or travel times [1]. The VRP includes the search for a path for  $k$  vehicles so that each vertex is met exactly once, and the total routing cost (like the total travel time or length of the path) is minimized.

Furthermore, as the world becomes more complex, our ability to understand it should also increase. A complex system consists of several components called a multi-agent system. Multi-agent systems are a new and promising area in the field of distributed artificial intelligence, as well as in traffic light scheduling [3]. If the system contains many agents, then there will be many interactions that making it difficult to predict the behavior of a complex system. The total behavior of these interacting components is nonlinear. Different systems of the real world can be considered complex systems. An example of a complex system is the traffic system. In this system, cars act as entities (agents), and the system is called a “multi-agent system”. When talking about the traffic, the motion of each vehicle and its response to the movement of other vehicles are considered. For example, when a route is blocked due to an accident, the car must first stop and then follow other cars at a low speed. The combination of the behavior of vehicles creates a multi-agent and complex traffic system.

The type of streets network and its characteristics can broadly categorize traffic simulation. The two main categories for simulators are highway environment and urban environment simulators. Furthermore, traffic simulators can be divided into microscopic and macroscopic groups based on the depth of vision. Microscopic simulators model the flow of traffic using complex mathematical models, often derived from fluid dynamics, with all vehicles alike, and they use input and output variables such as velocity, current, and density. These simulators cannot model complex routes, characteristics of precise traffic control, or different driver behaviors [4-6]. Macroscopic simulators are more useful to simulate large traffic systems requiring no detailed modeling such as highway networks. Although this method is fast and can be useful and accurate, it is unsuitable for urban models [5]. In microscopic simulators, interactions are usually controlled by vehicle following and lane changing models [7]. Microscopic simulators can more accurately model the flow of traffic than macroscopic simulators do, owing to the added details of modeling the vehicles [6]. Microscopic simulators are widely used to evaluate control and new traffic management technologies, and to analyze existing traffic operations [4, 8]. Urban traffic networks, with many streets and intersections, are extremely complicated. In most cases, they must handle a large number of vehicles in a small part of the street, which can lead to more density and complexity [9].

Recently, companies such as Google Maps [10] and Waze [11] have developed navigation applications trying

to find the fastest navigation. Nevertheless, fastness is not the only driver’s concern. However, these applications try to avoid traffic jams, guide routes with less travel time, and suggest multiple ways to navigate to increase the level of satisfaction, but drivers always do not want to travel fast, rather they want to enjoy the ways. In some cases, there could be roads that drivers like to cross even if they have light traffic or their travel time and distance increase (picking up someone, shopping, a memorable path, ease of driving in the driver’s opinion, beauty of a path, etc). Selecting a part of route could enhance the level of the driver’s satisfaction and increase the joy of travel in using these applications.

Moreover, traffic light scheduling algorithms have a considerable impression on the traffic density. These algorithms consider the expected arrival time to the signalized intersection or the density of competing for traffic flows [12]. Therefore, a well-chosen schedule of traffic lights contributes to the efficient processing of vehicle flows at intersections, thereby raising the average speed and decreasing the time spent by cars in traffic jams [13]. Traffic light scheduling algorithms have to evaluate the conditions of traffic flows. Such an evaluation should be conducted by considering one or more parameters. The results of the evaluations must reflect the current conditions and somewhat should predict the traffic flow. Scheduling of traffic lights should be carried out dynamically and intelligently. The proposed algorithm aims to increase the speed of flows in streets leading to the junction, and to decrease the waiting time of vehicles at traffic lights. It aims to provide the least possible delay for emergency vehicles (i.e. ambulances, fire engines, and police scouts) and buses. It schedules traffic lights by using the Gravitational Search Algorithm (GSA). It evaluates the conditions of flows using fuzzy logic. This paper, also, introduces a kind of multi-agent vehicular urban traffic guidance that uses the driver’s selection of multiple roads as intermediate destinations. This system includes finding the shortest path with a less travel time algorithm, dynamic rerouting, and considering the driver’s selection for intermediate destinations.

The remainder of this article is organized as follows: Section 2 discusses related works. Section 3 describes the proposed approach. Section 4 provides the simulation and results, and finally, Section 5 presents the conclusion.

## 2. RELATED WORKS

Multi-agent models have received increasing attention in traffic management, signal control, and route guidance. This method has special benefits such as greater flexibility, faster response, robustness, resource sharing, and better compatibility in aggregating pre-existing and independent systems [14].

Markov and Gray's models predict traffic volume [15]. Genetic algorithms are also used to predict the traffic volume. A comparison has been conducted in literature made to determine the optimal method to predict the short-term traffic volume between the three methods [16, 17]. The source offers model time windows provided by customers [18], which are considered fuzzy random variables. Group intelligence patterns, especially ant colony, have been widely used to solve complex problems. Ant colony algorithm utilizes the behavior of ants and uses the pheromone for routing. There are various conducted researches in this field [19-21].

Recent advances in ICT and accessories, including Global Positioning System (GPS) have focused on deterministic vehicle routing problem (DVRP) In particular, intelligent transport systems (ITS) can collect and process geographic data (such as vehicle position and status, and traffic information) in real-time and design operations [22]. More specific systems, including traffic simulators, are described in literatures [22, 23].

Most old VRPs rely on computing the least expected travel time (LET) paths for each vehicle to reduce total travel time [24, 25] that is not very useful for urban traffic. Furthermore, dynamic traffic, erratic events, and guidance of drivers toward a similar route lead to unstable global traffic behavior. The last situation could be avoided with dynamic user optimal traffic assignment [26] that periodically computes the assignment of traffic flows to routes.

Vehicle routing algorithms in vehicular traffic guidance systems have been modified several times. However, in recent research [27], arriving on time and total travel time have been considered as two important factors in vehicle routing guidance systems. However, the older systems always consider them separately, since they may have conflicts. This article proposes a semi-decentralized multi-agent system that adaptively provides route guidance at each road intersection by considering the intentions of vehicle agents.

An intelligent traffic routing algorithm was introduced in literature [28]. This algorithm considers the real-time traffic flow intending to cross the intersection of interest and schedule traffic light's time phases of each. This algorithm aims to increase traffic fluency by decreasing the waiting time of traveling and to increase the number of vehicles crossing the intersection per second.

Faye et al. introduced an algorithm to exploit Wireless Sensor Networks (WSN) [29]. Sensor nodes are deployed in lanes that sample the number of vehicles passing them. This is used to schedule traffic lights. The sensor nodes are organized in hierarchical order. Each sensor reports to its upper-level sensor. The traffic light scheduling is carried out by the sensor nodes at the third level. Rezgui et al. proposed three different algorithms to schedule traffic lights [30]. These algorithms are

developed by using a heuristic method considering the density of flows as well as the waiting time of vehicles. These algorithms are named Smart Traffic Light Scheduling based on traffic density (STLSD), Smart Traffic Light Scheduling based on Density and delay Time (SDLSDT), and the STLSDT that takes into account emergency vehicles (SDLSDE). The STLSD considers the density of flows for scheduling. The SDLSDT regards density and the wait time of vehicles for scheduling. The SDLSDE improves SDLSDT by considering emergency vehicles. Different configurations of a genetic algorithm to find an effective traffic light schedule, which is based on a genetic algorithm by the process of natural selection, are discussed in literature [12].

Sharma and Indu [31] proposed a new traffic light scheduling, which uses GPS information. This algorithm assumes that a significant proportion of people use map services like Google Maps, and they follow the route direction using these services. Razavi et al. proposed an algorithm by combining Internet of Things (IoT), image, and video processing techniques in [32]. It considers the flow density and the number of passing vehicles. This algorithm employs two approaches: 1) uses an image of the traffic conditions to determine the density of flow, and 2) uses live video to count the number of vehicles in streets leading to junctions.

Hu et al. proposed a hybrid cellular swarm optimization method to optimize the scheduling of urban traffic lights [33]. Particle swarm optimization plays a pivotal role in this framework which exploits two models: Phase scheduling model (PSM) and Phase timing scheduling optimization model (POM). The PSM conserves the scheduling settings and applies transition rules to achieve comprehensive scheduling. The POM is devoted to the optimization of the phase timing schedule.

The main contributions of this paper are summarized as follow:

- Determine a criterion for assessing the traffic situation
- Consider the type of vehicles in routing
- Introduce an algorithm to route with intermediate destination selection for ordinary vehicles
- Introduce of an algorithm to schedule traffic lights to decrease traffic density and routing delay
- Evaluates the status of the traffic flow by fuzzy logic
- Regards status of the flow, the priority of the traffic flow, and the distance of the emergency vehicles to the traffic light with the GSA method

### 3. PROPOSED APPROACH

The proposed vehicle routing solves the complicated routing problems in three levels to make it a simpler

problem. The three levels are local, regional, and urban levels. This kind of partitioning allows solving the routing problem part by part. A grid is laid out in Tehran city. The entire city surface is related to the urban level. The cells in Figure 1 represent the regional level, and the cells in Figure 2 represent the local level partitioning. An agent is placed for each cell at local and regional levels. An agent is also placed at the urban level that is the central agent. The map of city is available for all agents.

A way to evaluate the traffic conditions of each road in the city should be devised. In this paper, the fourth agent is proposed to count the number of vehicles and the speed of traffic flows on each road. The proposed approach needs to place another agent on each road. This agent reports only the number of vehicles and the speed of each traffic flow. These reports are sent to the local agents. The agents evaluate the conditions of each road and report the overall conditions of their cell.

It is obvious that any type of vehicle cannot enter any road. Some roads have traffic limitations. For example, there may be a bridge in the road that cannot bear vehicles with a weight of more than 10 tonnes. These limitations should be considered in the routing. In this paper, we consider only the size and weight of vehicles. Equation (1) evaluates whether a vehicle can enter a street or not. Equation (2) checks the length limitation of the street  $road_l$  for the requested vehicle  $vehicle_l$ . Equation (3) checks the weight limitation of the street  $road_w$  for the vehicle  $vehicle_w$ .

$$u(x) = \begin{cases} 1 & L(x) = 1 \wedge w(x) = 1 \\ 0 & L(x) = 0 \vee w(x) = 0 \end{cases} \quad (1)$$

$$L(x) = \begin{cases} 1 & road_l \geq vehicle_l \\ 0 & road_l < vehicle_l \end{cases} \quad (2)$$

$$w(x) = \begin{cases} 1 & road_w \geq vehicle_w \\ 0 & road_w < vehicle_w \end{cases} \quad (3)$$

**3. 1. Routing at the Local Level** In the proposed algorithm, the driver sets his/her destination by searching



Figure 1. Regional partitioning on Tehran city



Figure 2. Local partitioning

and pinning on the map. The driver is allowed to select a road that he or she wants to pass along the road. This information and the vehicle weight, length, and width are sent to the local agent. Table 1 shows the request format. This request may be sent by the vehicle or the regional agent of the local agent. Each local agent is supervised by a regional agent. All regional agents are supervised by the urban agent.

The map of the city is converted to a directed graph. Each branch in the network of roads is represented by a node. Each node has a label. The information embedded in the label includes the identification (ID) and the coordination of that branch. Each road is represented by an edge. These edges are labeled by the ID, length, traffic flow density, traffic flow speed, and traffic limitations of the road. The local agents use this graph in their routing process.

The local agent locates the source and destination coordination on the map and identifies the ID of the road on which the source and the destination points are placed. The agent should determine whether the routing is within the borders of its cell or not. If the routing is beyond its borders, it forwards the routing request to its regional agent. The regional agent will do the same process. In other words, if the routing is beyond the regional cell, then the urban agent should start the routing process. Figure 3 presents the algorithm.

Let suppose that the local agent has received a routing request whether from the vehicle or the regional agent. This agent extracts the vehicle specifications and its driver’s demands. It starts a routing process based on the

TABLE 1. The request format for a vehicle

| Requested formats |               |                    |
|-------------------|---------------|--------------------|
| Id                | Source Coords | Destination Coords |
| Length            | Width         | Height             |
| Desired Street Id | Arrival time  | Tolerable delay    |

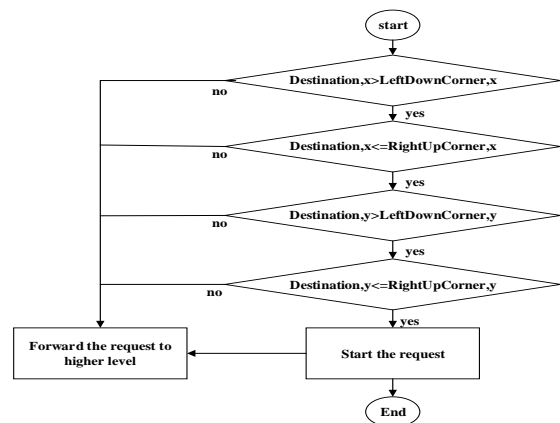


Figure 3. Algorithm of routing at local or higher level

ant colony optimization. It first creates a sufficient number of ants with a label including the vehicle specifications and demands. These ants move in the direction of the flow traffic in which the vehicle is included. When each ant meets a branch, it will calculate a probability value for each road. These values are called the probability selection value. It will, then, define a selection interval for each road. These intervals are used to select the roads randomly. The selection probability value is calculated based on Equation (4).

$$P_i = \begin{cases} \frac{flow\ speed}{max\ Speed + density} & u(x) = 1 \\ 0 & u(x) = 0 \end{cases} \quad (4)$$

The selection interval values are defined based on the algorithm shown in Figure 4. The ant will generate a random number in the interval [0,1]. Each interval is assigned to a road. The road will be selected only if this number falls within its selection interval. The union of intervals is the interval [0,1]. Longer intervals are likely selected. According to the algorithm in Figure 4, those roads having higher selection probability value will have longer selection intervals.

The ants will pass the selected road. The aim of the algorithm in Figure 4 is to discover all the possible paths between the source and the destination. The ants will return to the source only if they meet the destination. The ants will record the ID of roads that they pass. No ant will pass a road twice. If an ant meets a branch or road twice, it will die. The ants have a lifetime so that if they do not meet the destination in their lifetime, they will die. In other words, those ants are lost. If the driver demands to pass a certain road and an ant meets a destination but does not pass the demanded road, the ant will die.

The ants that have succeeded in meeting the destination will return to the source by the path that they have passed along it. They will lay some phomone out according to Equation (5). The Equation calculates the phomone-based on the reported traffic conditions of the

road and the driver's demands, where the  $road_i$  is the length of  $i$ th road along the discovered route.

$$\rho = \begin{cases} \left( \frac{\sum_{i \in path} road_i}{flowSpeed} - \frac{\sum_{i \in path} road_i}{maxSpeed} \right), & Delay_{real} < Delay_{deamanded} \\ \frac{flowSpeed}{maxSpeed + flowDensity + \sum_{i \in path} road_i}, & Delay_{real} \geq Delay_{deamanded} \end{cases} \quad (5)$$

### 3. 2. Routing at Regional and City Level

As mentioned earlier, when a local agent receives a routing request, it will forward the request to its regional agent. The regional agent evaluates the request to determine whether it is within its borders or not. If the request is beyond its borders, it will forward the request to the urban agent. Urban and regional agents will run the same algorithm with a slight difference.

Each upper-level agent will evaluate the conditions of its scope by the report of lower-level agents. Local agents evaluate the conditions of its scope by the reports of traffic control agents. Regional agents evaluate its region by the reports of local agents. Urban level agents evaluate the conditions of the city through the reports provided by regional agents. The agents at these levels perform the same algorithm.

The vehicle routing in these two levels is as simple as drawing a straight line between two points. The two points are the source and the destination of a vehicle. This line will determine local and regional cells that should be activated for routing. The city agent determines regional cells that should go through the routing process. If the traffic condition in a regional cell is more than the threshold, the urban routing agent will replace the cell with its neighbor regional cells. This replacement will be done if the traffic condition of the neighbor regional cell is more than the threshold, otherwise, there will be no replacement.

The urban agent will distribute the driver's demands between the regional agents that should run the routing algorithm. The regional agent will distribute the driver's demands between the local agents. Suppose that a driver has demanded a route to a location at 20 O'clock in such a way that he or she wants to be at the destination in 22 O'clock with a tolerable delay of 30 minutes. Suppose that there are four regional cells between the source and the destination. The urban agent will request the first regional agent to route the vehicle to the out of its borders until 20:30 with a delay of 12 minutes. The other regional agents have to route the vehicle out of their borders at 21:00, 21:30, and 22:00, respectively.

### 3. 3. Traffic Control Using Intelligent Traffic Lights

Traffic lights are mostly located at intersections. In any intersection, there are usually four incoming traffic flows and eight outgoing traffic flows. This algorithm [34] assigns an integer number between 1 and 8 to each outgoing flow and evaluates incoming traffic flows by

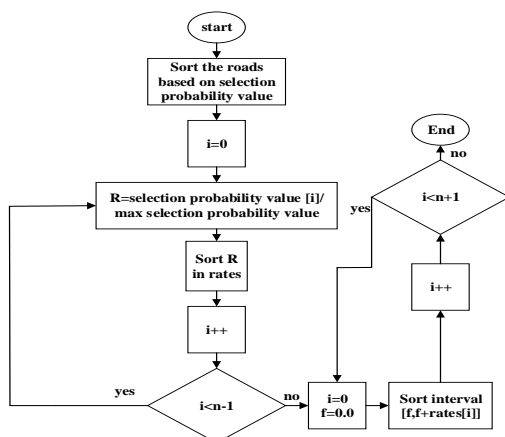


Figure 4. Interval selection setting

considering the speed and density of the flow as well as the distance of the emergency vehicle to the traffic light. Four types of emergency vehicles are considered: ambulance, fire engine, police scouts, and busses. A priority is an integer between 1 and 5 to each flow. The fuzzy engine aims to evaluate the status of each incoming flow to a junction. The evaluations are conducted by two parameters, namely speed, and density of the flow. The outcome of this test is the ratio of vehicles that can pass the traffic light. If the outcome is zero, no vehicle can pass and if the outcome is one, all vehicles can pass the junction. Figure 5 illustrates the fuzzy system.

**3. 3. Scheduling Traffic Lights** Our proposed scheduling [34] is carried out based on the passing ratio, the priority of the flow, and the distance of the emergency vehicle to the junction. Total green intervals in each junction fall into [min, max]. In other words, the  $i$  total green interval could be 3 minutes, and the  $i+1$  is 2.5 minutes. This allows the emergency vehicles, which are close to the traffic light, to pass immediately, and another flow receives the green light. Scheduling is performed by the gravitational search algorithm. The initial population is produced by the algorithm in Figure 6.

The members of the initial population are created randomly. The population for each traffic light may differ

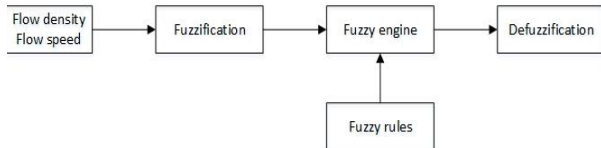


Figure 5. Block diagram of the fuzzy system for vehicle routing.

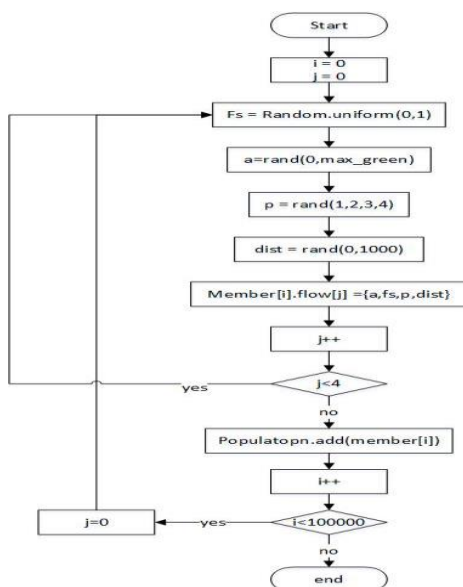


Figure 6. Initial population production

and is optimized gradually. Each member has four flows and an eligibility value, which is determined using Equation (6), where  $st_i$  is the status value of the flow,  $p_i$  is the priority of the flow,  $d_i$  is the distance of the emergency vehicle to the traffic light, and  $\alpha_i$  is the green interval assigned to the flow.

$$A = \pi r^2 = \frac{\alpha_1 \times p_1 \times d_1}{st_1} + \frac{\alpha_2 \times p_2 \times d_2}{st_2} + \frac{\alpha_3 \times p_3 \times d_3}{st_3} + \frac{\alpha_4 \times p_4 \times d_4}{st_4} \quad (6)$$

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$$

**4. SIMULATION AND RESULTS**

The simulations are run on a platform programed in C#. In this platform, the unit of time is second (s), the length is in meter (m) and the speed is meter per second (m/s). A total of 16 regional cells are defined and each regional cell is further divided into 16 local cells. To evaluate the proposed algorithm, three different scenarios are run. The difference between these scenarios is the number of vehicles. There are 144 intersections in the scenarios. In the meantime, the proposed algorithm by Razavi et al. [33] is considered as the base algorithm and have considerable results. Table 2 presents the conditions of the simulation.

**4. 1. The Average Speed of Vehicles** Average speed of vehicles is considered to evaluate the fluency of flows on intersections. This parameter is calculated by averaging the average speeds of all vehicles that traveled to their destination. As Figure 7 shows, the results of average speed for 2500, 5000 and 10000 vehicles determine that the proposed algorithm has better performance.

**4. 2. Average Delay** This parameter shows the average delay of vehicles in intersections. The delay interval of vehicles in the control area of traffic lights is recorded and is averaged at the end of the simulation. The simulation results in Figure 8 indicate that the proposed algorithm had a shorter delay.

**TABLE 2. Simulation Conditions**

| Items  | Value  |      |        |
|--|--------|------|--------|
| Junctions  | 144    |      |        |
| Junctions type   | Grid   |      |        |
| Number of ordinary cars                                    | 2500   | 5000 | 10,000 |
| Number of emergency vehicles (ambulance, fire, and police) | 100    | 200  | 300    |
| Simulation time  | 1 hour |      |        |

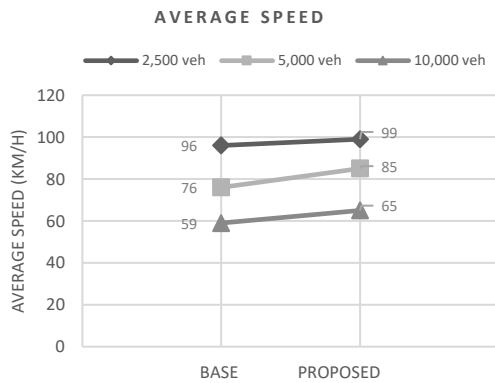


Figure 7. Average speed by the 2500, 5000 and 10000 vehicles.

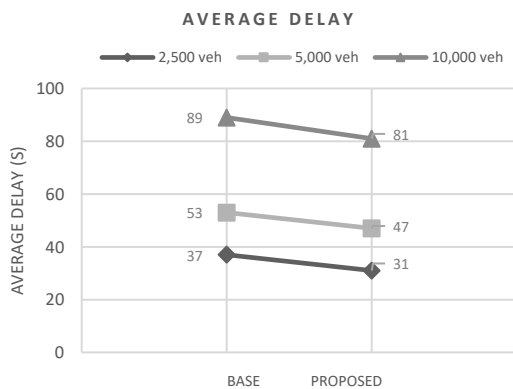


Figure 8. Average delay by the 2500, 5000 and 10000 vehicles

**4. 3. Arrival Time** The proposed algorithm has considered the density of each road and the speed of flow in each road. This strategy resulted in routing with more reliability and had a better prediction.

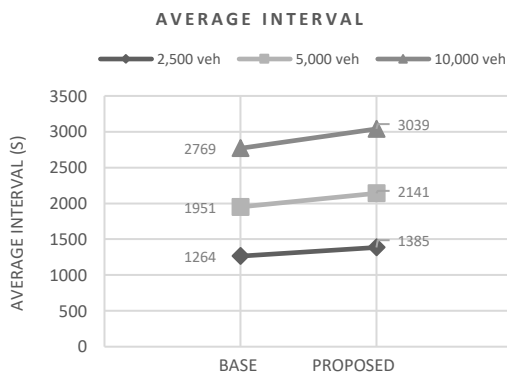


Figure 9. Average arrival time for the 2500, 5000 and 10000 vehicles.

According to Figure 9, the vehicles have averagely arrived at their destination approximately 120 seconds earlier in the 2500 vehicle scenario, 190 seconds earlier in the 5000 vehicle scenario, and 270 seconds earlier in the 10000 vehicle scenarios. As can see, the results are better than the results of the base algorithm.

**4. 4. Total Travel Time** The focus of this research was to route the vehicles in such a way that they arrive at their destination on time. To accomplish this, the total travel time should be reduced. The proposed algorithm, therefore, considered the density and speed of flows. Thus, the discovered paths are faster and more reliable. Furthermore, the paths discovered by the ant colony optimization tend to be the shortest ones, since this optimization method tends to find the shortest path. However, the shortest paths are not always the best paths. This tendency is controlled by Equation (6). The amount of pheromone on each path is a function of the driver’s demands. According to Figure 10, the total travel time of 2500, 5000 and 10000 vehicles was respectively 378s, 466s, and 694s earlier under the routing of the proposed algorithm.

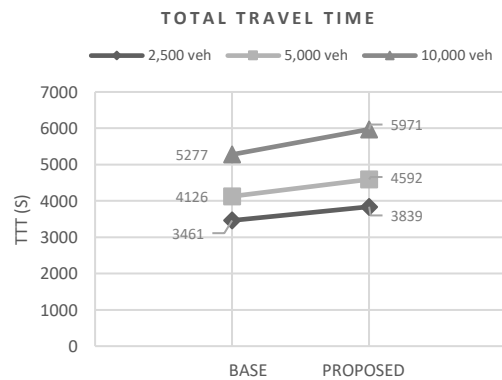


Figure 10. TTT of the 2500, 5000, and 10000 vehicles

**5. CONCLUSION**

Traffic congestion, transportation, and route guidance are integral development parts of large cities. Route guidance aims to find the fastest path from the origin to the destination. This is while drivers may be interested in passing along a particular road as an intermediate destination (picking up someone else, shopping, a memorable path, ease of driving in the driver’s opinion, beauty of a path, etc.).

This paper provides an intelligent solution to control and manage the crossroad traffic using the fuzzy logic and gravitational search algorithm. Scheduling of green

and red traffic lights is determined based on the density and speed of the flows passing through the main streets leading to the junctions. The proposed algorithm has provided less delay, average arrival time, and total travel time for vehicles. It has also considered emergency vehicles and busses. It performed more efficiently than the base algorithm. In future studies, the proposed algorithm will be tested for different penetration rates, and its behavior will be investigated in the case that all vehicles are not equipped with vehicular transceivers.

## 5. REFERENCES

- Ulmer, M.W., Approximate dynamic programming for dynamic vehicle routing, Springer, 2017.
- Helsgaun, K., "An extension of the lin-kernighan-helsgaun tsp solver for constrained traveling salesman and vehicle routing problems", *Technical Report, Roskilde: Roskilde University*. (2017).
- Badjonski, M., Ivanović, M. and Budimac, Z., Agent oriented programming language lass, Object-oriented technology and computing systems re-engineering, 1999.
- Ferber, J. and Weiss, G., Multi-agent systems: An introduction to distributed artificial intelligence, Addison-Wesley Reading, Vol. 1, 1999.
- Lemessi, M., "An slx-based microsimulation model for a two-lane road section", in Proceeding of the 2001 Winter Simulation Conference (Cat. No. 01CH37304), IEEE. Vol. 2, 1064-1071, (2001), doi: 10.1109/WSC.2001.977415.
- Owen, L.E., Zhang, Y., Rao, L. and McHale, G., "Traffic flow simulation using corsim", in 2000 Winter Simulation Conference Proceedings, IEEE. Vol. 2, Issue, 1143-1147, (2000), doi: 10.1109/WSC.2000.899077.
- Schulze, T. and Fliess, T., "Urban traffic simulation with psycho-physical vehicle-following models", Proceedings of the 29th conference on Winter simulation. 1222-1229, (1997), doi: 10.1145/268437.268764.
- Lansdowne, A., Traffic simulation using agent-based modelling, University of the West England Bristol, (2006).
- Nagel, K. and Schreckenberg, M., "A cellular automaton model for freeway traffic", *Journal de Physique I*, Vol. 2, No. 12, (1992), 2221-2229, doi:10.1051/jp1:1992277.
- Google apps. 2021, URL: <https://www.google.com/maps>.
- Waze, *Free community-based gps, maps & traffic navigation app*. 2021, URL: <https://www.waze.com/>.
- Younes, M.B. and Boukerche, A., "An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems", *Wireless Networks*, Vol. 24, No. 7, (2018), 2451-2463, doi: <https://doi.org/10.1007/s11276-017-1482-5>.
- Davydov, I., Tolstykh, D., Kononova, P. and Legkih, I., "Genetic based approach for novosibirsk traffic light scheduling", in 2019 15th International Asian School-Seminar Optimization Problems of Complex Systems (OPCS), IEEE. Vol., No. 31-36, (2019), doi: 10.1109/OPCS.2019.8880158.
- Wahle, J., Bazzan, A.L.C., Klügl, F. and Schreckenberg, M., "The impact of real-time information in a two-route scenario using agent-based simulation", *Transportation Research Part C: Emerging Technologies*, Vol. 10, No. 5-6, (2002), 399-417, doi: [https://doi.org/10.1016/S0968-090X\(02\)00031-1](https://doi.org/10.1016/S0968-090X(02)00031-1).
- Gao, B., Zhang, Q.-Y., Liang, Y.-S., Liu, N.-N., Huang, C.-B. and Zhang, N.-T., "Predicting self-similar networking traffic based on emd and arma", *Journal of China Institute of Communications*, Vol. 32, No. 4, (2011), 47-56.
- Bie, Y., Yang, M. and Pei, Y., "Development of short-term traffic volume prediction models for adaptive traffic control", in International Conference on Advances in Mechanical Engineering and Industrial Informatics (AMEII 2015), (2015).
- Ma, X., Yu, H., Wang, Y. and Wang, Y., "Large-scale transportation network congestion evolution prediction using deep learning theory", *PloS One*, Vol. 10, No. 3, (2015), e0119044, doi: 10.1371/journal.pone.0119044.
- Ma, Y., Feng, C., Zhang, J. and Yan, F., "Judging customer satisfaction by considering fuzzy random time windows in vehicle routing problems", in International Conference on Management Science and Engineering Management, Springer. 204-211, (2017), doi: [https://doi.org/10.1007/978-3-319-59280-0\\_16](https://doi.org/10.1007/978-3-319-59280-0_16).
- Ando, Y., Masutani, O., Sasaki, H., Iwasaki, H., Fukazawa, Y. and Honiden, S., "Pheromone model: Application to traffic congestion prediction", in International Workshop on Engineering Self-Organising Applications, Springer, 182-196, (2005), doi: [https://doi.org/10.1007/11734697\\_14](https://doi.org/10.1007/11734697_14).
- Jiang, S., Zhang, J. and Ong, Y.-S., "A multiagent evolutionary framework based on trust for multiobjective optimization", in *AAMAS*, 299-306, (2012).
- Kurihara, S., Tamaki, H., Numao, M., Yano, J., Kagawa, K. and Morita, T., "Traffic congestion forecasting based on pheromone communication model for intelligent transport systems", in 2009 IEEE Congress on Evolutionary Computation, IEEE. 2879-2884, (2009), doi: 10.1109/CEC.2009.4983304.
- AbdAllah, A.M.F., Essam, D.L. and Sarker, R.A., "On solving periodic re-optimization dynamic vehicle routing problems", *Applied Soft Computing*, Vol. 55, (2017), 1-12, doi: <https://doi.org/10.1016/j.asoc.2017.01.047>.
- Grzybowska, H. and Barceló, J., "Decision support system for real-time urban freight management", *Procedia-Social and Behavioral Sciences*, Vol. 39, (2012), 712-725, doi: <https://doi.org/10.1016/j.sbspro.2012.03.142>.
- Li, J., Wu, Q. and Zhu, D., "Route guidance mechanism with centralized information control in large-scale crowd's activities", in 2009 International Joint Conference on Artificial Intelligence, IEEE. 7-11, (2009), doi: 10.1109/ICAI.2009.41.
- Yamashita, T., Izumi, K., Kurumatani, K. and Nakashima, H., "Smooth traffic flow with a cooperative car navigation system", in Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. 478-485, (2005), doi:10.1145/1082473.1082546.
- Chiu, Y.-C., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, S. and Hicks, J., Dynamic traffic assignment: A primer (transportation research circular e-c153), 2011.
- Cao, Z., Guo, H. and Zhang, J., "A multiagent-based approach for vehicle routing by considering both arriving on time and total travel time", *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 9, No. 3, (2017), 1-21, doi: <https://doi.org/10.1145/3078847>.
- Younes, M.B. and Boukerche, A., "An intelligent traffic light scheduling algorithm through vanets", in 39th Annual IEEE Conference on Local Computer Networks Workshops, IEEE. 637-642, (2014), doi: 10.1109/LCNW.2014.6927714.
- Faye, S., Chaudet, C. and Demeure, I., "A distributed algorithm for adaptive traffic lights control", in 2012 15th International IEEE Conference on Intelligent Transportation Systems, IEEE. 1572-1577, (2012), doi: 10.1109/ITSC.2012.6338671.
- Rezgui, J., Barri, M. and Gayta, R., "Smart traffic light scheduling algorithms", in 2019 International Conference on Smart



- Applications, Communications and Networking (SmartNets), IEEE. 1-7, (2019), doi: 10.1109/SmartNets48225.2019.9069760.
31. Sharma, K.K. and Indu, S., "Gps based adaptive traffic light timings and lane scheduling", in 2019 IEEE Intelligent Transportation Systems Conference, IEEE. 4267-4274, (2019), doi:10.1109/ITSC.2019.8917153.
  32. Razavi, M., Hamidkhani, M. and Sadeghi, R., "Smart traffic light scheduling in smart city using image and video processing", in 2019 3rd International Conference on Internet of Things and Applications (IoT), IEEE, 1-4, (2019), doi: 10.1109/IICITA.2019.8808836.
  33. Hu, W., Wang, H., Yan, L. and Du, B., "A hybrid cellular swarm optimization method for traffic-light scheduling", *Chinese Journal of Electronics*, Vol. 27, No. 3, (2018), 611-616, doi: 10.1049/cje.2018.02.002.
  34. Habibi, M., Broumandnia, A. and Harounabadi, A., "An intelligent traffic light scheduling algorithm by using fuzzy logic and gravitational search algorithm and considering emergency vehicles", *International Journal of Nonlinear Analysis and Applications*, Vol. 11, Special Issue, (2020), 475-482, doi: 10.22075/IJNAA.2020.4706.

---

#### Persian Abstract

---

#### چکیده

تراکم ترافیک، حمل و نقل و راهنمایی مسیر از اجزای اساسی در توسعه شهری در شهرهای بزرگ است. نحوه مسیریابی خودروها در شهرها و ترافیکی که در هر جاده جریان دارد تأثیر مستقیم بر یکدیگر دارند. بنابراین، اولین قدم تعیین معیاری برای ارزیابی وضعیت ترافیک در هر جاده است. نوع وسایل نقلیه نیز باید در مسیریابی در نظر گرفته شود. وسایل نقلیه اورژانس باید در اسرع وقت به محل مأموریت خود برسند. حمل و نقل عمومی نیز باید طبق برنامه ریزی های خود پیش برود. رانندگان وسایل نقلیه معمولی می توانند خیابانی را به عنوان مقصد میانی انتخاب کنند (از روی علاقه، سوارکردن شخصی و غیره). در این مقاله، دو الگوریتم جدید برای (1) مسیریابی با انتخاب مقصد میانی برای عامل های وسایل نقلیه عادی، و (2) برنامه ریزی چراغ های راهنمایی برای کاهش تراکم ترافیک و تاخیر در مسیریابی پیشنهاد شده است. اولین الگوریتم یک مدل راهنمای مسیر مبتنی بر عامل را پیشنهاد داده است که علاوه بر یافتن مسیرهایی با کمترین زمان سفر (LET)، رانندگان می توانند با توجه به علائق خود، بخشی از مسیر را به عنوان مقصد میانی انتخاب کنند تا سطح رضایت آنان افزایش یابد. الگوریتم دوم تراکم جریان ترافیک و حضور عامل های خودروهای اورژانس را در نظر می گیرد. این الگوریتم وضعیت جریان ترافیک را با منطق فازی ارزیابی می کند. ارزیابی با در نظر گرفتن سرعت و تراکم جریان ترافیک انجام می شود. خروجی منطق فازی توسط الگوریتم جستجوی تدریجی (GSA) استفاده می شود. وضعیت جریان، اولویت جریان ترافیک و فاصله وسیله نقلیه اضطراری تا چراغ راهنما را در نظر می گیرد. نتایج شبیه سازی نشان می دهد که الگوریتم های پیشنهادی عملکرد بهتری دارند.

---