An Agent Based Approach for Modeling Traffic Flow

Amgad Naiem*†, Mohammed Reda*†, Mohammed El-Beltagy†
and Ihab El-Khodary†
*Peerialism Inc., Sweden
†Faculty of Computers and Information
Cairo University, Egypt
{amgad, reda}@peerialism.com, mohammed@computer.org, ihabelkhodary@yahoo.com

***Abstract*—Driver behavior is a key factor that gives rise to traffic congestion. In this paper we make use of agent based modeling (ABM) in simulating a traffic system. To investigate how drivers' behavior effect on the performance of the traffic system, we modeled four different types of car drivers. In order to enhance the traffic system's performance, we used a genetic algorithm to optimize the schedule of traffic lights to maximize the overall network throughput. Moreover, we built our model using a 3D modeling environment (Blender) for a more realistic simulation of vehicles' motion.**

I. Introduction

Traffic systems are characterized by a number of features that make them hard to analyze, control and optimize as the number of active participants is high and there is a conflict between the individual driver's self-interest and the overall public utility of the system as a whole. In order to study such a complex system we resort to simulation to understand its salient features. There are many approaches to model traffic flow, one of the most famous approaches in theoretical research is the car-following analysis based modeling described in [1] where a differential equation governs the movement of each vehicle and can be used to predict traffic density. Other approaches are to treat the traffic road as a queue and using queuing models to simulate the traffic system as in [2], [3]. Another perspective used in analyzing traffic systems is the simulation of the travel demands as in SAMS and SMART [4] as well as using discrete event simulation where the simulation is made through discrete time steps as in DEVSIM [5]. Taking the traffic system flows without considering the drivers' behavior is like finding a very efficient way to solve another problem than the one affecting the system as drivers' behavior is one of the key factors responsible for the traffic congestions. In this paper we highlight the effect of the drivers' behavior on the traffic system by using an agent based modeling approach, and combining it with a layer of optimization for traffic lights scheduling.

In section II, we will be discussing the traffic simulation and why we need agent based modeling. Section III will discuss the agent based modeling and how it can be useful in simulating traffic systems. In section IV, we will introduce the model infrastructure and how the model is built using Blender and Python as well as discussing the driver agent types and how it is taking decisions based on a set of rules that will be introduced. We explore optimizating the traffic lights schedule for the system in section V. Finally, in section VI we present a case study for applying our approach on a part of the city of Giza in Egypt.

II. Traffic Simulation

Traffic simulation can be used as a tool to explore traffic systems and test different scenarios that can be helpful to discover the flaws and bottlenecks of the system. To make the model more realistic, we need to consider how drivers interact with the system as they are the core building block of the system. Consequently, this would increase the validity of the results, and the ability to model unusual geometric or traffic control features (such as roundabouts, transit signal priority treatment, and pedestrians) that are not handled in traditional methodologies. Traffic simulation models have the ability to describe the observed spectrum of non-linear phenomena and their characteristic properties. Traffic simulation models that include a real-time animation of traffic also allow non-technical audiences to visualize the potential results of alternative traffic scenarios.

Several commercial packages are available that are designed specially to simulate traffic systems. Some of these packages are TransModeler [6], CORSIM [7], VISSIM [8]. Although there is plethora of traffic simulation packages available, we found that none of these packages allows customization of the drivers' behavior in order to make a more realistic simulation as in our case study such customization is essential.

III. Multi-Agent Based Modeling

Agent-based modeling (ABM) is a relatively new computational modeling paradigm that has been used since the mid-1990s to solve a variety of business and technology problems. One of the successful cases of using ABM is [9] which also used Blender and Python in doing their model. The agent is an autonomous entity that acts according to set of preferences and rules according to which its makes decisions and movements. Emergence of higher order patterns explanation is considered one of the benefits of using ABM. Agents most commonly include these concepts

Autonomy
: Agents have capabilities of task selection, prioritization, goal-directed behavior, decision-making without human intervention

Social ability
: Agents are able to engage other components through some sort of communication and coordination, they may collaborate on a task

Reactivity
: Agents perceive the context in which they operate and react to it appropriately

Multi-agent systems (MAS) are composed of several agents that can manifest self-organization and complex behaviors even when the individual strategies of all their agents are simple. Hence macroscopic system behaviors are then observed as the emergent properties of the MAS.

One of the MAS traffic simulation packages is MATSim which has been used to model the streets of Berlin in a large scale multi-agent simulation involving several hundred thousand agents [10], [11].

IV. The Model Infrastructure

The agent, which is the main brick of our model, is the car driver. There are many types of car drivers each has a different behavior. By using ABM; we could catch the different drivers' behaviors, which was modeled as simple rules for each agent that allow us to catch the complex behavior of the system that cannot be captured without modeling each agent individually.

The greatest challenge is that the observations and reactions of drivers are governed by human perception and not by technology based sensor and monitoring systems, which makes it harder to model this human perception. One of the major difficulties when building such a model is how to integrate the different components to build the infrastructure that allows linkage and communication. The key factor that directs us is that traffic simulation is never convincing until it's clearly visualized. As a result we have chosen Blender a 3D Graphics Modeling environment and Python scripts that acts as general purpose programming language that allow us to access the blender objects to model the agents. To build a traffic simulation model, first we need to build the roads map where the roads are constituted of different lanes where the car is supposed to move in.

The main concept on which we built the animation and the movement of agents is Interpolation Curves. An Interpolation Curve is a curve which describes the variation of an object's attributes over time. Examples of attributes are: Location on the X, Y, Z axis, rotation, size , etc... The use of Interpolation curves in objects motion conceptually very different appraoches that only considers the movement of the objects on X-Y axis in a discrete way. In our work we have made an Interpolation curve for each lane as a track where the cars can move on. The speed of the car movement on the Interpolation Curve is controlled by the length of the time span that the curve points lay on. Actually we are considering the real road as a set of Interpolation curves for each single lane where critical points of the road (Intersections, U-turns and end of roads) are nodes of the roads network where cars can move throughout these nodes in the network. Those critical points would represent the part where the agent should decide where to go. There are many types at either end of road so the agent should change the road or is it out of map so the agent would disappear or a U-Turn so that the agent will be able to turn in the opposite direction if the agent decides to.

Once we have created the Interpolation Curve of each lane of each road we are now ready to attach the car agents represented in the blender by a car object to this Interpolation Curve and advancing the time frame so that the agent would move on the track defined by the curve. On the other hand corresponding to this car object moving on the map in the blender environment we created an object car agent that has set of attributes like (car speed, car lane, car road). Every road is represented by an array of several lanes where each lane is an array of car agents so that the road is multidimensional array carrying the cars circulating in this road. Each car carries an attribute saving the Euclidian distance of the car from the start of the lane. Then the cars of the road are sorted in the array depending on their Euclidian distance from its position to the start of the lane to make the car observe the car in front of it and what is its speed and how far is this car from the deciding car agent.

The cars generation is made in different places of the map. The car generator module watches the generation roads and generates cars if it is needed after testing the availability of space before the last car's position in the generation roads. The generation roads are the beginning segments of the map depending on the road direction, and cars are generated following empirical probability distribution according to gathered data of the area under study. This is also the case for directions choices of cars within intersections in the network.

A. Agent

This is the building block of our model. It contains the agent's internal states as well as the agent's rules. Different values are assigned to these attributes depending on the type of the agent that explained in section IV-B. These attributes are categorized into three categories as follows

1) Location Attributes:

Road
: It is the road that the car is currently in within the map

Euclidian distance from start
: It is the location of the car within the road by calculating the Euclidian distance between the point on the map where the car is located and the start of the road. This attribute is used to find the distance between two agents on the same road

Relative Euclidian distance from start
: As lanes of the same road may have different length, so this attribute defines where the agent would be if it changes to another lane

2) Car's Specific Attributes:

Car quality
: It is the class of the car. It defines how fast is the car and what is the max speed that the car can reach, and also the acceleration and the braking capabilities of the car

3) Driver's Specific Attributes:

Speeding attitude
: It is an attribute that differentiates between the different drivers' behavior regarding their attitude towards speeding and the maximum speed that the driver is willing to reach.

Perception time
: It is the time that the driver needs in order to precive of an existing object in front of him. It also defines the time that the agent needs to take a certain action like realizing that the traffic light is open and starting to move the vehicle.

From these set of attributes, we derive secondary attributes (maximum speed, acceleration, deceleration) that will define how each agent will react. The derivation of these attributes is case dependent. For example, in our case according to the area under study and the real nature of drivers' behavior we derived the maximum speed a driver is willing to reach as follows

$$80 + 100 * \ln((speeding\ attitude * 1.7) + 1)$$

It is assumed that the maximum speed that a driver is willing to drive by in the study area is 180 and this is achieved by having a speeding attitude of 1. we used $\ln$ to make the speed rate of increase for higher values small, as the drivers is not to increase its maximum speed from 170 to 180 as from 80 to 90. also from the equation that the minimum value that the maximum speed can take is 80 and this is achieved when the speeding attitude is zero then the $\ln$ output will be zero as $\ln(1) = 0$. Noticing this value can only be reached if and only if the car that the agent is driving can achieve such speed depending on its quality.

B. Agent Types

In our model there are different agent types that represent different classes of cars that move on the road. These types are as follows

- Private Car Driver
- Taxi Driver
- Minibus Driver
- Bus Driver

All the agents will have the same attributes that do not depend on a specific type while the changing attributes are perception, speeding attitude, and car quality as well as their derivation that is sometimes depending on the type of the agent. An example for the changing of the attributes values depending on the agent type is the speeding attitude of the taxi driver varies according to whether the taxi is occupied or not. Moreover, the agent types will differentiate between using general rules of interaction that are common for all agent types and special rules that defines the behavior of each specific agent type, that will be described in the next section.

C. Agent Rules

In our model, the agents should decide according to set of rules that are categorized into general and special rules. The general rules are common for all agent types, while special rules are types dependent as in figure 1

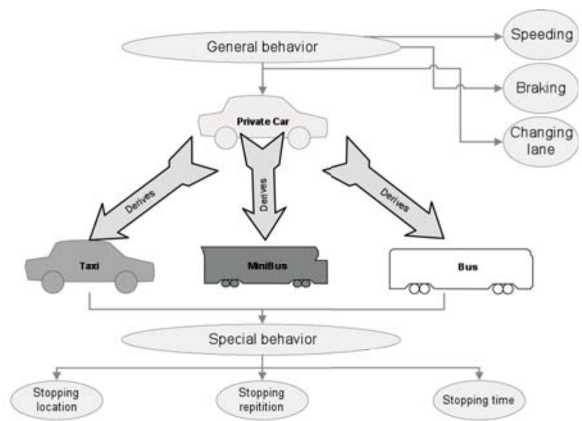

Fig. 1. General and special rules classified according to agent types

1) Speeding: The distance to be traveled by this car in the next time frame is less than the distance traveled by the following car in the next time frame + the distance between the two cars then the car accelerates by certain amount, this amount is determined by the maximum speed and acceleration that are depending on the speeding attitude of the driver and the car quality. But it will not exceed the max speed limit. This means that the car would accelerate as it is allowed till it will has a car in front of it or till it reaches its speed limit. That was for the case where the car needs to increase its speed but what about the car is in a critical situation the agent should decide whether to decelerate or to change lane.

2) Braking: To check if the braking will allow the car to stop before collision is done by knowing the distance between the two cars then deceleration that is determined by the car quality. And using this, we will know if the distance between the two cars is sufficient for braking or not. Changing lane will be the decision of the agent if the braking will cause a collision or if the agent is tending to change lane from the beginning.

3) Changing Lane Rule: Changing lane occurs when the drivers find the incentive to change and when changing lane is safe. The incentive is that the status of the car in the new lane would be better than the status of the car in the actual lane. In other words the distance to next car in the new lane should be greater than the distance to next car in the actual lane. Then comes the safety condition which states that this movement should be safe that there is enough space to make the lane change. And the speed of the car coming from behind in the new lane will not allow it to reach the same position where the car changing lane would be considering the speed of the car that is changing lane. And this would avoid collision when the car is changing lane.

Finally the flowchart in figure 2 summarize the agent general rules.

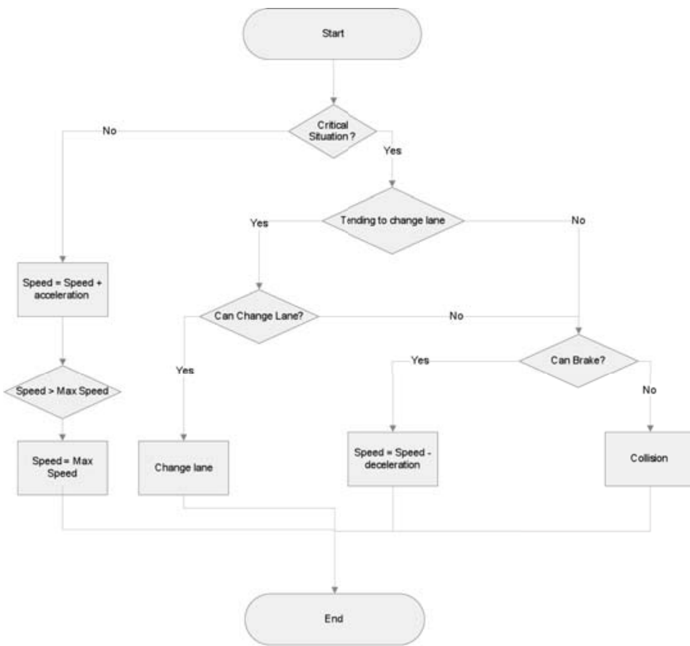

Fig. 2. Flowchart of the agent's general rules

4) Special Rules: Special rules are *stopping location, stooping repetition* and *stopping time* and they are designed for all agent types other than the private car driver and it focus only on the irregularity of stopping which is not the case for private car driver which stops only at its destination. For instance, the bus driver stops at fixed points on the map which are the bus stations, while the taxi driver can stop anywhere to pick up or drop of a customer. We've put the values based on empirical probability distribution according to gathered data for the stopping frequency of each agent type in our case study.

D. Traffic Lights

There are points where the car should take decision to stop or to continue movement. We made the agent observe the traffic lights by inserting an invisible car in its way at the traffic lights node where the agent is forced to stop when the traffic lights are red, and by removing those invisible cars that does not appear on the map off its way when the lights are green. And in this part we are reusing the braking and speeding rules stated above. The traffic lights at intersection are built as one entity that controls all intersections at the same time. This means for example considering traffic lights for two ways intersection if one of them is opened (Green Light) then the other direction will be closed (Red Light) for the same period as the opening period of the Opened direction. For the three ways intersection only one direction is opened at a time and the others are closed for the same period as the opened direction.

V. Optimization

The aim of this study is how we can reach a good enough solution that maximizes the flow of cars passing through the network. More precisely, an optimal solution for a traffic system is something far from reality as it is too complex to reach, so what we mean by optimization is to find traffic lights schedule that maximizes the traffic flow as much as possible. For such a case a genetic algorithm is a suitable approach to use. First the formulation of this problem into a genetic optimization problem is one of the key factors responsible of success or failure of such an optimization. As mentioned in the traffic lights section we have two types of traffic lights those which control two ways intersection and those which control three ways direction.

A. Representation

The chromosome representation is a real encoding chromosome that consists of the time periods for each traffic light in the entire map and depending on the type of the traffic light (2-way or 3-way) the number of cells that define that traffic light is made (either 2 cells or 3 cells) and by changing the numbers in the chromosome the schedule of the traffic lights is changed.

Table I illustrates the encoding of the chromosome that is representing three traffic lights the first one T_1 is a 2 way traffic signal. the two cells for this traffic signal represent the time for each way to be opened or in other words the time of red light and the time for the green light. T_2 represents a 3-way traffic light where each way is opened by the value assigned to the corresponding cell from the three. T_3 is similar to T_1.

TABLE I
Chromosome representation for a network with three traffic lights

T_1		T_2			T_3	
R	G	R	G	R	R	G
1	2	1	2	1	1	2

Then we combine all the traffic lights in the network in one chromosome representing the schedule of the whole traffic lights. For every traffic light this timing would be repeated throughout the simulation time. Actually this approach is efficient as we are considering the gene length would be significant factor in the complexity and the running time. Every gene evaluation means the running of the simulation model for the simulation time which is 1 hour of the peek hours of the day. The memory is one of the aspects that we considered while developing this representation. The chromosome representation as mentioned above is real valued vector holding the length of each time period of the traffic light (Red / Green).

B. Fitness Function

The fitness function of each chromosome or individual is calculated by running the agent-based simulation model with the given traffic lights schedule indicated within that chromosome and evaluate the network throughput throughout the simulation iterations. The throughput of the Network is equal to the number of cars that passes through the network per unit of time and it is calculated as follows.

$$Throughput = \frac{\eta}{\tau}$$

where η is the number of cars passing through the network and τ is the amount of time

C. Survival of the Fittest

The GA Selection that we used is the tournament selections based on many trials tournament selection provides better results. It is simple and fast and maintains sufficient selection pressure. We take two individuals randomly then evaluate their fitness then the fittest is chosen and put into the new population.

D. Crossover

We used arithmetic crossover where the two individuals are divided into parts. A part of the first individual is multiplied by a random weight W_i and added to the corresponding part of the second individual multiplied by $(1 - W_i)$ and so on for the rest of the gene. The recombination happens by a probability of 0.7

E. Mutation

The mutation is used to maintain genetic diversity from one generation to the next. We used uniform mutation such that a time interval of any traffic light is randomly mutated (i.e. changed its value) by +/- 10 sec. The probability of mutation is set to 0.01

VI. Case Study and Results

We applied our approach on a map segment in the city of Giza in Egypt, where the traffic flow data has been gathered . When talking about traffic simulation you are talking about a huge amount of computations. This is due to the large number of entities traversing the map. As the map gets bigger as the amount of computations gets more complicated. The concept of the simulation is that we need every entity to take a step with each time step. So it is clear too that as the number of simulation time increases the complexity increases too. This is seems to be so complicated, but this is nothing compared to an agent based simulation. In an agent based simulation each entity is not just taking a step in the model with each time step but it takes a decision with each time step. And this decision will determine the new state of the agent in the system. In the process of taking decision the agent goes through a huge set of rules depending on how your agent is sophisticated. Now imagine that each agent goes through this set of rules to test a set of conditions and to do certain amount of orders at each time step. That is not all it is a traffic simulation so you need a huge number of agents (cars or drivers) to interact for not short time period that by consequent will make the computations huge. This is why we have very few number of generations in our results.

The traffic lights schedule was optimized with the following parameters of the genetic algorithm: population size 5, number of generations 6, Crossover rate 0.7, mutation rate 0.1, elitism 1 individual from generation to another, while the total simulation time that we used in our experiments was 10 minutes. The final results are presented in table II

TABLE II
Results of the genetic algorithm for our agent-based model

Generation number	1	2	3	4	5	6
Best individual fitness	568	672	704	712	716	720

The individuals' fitness represents the small scale network throughput per 10 minutes. As the best solution is at generation 6 where the throughput reaches 720 cars per simulation time used. The final generation had enhanced the throughput of the network by more than 26%, which is a huge improvement when talking about traffic flows

VII. Conclusion

Through this paper we have presented an agent based approach for modeling traffic simulation. The proposed framework is very customizable and can be used to model any transportation network. This paper presented a more realistic agent based interaction, which makes the results valid for implementation. In this paper we did get very close to the real motion of cars by using Blender as an open source 3D environment and python scripts.

The model is applied in simulating the traffic flow in Giza, Egypt. We have proved by the results how the drivers' behavior is the major factor that gives rise traffic congestions, it is well clear when we tried the scenarios of removing Minibuses (worst behavior) but conserving the same volume of vehicles we discovered how the results are extremely different. What we have achieved is the discovery of a new way to build powerful simulation models from scratch. This model that is built could be an infrastructure for any further study on modeling traffic systems. We have provided an optimization model using genetic algorithm for the traffic signals schedule aiming to reach a near optimal throughput.

The model is useful for transportation planning and traffic networks design and enhancements. We have faced many computational difficulties that are a considerable issue of our model but we have also overcome most of it. But on the other hand all agents based traffic simulation models that have been created used to run with great computational effort most probably on super computers or using parallel processing.

References

[1] D. Gazis, R. Herman, and R. Potts, "Car-following theory of steady-state traffic flow," *Operations Research*, pp. 499–505, 1959.

[2] I. Kosonen and M. Pursula, “A simulation tool for traffic signal control planning,” in *Road Traffic Control, 1990., Third International Conference on*, 1990, pp. 72–76.

[3] K. Hewage and J. Ruwanpura, “Optimization of traffic signal light timing using simulation,” in *Proceedings of the 36th conference on Winter simulation*. Winter Simulation Conference, 2004, p. 1436.

[4] B. Spear, “New approaches to transportation forecasting models,” *Transportation*, vol. 23, no. 3, pp. 215–240, 1996.

[5] D. Donoval, M. Bedlek, M. Barus, J. Racko, and J. Hromcova, “DEVSIM, A new simulator for better understanding of internal semiconductor structure behavior,” *Computational materials science*, vol. 1, no. 1, pp. 51–62, 1992.

[6] Caliper, “Transmodeler traffic simulation software.” [Online]. Available: http://www.caliper.com/transmodeler

[7] McTrans, “Tsis: Traffic software integrated system.” [Online]. Available: http://mctrans.ce.ufl.edu/featured/TSIS

[8] P. America, “Vissim.” [Online]. Available: http://www.ptvamerica.com/software/ptv-vision/vissim

[9] J. Thorp, S. Guerin, F. Wimberly, M. Rossbach, O. Densmore, M. Agar, and D. Roberts, “Santa fe on fire : Agent-based modeling of wildfire evacuation dynamics,” RedfishGroup, Tech. Rep., 2006.

[10] K. Nagel, “Multi-agent transportation simulations,” *Feedback*, vol. 2, p. 3, 2005.

[11] M. Balmer, K. Axhausen, and K. Nagel, “Agent-Based Demand-Modeling Framework for Large-Scale Microsimulations,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1985, no. -1, pp. 125–134, 2006.