DIVISION OF TRAFFIC NETWORK FOR DISTRIBUTED MICROSCOPIC TRAFFIC SIMULATION BASED ON MACROSCOPIC SIMULATION

Tomas Potuzak[1]

[1]University of West Bohemia, Faculty of Applied Sciences,
Univerzitni 22, 306 14 Pilsen, Czech Republic

tpotuzak@kiv.zcu.cz (Tomas Potuzak)

Abstract

Computer simulation of the road traffic is a very important tool for analysis and control of traffic networks. Since the simulation of a large traffic network (e.g. an entire city and larger) can be computationally intensive, many simulators have been adapted for distributed computing environment. Using this approach, the combined computing power of several computers connected via a computer network is utilized. Nevertheless, an important issue, which must be solved, is the division of traffic network into the required number of traffic sub-networks. A good division method should equalize the load of the simulation processes and/or minimize the inter-process communication. Both of these issues affect the performance of the distributed simulation. In this paper, we present a new method for division of road traffic network. The method is focused on uniform load of the simulation processes, which simulate the resulting traffic sub-networks. Using multiple macroscopic simulation runs, the particular traffic lanes of the network are assigned by weights. Based on these weights, the traffic network is divided into sub-networks with similar numbers of vehicles moving within them. The method is fast and enables to divide a traffic network with hundreds of crossroads in matter of seconds on a standard desktop computer.

Keywords: Distributed traffic simulation, network division, macroscopic simulation.

Presenting Author's biography

Tomas Potuzak was born in Sušice, Czech Republic. He went to University of West Bohemia (UWB) where he studied software engineering and obtained his degree in 2006. Then, he entered Ph.D. studies at the Department of Computer Science and Engineering (DCSE) at the same university and has worked on issues of distributed simulation of road traffic. He obtained his Ph.D. in 2009. He is now teaching assistant at the DCSE UWB. His research is focused on the issues of distributed simulations and component-based simulations.

1 Introduction

The computer simulation of the road traffic is an important tool for analysis and control of road traffic networks. However, a detailed simulation of a large traffic network (e.g. an entire city with hundreds of crossroads) is still problematic. It requires a great amount of computational power to be performed several times faster than in a real time. This is necessary, because multiple simulation runs are often needed in order to guarantee fidelity of the collected results. Hence, many simulators of road traffic have been adapted for distributed computing environment [1, 2, 3]. Using this approach, the combined computing power of several computers (nodes) connected via a computer network is utilized.

An important issue, which must be solved by adaptation of the traffic simulation for the distributed computing environment, is the division of the traffic network into required number of sub-networks. Each sub-network can be then simulated by a simulation process (typically running on one node of the distributed computer). A good division method equalizes the load of the simulation processes [2] and/or minimizes the inter-process communication [1] (see Section 3 for more details).

In this paper, we present a new method for division of road traffic network. The method is focused on uniform load of the simulation processes, which simulate the resulting traffic sub-networks. During the division of traffic network, multiple macroscopic simulation runs are utilized to assign weights to particular traffic lanes of the divided traffic network. Based on these weights, the traffic network is divided into sub-networks with similar numbers of vehicles moving within them.

The rest of this paper is structured as follows. In Section 2, the distributed simulation of road traffic and its issues are briefly described. Section 3 discusses common traffic network division approaches. In Section 4, the newly developed method for traffic network division is described in detail. Section 5 contains the plan of complex testing of the proposed method along with preliminary results. Possible directions of our future research are mentioned in Section 6 and the paper is concluded in Section 7.

2 Distributed simulation of road traffic

The algorithm for the network division is designed for discrete microscopic simulators of road traffic.

2.1 Simulation for network division testing

For testing of the load of the particular simulation processes, the Distributed Urban Traffic Simulator (DUTS) was used. The DUTS system is a discrete microscopic time-stepped simulator developed at our department. Every single vehicle is considered in the simulation and the simulation state is recomputed regularly once per time step. These features are common for most microscopic simulators of road traffic [1, 2, 3, 4].

The DUTS system incorporates three traffic models inspired by three existing traffic simulators – JUTS [5], TRANSIMS [1], and AIMSUN [3]. Besides the JUTS (Java Urban Traffic Simulator), which has been developed at our department, the implementations were realized according to the description in the scientific papers [6]. The utilization of three different microscopic traffic models enables to demonstrate the universality of the developed algorithm for traffic network division.

2.2 Simulation decomposition

The division of the traffic network into several sub-networks represents *spatial decomposition* of the simulation. Using this approach, the simulation is decomposed using a spatial (i.e. geographical) criterion into several parts (traffic sub-networks). Each simulation process then performs simulation of one assigned traffic sub-network. In case of simulation of road traffic, this type of decomposition is most common. There are several commonly used approaches to the traffic network division, which will be discussed in Section 3.

Nevertheless, generally, there are also completely different approaches to decomposition of simulation. Although their utilization in the field of traffic simulation is quite rare, we will briefly describe them in following paragraphs.

First, we will mention the *task parallelization*. Using this approach, the entire simulation program is decomposed into several modules. Each module is then performed on different node of the distributed computer. This approach is quite straightforward. However, the simulation speed is limited by the slowest module. So, it is necessary for all modules to consume similar amount of computing power and time [7]. However, this is not the case of distributed simulation of road traffic, where the major part of the computing power is often consumed by the vehicle movement module and by the statistical results collection module. The requirements of other modules such as map depiction are negligible [8]. Hence, the task parallelization is used only in some special cases of the distributed road traffic simulation (for example, see [9]).

Second, the *temporal decomposition* will be mentioned. Using this approach, the simulation is divided into (equally-sized) time intervals. Every simulation process then performs entire simulation, but only for one time interval [10]. The main advantage of this approach is the possible utilization of massive parallelism, because there is no need for communication among processes during the parallel computation. The main disadvantage is that, after the computation of all temporal intervals, the states on the boundaries of the intervals may not match, especially in traffic simulation, which exhibits very complex states. An example of the utilization of temporal decomposition in field of traffic simulation can be found in [11].

2.3 Inter-process communication

If we consider the most common spatial decomposition of the road traffic simulation, there are several sub-networks, which are interconnected by set of traffic lanes. In order to maintain the passing of the vehicles in these lanes (from one sub-network to another), the simulation processes must communicate with each other during the simulation run. In the distributed computing environment, the only means of communication is the message passing [10]. Hence, the vehicles are transferred in form of messages sent from one traffic sub-network to a neighbouring one [12].

Besides the transfer of vehicles, the communication is used for synchronization. The synchronization is necessary to ensure correct mutual running of the simulation processes (i.e. each process performs the same time step at the same moment). Both tasks – the transfer of vehicles and the synchronization are ensured by a communication protocol [8].

There are many communication protocols for distributed simulation of road traffic, which differ in efficiency and applicability. Various protocols can be sensitive to various features of the simulated traffic network, such as number of divided traffic lanes or vehicle density in these lanes [8]. However, we have developed several communication protocols, whose efficiency is minimally affected by these features [8]. Two of these protocols – D-LS (Distributed Long Step) and D-LSB (Distributed Long Step Binary) are described in [6] and [13] in detail.

3 Traffic network division approaches

There are two main issues, which should be considered during the division of the traffic network into sub-networks. First, the simulations of particular sub-networks should be approximately equally computation-consuming. This is necessary, because the slowest sub-network determines the resulting speed of entire simulation [8].

Second, the communication between the neighbouring traffic sub-networks (i.e. simulation processes) should be minimal. The reason is that the inter-process communication is relatively slow [10]. However, if an efficient communication protocol is used (see previous section), this problem can be minimized.

There are several techniques for division of traffic network. Usually, they are focused on the first or the second issue. However, they can be also focused on both or neither. Three approaches utilized in three different distributed simulators of road traffic are described in following sections.

3.1 Division into equally-sized pieces

The easiest solution is to divide the traffic network into equal-sized pieces. This approach is used for example in *ParamGrid* [4, 14]. The entire traffic network is divided into a grid of rectangular geographic areas arranged in rows and columns. The main advantage is that the entire traffic network can be watched on a grid of displays [4]. Nevertheless, the main disadvantage is that the number of vehicles in every geographic area can be very different because of various road densities and also various vehicle densities in the roads [12].

Another problem can be the potentially large amount of traffic lanes affected by the network division, since the number of lanes that are crossing the boundaries of the sub-networks is not considered [8]. However, this issue can be neglected using an efficient communication protocol (see Section 3).

Considering the issues of this traffic network division approach, it is quite suitable if the traffic lanes density and vehicle densities in the roads (or traffic lanes) are more or less uniform [12].

3.2 Minimization of divided lanes and neighbours

A more sophisticated method for traffic network division is employed in TRANSIMS [1]. The method in this simulator is focused on minimization of the neighbours count of the particular sub-networks.

Using this method, the entire traffic network is divided into sub-networks of similar size. The size is measured as accumulated length of the lanes associated with the sub-network. During the division of the network, the number of divided lanes and the number of sub-networks' neighbours are minimized. For this purpose, the graph partitioning methods are used (e.g. orthogonal recursive bi-section) [1].

As arise from previous paragraph, this method is focused on minimizing of the inter-process communication, but the size of the particular sub-networks is also considered. Nevertheless, the traffic density in the lanes is not considered during the division. If the traffic in particular lanes were very different, the traffic sub-networks could be differently computation-consuming.

3.3 Utilization of vehicles count

An interesting approach can be found in the implementation of the traffic simulator *vsim* [2]. In this case, the traffic network is divided according to the number of vehicles passing along the particular traffic lanes [12].

Using this approach, the resulting sub-networks are similarly computation-consuming. That means that the load of the processes are well-balanced and the simulations of particular sub-networks are running at similar speeds. Hence, there is no "slowest" process, which would delay the distributed simulation.

Similar to the approach described in Section 3.1, the number of divided lanes is not considered during the network division. Again, the potential large number of affected lanes can be neglected using an efficient communication protocol.

There is a bigger issue, however. The division is performed according to the numbers of vehicles in particular lanes. In *vsim*, the numbers of vehicles are collected after one simulation run [2]. However, if the simulation is intended to be distributed, its sequential run can last very long or can be difficult to perform (e.g. due to memory requirements) and the distributed run cannot be performed prior to the traffic network division. Hence, the collection of the data for network division from the simulation can be problematic.

In relatively rare cases, the real measured traffic intensity data can be at our disposal [12]. In that case, these data can be used for network division without necessity to collect them from simulation.

4 Macroscopic-simulation-based division

Generally, the task of the traffic network division is to mark the lanes, which should be divided in order to form required number of traffic sub-networks. Although the lanes can be marked manually (by a human user), we focus only on automatic division. Possible approaches to this issue were discussed in Section 3. A new approach, which we developed, is described in this section in detail.

4.1 Main idea

As it was said before, two main issues of the traffic network division are the well-balanced load of the simulation processes and the minimized inter-process communication. Since the communication issue can be solved using an efficient communication protocol we developed (see Section 2.3), we will now focus on the balancing of the load of the particular simulation processes. The approach is intended for distributed microscopic traffic simulation, which will be performed on a cluster of homogeneous (i.e. with the same computational power) computers.

The main idea of our approach is inspired by network division utilized in the *vsim* [2] simulator (see Section 3.3). This division is based on numbers of vehicles in particular traffic lanes collected after one simulation run. Since the movement of vehicles consumes the major part of the simulation time, a division based on the vehicles counts can ensure well-balanced load of the simulation processes. Nevertheless, utilization of the microscopic simulation for this purpose can be difficult (see Section 3.3).

Hence, we proposed a method that uses the macroscopic simulation to assign weights to the traffic lanes based on traffic flow density in these lanes. These weights can be then used for network division similar to the vehicle counts in the *vsim* simulator. This macroscopic-simulation-based division will be abbreviated as MSBD later in the text.

4.2 Macroscopic traffic simulation description

Unlike the microscopic simulation of road traffic, which deals with particular vehicles, the macroscopic simulation deals with aggregate traffic flows in traffic lanes or roads [15]. The flows in particular streets are described by set of parameters (e.g. mean speed and concentration). No individual vehicles are considered.

These models are the oldest and simplest traffic models used [16] and they exist in many modifications [17]. Because of their simplicity, they are significantly less computation-consuming than their microscopic counterparts. Hence, in most cases, a single-processor computer nowadays provides sufficient power to perform macroscopic simulation of a very large area (e.g. an entire city and larger) many times faster than in a real time [18].

4.3 Macroscopic simulation for network division

Macroscopic simulation used for MSBD method is inspired by the *macro-JUTS* model developed for hybrid traffic model of the JUTS simulator [15].

In the MSBD macroscopic model, each traffic lane is divided into small segments (S) with length of Δx (see Fig. 1). In an ideal case, the length of all segments of all traffic lanes would be the same. However, the traffic network for the macroscopic model is based on the microscopic model's traffic network, which is being divided. So, the length of the lanes does not have to be divisible by the length of the segment Δx. Hence, there are some corrections to the Δx needed in each traffic lane depending on its actual length.

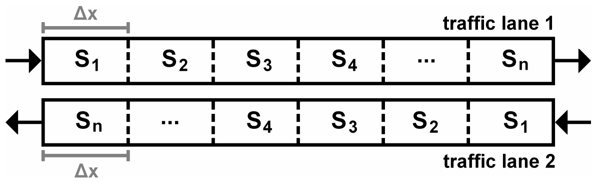

Fig. 1 Division of traffic lanes into segments

The traffic flow in the lanes is described by two parameters – the mean speed (v) and the vehicle density (ρ). These parameters are calculated for each segment in the traffic lane once per time step, which is Δt long. The values of the parameters of the i*th* segment are calculated from the parameters of the previous (i - 1)*th* segment (see Fig. 2).

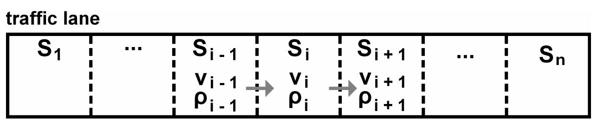

Fig. 2 Calculation of parameters in segments of a lane

The crossroad in the MSBD macroscopic model is constructed as a probabilistic division segment similar to [15]. There is no crossroad dynamic simulated. The crossroads consist of set of entries (E) and exits (X) connected to the incoming and outgoing traffic lanes, respectively. The values of the parameters for the exits are calculated based on the values of the entries, branching probability (p), and the ratio of the green period of the traffic lights cycle (g) if traffic lights are present and active. The scheme of the crossroad element is depicted in Fig. 3.

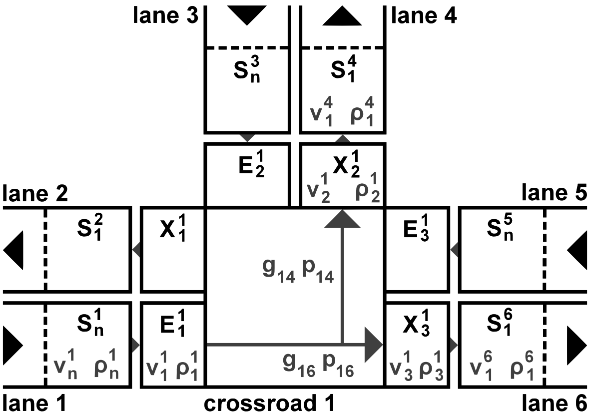

Fig. 3 The crossroad element

The parameters of traffic flows ingoing to the simulated area are generated using generators elements. These generators are connected to the lanes ingoing to the simulated area and can generate either stochastic traffic flow with exponential distribution or deterministic traffic flow with uniform distribution. Depending on the distribution, the whole simulation can be stochastic or deterministic, respectively.

4.4 Assigning weights to traffic lanes

Using the macroscopic simulation described in previous section, the particular traffic lanes can be assigned with weights. The weight of each traffic lane is calculated as the sum of vehicle densities in all segments of the lane during the simulation.

If the simulation is deterministic, one simulation run is sufficient to assign weights to the lanes, because all simulation runs give the same results. If the simulation is stochastic, several simulation runs are required in order to ensure fidelity of the weights. In that case, the weight of the i*th* traffic lane is calculated as an average of its weights from all simulation runs. Since the lane weight in one simulation run is calculated as sum of vehicle densities in segments of the lane, the total weight of the i*th* lane can be calculated as:

$$w_i = \frac{\sum_{k=1}^{N} w_{ik}}{N} = \frac{\sum_{k=1}^{N} \sum_{j=1}^{S_i} \rho_{ijk}}{N}, \quad (1)$$

where w_i is the total weight of the i*th* lane, w_{ik} is the weight of the i*th* lane from k*th* simulation run, N is the number of simulation runs, S_i is the number of segments of the i*th* lane, and ρ_{ijk} is the vehicle density in the j*th* segment of the i*th* lane in k*th* simulation run.

Although the stochastic simulation requires more simulation runs than the deterministic one, it is more similar to the microscopic simulation, for which the traffic network is divided, because the microscopic simulation is stochastic in nature. So, the stochastic macroscopic simulation seems to be able to give results more similar to the microscopic simulation than the deterministic macroscopic simulation. And since one macroscopic simulation run can be performed very quickly, it is possible to perform dozens of simulation runs in a suitable time.

Nevertheless, the fidelity of both the stochastic and deterministic macroscopic simulations in comparison to the microscopic can be sufficient for assigning of the weights to the traffic lanes. Hence both simulations will be tested (see Section 5.1).

4.5 Marking of traffic lanes

Once all traffic lanes are assigned with weights, it is possible to divide the traffic network into required number of sub-networks (i.e. mark the lanes that shall be divided). Using the weights of the traffic lanes, this can be accomplished easily is several steps.

First, it is necessary to calculate total weight of the entire traffic network as:

$$w = \sum_{i=1}^{L} w_i, \quad (2)$$

where w is the total weight of the network, L is the number of traffic lanes and w_i is the total weight of the i*th* lane. With the total weight of the network a weight per sub-network can be calculated as:

$$w_s = \frac{w}{M}, \quad (3)$$

where w_s is the weight per sub-network, w is total weight of the network, and M is the required number of sub-networks.

With known weight per sub-network, it is possible to proceed with second step, in which the modified breadth-first search [19] is employed. The entire network is considered as a weighted graph, where crossroads are the nodes and the sets of lanes connecting the particular neighbouring crossroads are the weighted edges (see Fig. 4).

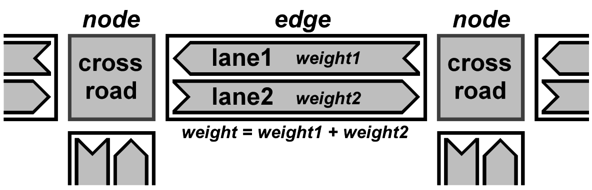

Fig. 4 The traffic network as a weighted graph

At the beginning of the breadth-first search algorithm, a crossroad is selected as the starting node and is assigned with the current sub-network ID (at the beginning, this ID is set to zero). As the breadth-first search is performed, more and more nodes (i.e. crossroads) are becoming explored. When a crossroad becomes explored, it is assigned with the ID of the current sub-network. Moreover, the weights of all lanes connecting this crossroad to its neighbours are added to the current weight of current sub-network. When the current weight of the sub-network reaches the weight per sub-network, the ID of the current sub-network is incremented and the current weight of the current sub-network is set to zero. This repeats, until all crossroads are explored. The pseudocode of the whole algorithm is depicted in Fig. 5.

```
currentID = 0;
sumPerSubnet = totalSum / subnetsCount;
currentSum = 0.0;
nodesToExplore = new List();
currentNode = nodes.pop();
nodesToExplore.push(currentNode);
while (!nodesToExplore.empty()) {
  currentNode = nodesToExplore.pop();
  neighbours = currentNode.neighbours();
  for (i, 0, neighbours.count()) {
    n = neigbours.get(i);
    if (n.node.state == WHITE) {
      n.node.state = GRAY;
      nodesToExplore.push(n.node);
      currentSum += n.weight;
    }
  }
  currentNode.state = BLACK;
  currentNode.subnetID = currentID;
  if (currentSum > sumPerSubnet;
    currentID++;
    currentSum = 0;
  }
}
```

Fig. 5 Algorithm of modified breadth-first search

Finally, in the third and last step, the traffic lanes, which shall be divided in order to form required number of traffic sub-networks, are being marked. Since the crossroads are assigned with the ID of traffic sub-network, in which they belong, the borders of the sub-networks can be found between each two neighbouring crossroads with different sub-network IDs. All lanes, which connect such pairs of crossroads, are marked as divided.

4.6 Marking issues and optimizations

There are two issues of the algorithm of marking of traffic lanes described in previous section.

First, it is possible that the resulting traffic sub-networks are not formed by connected graphs. This is implied by the switching of the sub-network ID during the performing of the breadth-first search. However, this is not a problem. Although such a division, when one or more sub-networks are formed by two or more unconnected parts, may seem odd to a human observer, it does not have to negatively influence the performance of the simulation.

However, it would be possible to employ a different algorithm of marking of traffic lanes to ensure division into sub-networks, which are formed by connected graphs (see Section 6).

The second issue is the selection of the starting crossroad for the breadth-first search algorithm. This selection can significantly influence the features of the resulting sub-networks including the connectivity of the particular sub-networks (see previous paragraph) and the number of divided traffic lanes. It was stated that the number of traffic lanes is not important using an efficient communication protocol for the distributed simulation (see Section 3). However, there is no reason not to minimize the number of lanes if it is possible.

Therefore, an optimization is added to the algorithm of marking of traffic lanes. The modified breadth-first search is not performed only once from one starting crossroad, but rather from all crossroads of the network. So, if there is for example one hundred crossroads in the network, the network division is performed one hundred times. This is not a problem however, because the algorithm is very fast.

The results of every division (i.e. the sub-networks IDs assigned to the crossroads and lanes marked to be divided) are stored. After the last attempt is finished, the division with minimal number of divided lanes is selected and the network is divided accordingly.

In most traffic networks, several divisions produce only sub-networks, where each sub-network is formed by one connected graph. If the connectivity of the particular sub-networks is required, the algorithm can find the best division with each particular sub-network formed by a connected graph if such a division exists.

5 Plan of testing

Now, as we discussed the whole algorithm of the MSBD approach, we can proceed with description of its testing. So far, only preliminary tests were finished. However, intensive testing will be performed soon according to the plan of the tests described in following sections.

5.1 Precision of macroscopic simulation

The first set of tests will be focused on precision of the macroscopic simulation used during the network division in comparison with the microscopic simulation, for which the network is divided. In other words, it will be determined, whether the results of both simulations are similar. This is an important presumption in order that the MSBD approach can divide the traffic network properly.

Preliminary tests were performed only for stochastic macroscopic simulation. Also, only one microscopic traffic model was used, although there are three models implemented in the DUTS system, which will be used for testing. The tests suggest that the difference between the vehicle density in the particular lanes ranges from 5 to 25 %. The mean difference then ranges from 10 to 20 %.

Nevertheless, extensive testing is required to determine the difference more precisely. The test will be performed for both stochastic and deterministic macroscopic simulation and for all three microscopic traffic models incorporated in the DUTS system. Also, the tests will be performed for several different traffic networks including regular grid of crossroads and a traffic network based on a real one (see Fig. 6).

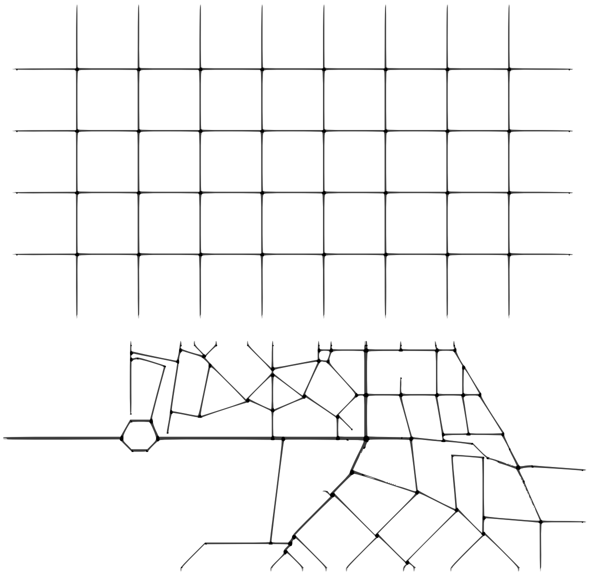

Fig. 6 Regular (up) and a real (down) traffic networks

5.2 Time performance of MSDB algorithm

The second set of tests will be focused on the time performance of the MSBD algorithm. The computation time necessary for the algorithm can be divided between two major steps – the macroscopic simulation (see Section 4.3) and the marking of traffic lanes (see Section 4.5).

The third remaining step – the assigning of the weights to traffic lanes (see Section 4.4) – is very little time-consuming, since it involves only one passing of the list of traffic lanes. On the other hand, the macroscopic simulation step can involve multiple simulation runs and the marking of traffic lanes step involves multiple breadth-first searches of the entire traffic network.

The preliminary tests were performed on a homogenous traffic network (regular grid of 256 crossroads) on a standard desktop computer (Intel Core 2 Duo 6700 2 × 2.66 GHz, 4 GB of RAM, Windows XP operating system). Ten macroscopic simulation runs and 256 breadth-first searches were performed during the traffic network division. Under these circumstances, the division required less then 15 seconds to be completed, and the vast majority of the computation time was consumed by the macroscopic simulation.

In order to investigate the performance of the MSBD algorithm more precisely, further tests will be performed. These tests will involve multiple traffic networks and multiple settings of the macroscopic simulation (e.g. deterministic or stochastic, number of simulation runs etc.).

5.3 Load of particular traffic sub-networks

The third and most important set of tests will be focused on the resulting performance of the distributed microscopic simulation. More precisely, the number of vehicles in particular sub-networks during the simulation run will be observed.

Although it would be more convenient to observe computation time of the particular sub-networks, it is difficult to measure it during the distributed simulation run. The reason is that each simulation process is synchronized with the others. Hence, all simulation processes are limited by the slowest process. For this reason, the numbers of vehicles will be used to determine the load of particular simulation processes. If the traffic network will be good divided (i.e. the load will be well-balanced between the processes), the numbers of vehicles moving in particular sub-networks should be similar.

The tests will be performed for various traffic networks, especially for regular grids of crossroads and networks based on real ones (see Fig. 6).

6 Future work

After the finishing of the testing of MSBD method, the aim of our further research will be the optimization of the method. We will focus mainly on various algorithms of marking of traffic lanes, which shall be divided. It is probable that a method can be found, which will provide even better balancing of the load of the simulation processes.

Another direction of our future research can be the adaptation of the MSBD method for a heterogeneous computing environment (i.e. cluster with computers with different computational power).

7 Conclusion

We presented a method of division of traffic network for distributed microscopic simulation of road traffic. This method – MSBD – is based on the assigning of the traffic lanes with weights using a macroscopic simulation of road traffic. The network is then divided into required number of sub-network with similar numbers of vehicles moving within them.

Although preliminary tests confirming the functionnality of the MSBD method were performed, a more complex testing is required to investigate the difference between the macroscopic and microscopic simulations and the resulting load of the particular simulation processes. Our further research will be focused on optimization of the method and its adaptation for different computing environments.

8 Acknowledgement

This work is supported by Ministry of Education, Youth, and Sport of Czech Republic – University spec. research – 1311.

9 References

[1] K. Nagel and M. Rickert. Parallel implementation of the TRANSIMS micro-simulation. *Parallel Computing*, 27(12):1611-1639, 2001.

[2] P. G. Gonnet. *A Queue-Based Distributed Traffic Micro-Simulation*. Technical report, 2001.

[3] J. Barcelo, J. F. Ferrer, D. Garcia, M. Florian, and E. Le Saux. The parallelization of AIMSUN2 microscopic simulator for ITS applications. *Proceedings of 3rd World Congress on Intelligent Transportation Systems*, Orlando, 1996.

[4] R. Klefstad, Y. Zhang, M. Lai, R. Jayakrishnan, and R. Lavanya. A Scalable, Synchronized, and Distributed Framework for Large-Scale Microscopic Traffic Simualation. *The 8th International IEEE Conference on Intelligent Transportation Systems*, Vienna, September 13 2005.

[5] D. Hartman. Leading Head Algorithm for Urban Traffic Model. *Proceedings of the 16th International European Simulation Symposium ESS*, Budapest, October 2004.

[6] T. Potuzak. Distributed and Centralized Version of an Efficient Communication Protocol for Distributed Traffic Simulation. *CSSim 2009*, Brno, September 7 2009.

[7] N. Cetin, A. Burri, and K. Nagel. A Large-Scale Agent-Based Traffic Microsimulation Based on Queue Model. *Proceedings of 3rd Swiss Transport Research Conference*, Monte Verita, March 19 2003.

[8] T. Potuzak. *Methods for Reduction of Inter-process Communication in Distributed Simulation of Road Traffic*. Doctoral thesis, University of West Bohemia, 2009.

[9] U. Klein, T. Schulze, S. Strassburger, and H. Menzler. Distributed Traffic Simulation Based on the High Level Architecture. *Proceedings of Simulation Interoperability Workshop*, Orlando, 1998.

[10] R. M. Fujimoto. *Parallel and Distributed Simulation Systems*. John Wiley & Sons, 2000.

[11] T. Kiesling and J. Lüthi. Towards Time-Parallel Road Traffic Simulation. *Proceedings of the Workshop on Principles of Advanced and Distributed Simulation (PADS'05)*, Monterey, 2005.

[12] T. Potuzak and P. Herout. Use of Distributed Traffic Simulation in the JUTS Project. *Proceedings of EUROCON 2007*, Warsaw, September 9 2007.

[13] T. Potuzak and P. Herout. An Efficient Communication Protocol for Distributed Traffic Simulation: Introduction of the Long Step Method. *Sofsem 2009: Theory and Practice of Computer Science, Proceedings, Volume II*, Špindlerův mlýn, January 24 2009.

[14] H. X. Liu, W. Ma, R. Jayakrishnan, and W. Recker. *Large-Scale Traffic Simulation Through Distributed Computing of Paramics*. California PATH Research Report, UCB-ITS-PRR-2004-42, 2004.

[15] D. Hartman and P. Herout. Construction of a Hybrid Traffic Model Based on JUTS Cellular Model. *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation*, Ljubljana, September 9 2007.

[16] M. H. Lighill and G. B. Whitman. On kinematic waves II: A theory of traffic flow on long crowed roads. *Proceedings of the Royal Society of London*, s. A, 229, London, 1955.

[17] C. Braban-Ledoux. *METACOR – A Macroscopic Modelling Tool for Corridor Application to the Stockholm Test Site*. Final Report CRT2000:05, Center for Traffic Engineering & Traffic Simulation, Kungl Tekniska Högskolan, Valhallavägen, 2000.

[18] T. Potuzak. Current Trends in Distributed Traffic Simulation. *Proceedings of 41th Spring International Conference MOSIS '07 – Modelling and Simulation of Systems*, Rožnov pod Radhoštěm, April 24 2007.

[19] D. E Knuth. *The Art of Computer Programming Vol. 1. 3rd edition*, Addison-Wesley, 1997.