Large-scale traffic flow simulation based on intelligent PSO

Zeyu Liu[1], and *Gongping* Yang[2,*]

[1]Jinan Foreign Language School International Center, Jinan 250018, China
[2]School of Software, Shandong University, Jinan 250101, China

Abstract. With the rapid development of urban traffic, a large number of vehicles in cities not only bring convenience to people, but also bring a series of traffic problems, including traffic congestion and high traffic accident rates. Driving speed and waiting time of vehicles are two important factors of traffic problems. To simulate the real urban road traffic flow, a one-dimensional traffic flow grid model was proposed, which considered the nearest and next neighbour car at the same time, and connected the front and rear neighbour cars to optimize the traffic flow. The experiment results showed that our traffic flow grid model can simulate the real urban road traffic flow. In addition, we tried to optimize the urban traffic network model and improved the traffic speed of vehicles and reduced the waiting time.

Keywords: Traffic flow simulation, Intelligent PSO, NetLogo.

1 Introduction

With the rapid construction of urban intelligent transportation systems, large-scale acquisition of urban traffic data has become possible. Common traffic information collection methods include fixed detection represented by induction coil detector and microwave radar detector, mobile detection represented by floating car system, and satellite navigation system. The traffic detector is still an important part of the urban traffic information collection system, and its optimal layout is of great significance.

The rapid growth of the number of vehicles in cities will bring about a series of traffic problems, including traffic congestion and high traffic accident rates. Efficient traffic signal control methods have been proven to be one of the important ways to alleviate traffic problems. The existing signal control mainly focuses on the design of online signal control algorithms, but this method has the problem of frequent signal switching. Modeling of traffic flow on a network has been considered using different approaches. For the present investigation we are interested in traffic flow models for road networks using the above models based on partial differential equations. Two such models were introduced by Holden and Risebro [1]. Both models use a cumulative description of traffic on each road without distinguishing between single lanes. Suitable conditions at the junctions are defined in order to obtain a uniquely solvable network problem.

[*] Corresponding author: gpyang@sdu.edu.cn

This paper is organized as follows. Section 2 contains a description of the related work and the models used for describing flow on the single roads. Section 3 describes the methodology that based on the one-dimensional traffic flow grid model models and will be used in next sections. In Section 4 we design an experiment to verify our approach. In Section 5 We describe the process and the tools that be used. In Section 6 We analyze the results of the experiment.

2 Related work

The micro model generates vehicle motion at a high level of detail: each vehicle is treated as a discrete agent that satisfies certain control rules. For specific urban traffic simulations, a large number of microscopic models have been developed because they can flexibly model the heterogeneous behavior of agents, different road topologies, and the interaction between surrounding vehicles.

Early microscopic models include the cellular automata model and the car-following model. The movement of the vehicle in the cellular automata model is described by the evolution rules in pre-specified time, space, and state variables. Specifically, the road is discretized into units, and the model determines when the vehicle moves from the current unit to the next unit. Due to its simplicity, the cellular automata model has high computational efficiency and can simulate a large number of vehicles on a large road network. However, due to its discrete nature, the generated virtual traffic can only reproduce a limited number of real traffic behaviors.

In contrast, the first car-following model introduced by Reuschel [2] and Pipes [3] can generate real driving behavior and detailed vehicle characteristics, but calculations are required. They assumed that the traffic flow is composed of scattered particles and modeled the interaction between cars in detail. These models express the position and speed of each vehicle through continuous-time differential equations based on a stimulus-response framework.

In the past few decades, a large number of changes and expansions of car-following models have been used to model the response of the host vehicle to the preceding vehicle. Two well-known examples are the optimal speed model and the intelligent driving model.

In the OVM model, it is assumed that the main vehicle maintains the optimal speed. Its acceleration is determined by the difference between its speed and the optimal speed of the vehicle ahead. In the IDM model, the acceleration and deceleration of the vehicle are calculated according to the current speed of the vehicle and the speed and position relative to the preceding vehicle. Vehicle-specific parameters enable IDM models to the preceding vehicle. Vehicle-specific parameters enable IDM models to simulate various vehicle types and driving styles.

In addition to simulating single-lane traffic flow, multi-lane simulation has also been studied. One example is the improved optimal speed model, which is used to simulate traffic on two-lane highways and single-lane highways with on-ramp, another example is the two-lane traffic model, which is used to simulate the horizontal effect of traffic.

To generate detailed simulations, Shen and Jin proposed an enhanced IDM and continuous lane-changing method [4]. Their method can generate traffic flow with smooth acceleration and deceleration strategies and flexible lane suspension behavior. This model revises the original IDM model to make it more suitable for signal processing in urban road networks. Specifically, the acceleration process is divided into a free road acceleration term and a deceleration term. The free road acceleration term describes the driver's willingness to reach the desired speed, and the deceleration term describes the driver's willingness to maintain a safer distance from nearby vehicles. The deceleration item has been modified and an activation control part has been added to make the response of the parked vehicle more stable. The model divides the lane change behavior of urban roads into two situations: free lane change and forced lane change, and provides a flexible continuous model for these two situations.

Under relatively free road conditions, the phenomenon of free lane changes occurs from time to time. This behavior is modeled by the two-lane MOBIL model. Forced lane change is applied to the main vehicle requiring lane change behavior due to some necessary factors, such as racing the end of the lane or turning at an intersection, and the gap between the main vehicle and surrounding vehicles does not support free lane change, land, etc. People extend the complete speed difference model to deal with close-car-braking situations in rural traffic simulation. Later, Lu et al. introduced a personality model in traffic simulation [5].

Compared with the one-way or multi-lane traffic simulation, the intersection traffic simulation is more difficult. Doniec et al. proposed a multi-agent traffic simulation behavior model, which treats intersection traffic as a multi-agent coordination task [6].

In the short-term traffic flow prediction, the traditional PSO optimized neural network prediction model directly takes the boundary value of the escaped particles and has no corresponding mutation mechanism, which is unfavorable for maintaining the diversity of the particle swarm and finding the optimal solution. To further improve the accuracy of short-term traffic flow prediction, based on the traditional PSO optimized BP neural network, boundary mutation operators and self-mutation operators will be introduced to perform double mutation on particles to optimize network configuration parameters. The improved prediction model is verified with the measured traffic flow data. The results show that the model is more conducive to searching for the global optimal solution, and the optimization time is shorter, which can effectively improve the performance of short-term traffic flow prediction.

To tackle these two time-invariant assumptions, a hybrid NN approach integrated with the particle swarm optimization (PSO) approach, namely intelligent PSO (IPSO), is proposed [7]. The IPSO uses the particles in the swarm to represent the neural networks that are used to forecast the short-term traffic flow, without making these two time-invariant assumptions. The IPSO has the following three main features:

1) Flexible neural network structure: In the IPSO, each particle is represented by a three-layer neural network, where switches are configured between links of neural nodes, to determine both the optimal NN structures and the parameters which are very concerning time. Each particle consists of two parts: the integer string and the hierarchical string. The integer string is used to represent the NN parameters. The hierarchical string is used to represent the NN structure. It is represented by the open/close actions of several switches which link the neural nodes. When the switch is opened, the link between the corresponding neural nodes exists. However, the link does not exist when the switch is closed. Based on this particle representation, both optimal NN structures and parameters can be adapted to newly-captured traffic flow data or time-varying configurations of on-road sensor systems. Also, the IPSO can automatically determine the optimal structures of NNs without involving trial and error methods. This is intended to overcome the limitations of the existing NN approaches for traffic flow forecasting in which the NN structure has to be fixed and cannot be adapted with time[7].

2) Active particle movement: In IPSO, the particle movement of classical particle swarm optimization is used, inspired by the social behaviors of animals. The particle movement is used to adapt to the optimal NN parameters and structures for short-term traffic flow forecasting, which is time-varying, since the particle movement can effectively tune the real-time adaptive controllers for many time-varying systems, including the Maglev transportation system and generator system for power applications. This mechanism can also be applied for neural network design effectively. Also, Chan et al. demonstrated that particle movement can effectively adapt optimal structures and parameters of time-varying systems,

where the parameters and the structures of the systems vary with time. Based on the particle movement, the IPSO intends to automatically and effectively tune both the parameters and the structures of the NNs, to obtain an optimal short-term traffic flow forecasting, which is time-varying[7].

3) Further enhancement of particle movement: In the classical PSO, the diversity of the solutions is likely to be lost when a solution with certain good quality is obtained. Hence, the classical PSO is likely to be trapped into this solution and no further progress in terms of better solutions can be made. To further assist the proposed IPSO to search for better solutions, activating components can be injected into the particles to increase the diversity of the particles. In the IPSO, a mechanism based on a fuzzy intelligence system is designed, to maintain the diversity of particles by artificially injecting them with activating components. It monitors the traffic flow accuracies obtained by the IPSO and the changing rates of the traffic flow accuracies. When the traffic flow accuracy is low or the traffic flow accuracy decreases sharply, more activating components are injected into the particles. This is intended to prevent particles pre-maturely converging to solutions with poor traffic flow accuracies and helps the IPSO to move the poor particles from a region with poor traffic flow accuracies to a better region[7].

Applying artificial intelligence to self-driving cars to make traffic run smoothly, reduce fuel consumption, and improve air quality monitoring models may sound like science fiction, but researchers at Berkeley Lab and the University of California Berkeley have launched two research projects to do this. The first project uses reinforcement learning to make self-driving cars drive in a way that increases traffic flow and reduces energy consumption, the second project uses deep reinforcement learning to analyze satellite images, combined with traffic information from mobile phones and environmental monitors to collect data to improve air quality forecasts.

3 Methodology

Based on the one-dimensional traffic flow grid model, a one-dimensional traffic flow grid model that considers the nearest neighbour car and the next neighbour car and considers the interaction of the front and rear neighbour cars to optimize the traffic flow was proposed. The linear stability theory and the nonlinear theory were applied. The analysis is carried out to obtain the stability conditions of the traffic flow, and the equation describing the phase transition of traffic jams was derived. The solution of the equation was verified by numerical simulation. The stability of the traffic flow, and considering the optimized traffic flow of the front and rear neighbouring cars will reduce the stability.

In probability theory and related fields, a stochastic or random process is a mathematical object usually defined as a family of random variables. Stochastic processes are widely used as mathematical models of systems and phenomena that appear to vary randomly. To use the stochastic process, we can analyze the traffic flow model in different situations.

Queueing theory is the mathematical study of waiting lines, or queues as well. A queueing model is constructed so that queue lengths and waiting times can be predicted. Queueing theory is generally considered a branch of operations research because the results are often used when making business decisions about the resources needed to provide a service.

Queueing theory has its origins in research by Agner Krarup Erlang when he created models to describe the system of Copenhagen Telephone Exchange Company, a Danish company[8]. The ideas have since seen applications including telecommunication, traffic engineering, computing[9], and, particularly in industrial engineering, in the design of factories, shops, offices, and hospitals, as well as in project management[10][11].

Using the stochastic process, queuing theory, and dynamic programming, we can realize accurate simulation and derivation of the road traffic flow model. Also, we need network analysis, that is graph theory. In mathematics, graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices (also called nodes or points) that are connected by edges (also called links or lines). A distinction is made between undirected graphs, where edges link two vertices symmetrically and directed graphs, where edges link two vertices asymmetrically.

4 Experiment design

This paper intends to answer whether or not, and in which conditions, does a traffic network system work among a society that uses the same roads to travel in their everyday lives. In the context of our problem, and the efficient result is considered every time the number of users served by the traffic network system is greater than or equal to the number of users not served. For example, considering the total number of people who want to use the traffic network system (whether as drivers or users alone) as 400 people in total, such a system is considered competent enough if at least 200 people are served by it (that is, higher than 50%).

The other issue besides the actual problem is which parameters, constraints, and/or variables we must define in our system to mirror an existing society. We describe every choice, assumption, and absence of our model, and how they are related to each other in the context of the entire system. We envisioned the whole simulation to always be as real as possible, and one of our primary goals was to create a world that mirrored ours as much as possible. As such, this paper also documents our attempts to encounter resemblances to the world we live in, and how we prioritized some similarities and discarded others we figured wouldn’t add much to the final result.

We use NetLogo as a multi-agent modeling environment to create and simulate a traffic network system model. NetLogo is well known to create the desired conditions to simulate social and natural phenomena where people behind the model being created are free to set their rules and instructions that all the agents must follow. We set up the following input variables: grid size, number of cars, number of passengers per car, parking probability and how much time cars remain parked, Accident probability and how much time it takes for the accident to resolve itself and priority areas and what population percentage is stacked upon those areas.

NetLogo is a multi-agent programmable modeling environment. It is used by many hundreds of thousands of students, teachers, and researchers worldwide. It also powers HubNet participatory simulations. It is authored by Uri Wilensky and developed at the CCL. You can download it free of charge. You can also try it online through NetLogo Web.

NetLogo was designed by Uri Wilensky, in the spirit of the programming language Logo, to be "low threshold and no ceiling". It teaches programming concepts using agents in the form of turtles, patches, links, and the observer[12]. NetLogo was designed with multiple audiences in mind, in particular: teaching children in the education community, and for domain experts without a programming background to model related phenomena[13]. Many scientific articles have been published using NetLogo. The NetLogo environment enables the exploration of emergent phenomena. It comes with an extensive models library including models in a variety of domains, such as economics, biology, physics, chemistry, psychology, system dynamics[14]. NetLogo allows exploration by modifying switches, sliders, choosers, inputs, and other interface elements[15]. Beyond exploring, NetLogo allows authoring new models and modifying extant models. NetLogo is open source and freely available from the NetLogo website. It is in use in a wide variety of educational contexts from elementary school to graduate school. Many teachers make use of NetLogo in their curricula.

5 Numerical experiments

In this experiment, we use NetLogo as a carrier for experiments. This software has a model called Traffic Grid. In this model, we can control traffic lights and global variables, such as the speed limit and the number of cars, and explore traffic dynamics. The maximum parameter setup we set in the model is 100. The major differences we compare are horizontal and vertical directions. The number of cars is 100, the car's speed in the model is out of step with reality, so we choose 1 and assume it will be 60miles/h. The tricks per cycle are 20. Let m be the number of vertical lanes and n be the number of horizontal lanes. We did two simulations:

The first condition to consider the square grid with *m*m* layout and we mainly consider these cases: 3*3, 5*5, 7*7, 9*9 lanes. And the second condition is to consider the non-square grid with *m*n* layout. Here, we implemented three kinds of comparisons on the following cases: (1) 6*3 and 9*3 compare with 3*3, (2) 7*5 and 9*5 compare with 5*5, (3) 9*7 compares with 7*7.

In the *m*n* grid, we compared what would happen if the lanes were different in the vertical and horizontal directions. The values that will be measured for comparison are the average waiting time of cars and average speed.

6 Results and analysis

In the first experiment, we record the data in the *m*m* grid. And create a time-average speed Figure 1 and time-wait timeline chart Figure 2. These two charts show that at the same speed and the same number of traffic, the more lanes and intersections there are, the longer the cars wait. But surprisingly, as the number of lanes increased, the average speed increased in Figure 1. However, the data graph also fluctuates more. It can be shown here that although the speed increases in smooth traffic, the speed of multi-lane traffic is much slower in congested traffic.

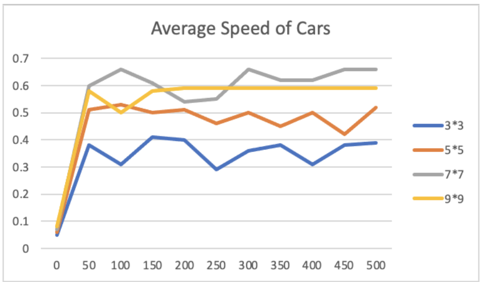

Fig. 1. The average speed of cars in *m*m* grid.

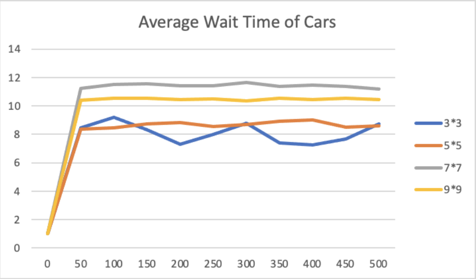

Fig. 2. The average wait time of cars in *m*m* grid.

In the second experiment, we compare the *m*n* grid and the *m*m* grid. The results show that the *m*n* grid's average speed of cars in Figure 3 is greater than the *m*m* grid. Same as the conclusion of the above paragraph, the *m*m* grid has a smaller wait time in Figure 4.

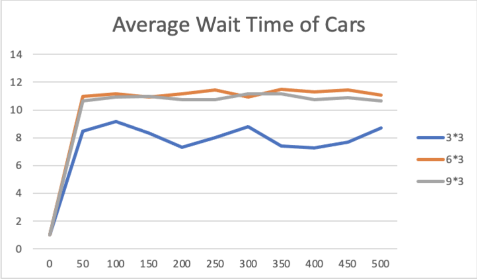

Fig. 3. The average speed of cars in *m*n* grid.

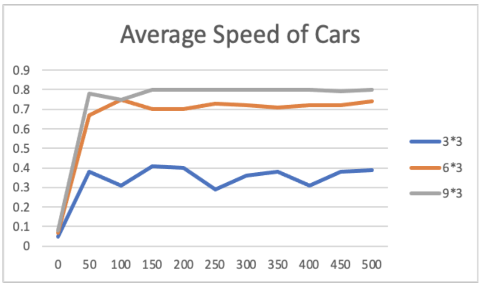

Fig. 4. The average wait time of cars in *m*n* grid.

7 Conclusion and future work

In the simulation, we use a standard graph and simple rules on NetLogo. This will cause my simulation to be a bit limited. These roads are all straight and do not simulate curves, traffic lights can only be set as a single control. Among the whole network, many factors can't be considered in a simple network system.

Many random terms are difficult to be quantified: driving behaviors, differences among individual drivers, emergent situations like traffic accidents will affect the simulation. The driving behaviors in different countries and cultures will cause variable conditions. In certain situations, different individual drivers have different driving habits. Some drivers are aggressive while some are cautious.

In the future, for more complex models, we can try to simulate a two-lane or even a four-lane intersection. On these roads, different lanes play different roles and give different commands to vehicles: left/right turn or turn round. Also, multiple traffic lights controlling will affect the simulation. For example, like in an intersection of an 8-line road and 2-line road, the problem is how to allot lines to maximize the efficiency of traffic. The traffic lights need to be optimized as well since it is very difficult to control the time as there is a huge gap between the traffic flows on the two roads.

Under complicated traffic rules and roads, the more complex simulation needs to be studied. In some countries and regions, the right turn lane does not set traffic lights while others set specific lights to control the right turn lane. Different types of roads such as freeways, urban expressways, and residential streets also have different driving speeds.

This work was supported in part by the Key Research and Development Project of Shandong Province under grant 2018GGX101032.

References

1. H. Holden and N. Risebro, A mathematical model of traffic flow on a network of unidirectional roads, SIAM J. Math. Anal., 26 (1995), pp. 999–1017.
2. A. Reuschel, "Vehicle movements in a platoon," Oesterreichisches Ingenieeur-Archir, vol. 4, pp. 193–215, 1950.
3. L. A. Pipes, "An operational analysis of traffic dynamics," Journal of Applied Physics, vol. 24, no. 3, pp. 274–281, 1953.
4. J. Shen and X. Jin, "Detailed traffic animation for urban road networks," Graph. Models, vol. 74, no. 5, pp. 265–282, Sep. 2012.
5. Lu, Xuequan, et al. "A personality model for animating heterogeneous traffic behaviors." Computer animation and virtual worlds 25.3-4 (2014): 361-371.

6. Doniec, Arnaud, et al. "A behavioral multi-agent model for road traffic simulation." Engineering Applications of Artificial Intelligence 21.8 (2008): 1443-1454.
7. K. Y. Chan, T. S. Dillon and E. Chang, "An Intelligent Particle Swarm Optimization for Short-Term Traffic Flow Forecasting Using on-Road Sensor Systems," in IEEE Transactions on Industrial Electronics, vol. 60, no. 10, pp. 4714-4725, Oct. 2013, doi: 10.1109/TIE.2012.2213556.
8. Sundarapandian, V. (2009). "7. Queueing Theory". Probability, Statistics and Queueing Theory. PHI Learning. ISBN 978-8120338449.
9. Lawrence W. Dowdy, Virgilio A.F. Almeida, Daniel A. Menasce. "Performance by Design: Computer Capacity Planning by Example". Archived from the original on 2016-05-06. Retrieved 2009-07-08.
10. Schlechter, Kira (March 2, 2009). "Hershey Medical Center to open redesigned emergency room". The Patriot-News. Archived from the original on June 29, 2016. Retrieved March 12, 2009
11. Mayhew, Les, Smith, David (December 2006). Using queuing theory to analyse completion times in accident and emergency departments in the light of the Government 4-hour target. Cass Business School. ISBN 978-1-905752-06-5. Retrieved 2008-05-20.
12. Wilensky, Uri (1999–2019). "Types of Agents in NetLogo". The Center for Connected Learning and Computer-Based Modeling (CCL). Evanston, Illinois: Northwestern University. Retrieved 2019-05-03.
13. Kornhauser, Daniel, Rand, William, Wilensky, Uri (November 15–17, 2007). Visualization Tools for Agent-Based Modeling in NetLogo (PDF). Agent2007. The Center for Connected Learning and Computer-Based Modeling (CCL). Chicago, Illinois: Northwestern University. Retrieved 2019-05-03.
14. Wilensky, Uri (1999–2019). "NetLogo Models Library". The Center for Connected Learning and Computer-Based Modeling (CCL). Evanston, Illinois: Northwestern University. Retrieved 2019-05-03.
15. Wilensky, Uri (1999–2019). "Interface Guide". The Center for Connected Learning and Computer-Based Modeling (CCL). Evanston, Illinois: Northwestern University. Retrieved 2019-05-03.