Distributed Computing Approach to Optimize Road Traffic Simulation

Amritanshu Sinha, Tapas Saini, S.V. Srikanth
Center for Development of Advanced Computing
Email: {amritanshus, stapas, venkatas}@cdac.in

***Abstract*—Distributed computing is the method of running CPU intensive computations on multiple computers collectively in order to achieve a common objective. Common problems that can be solved on the distributed systems include climate/weather modeling, earthquake simulation, evolutionary computing problems and so on. These type of problems may involve billions or even trillions of computations. A single computer is not capable to finish these computations in short span of time, which is typically in days. Distributed computation helps to solve these problems in hours, which could take weeks to solve on a single computer. Distributed computing generally uses the existing resources of the organization.**

Traffic simulation is the process of simulating transportation systems through software on a virtual road network. Traffic simulation helps in analyzing city traffic at different time intervals of a single day. Common use cases could be analyzing city wide traffic, estimating traffic demand at a particular traffic junction and so on. This paper discusses about the approach to use distributed computing paradigm for optimizing the traffic simulations. Optimizing simulations involves running a number of traffic simulations followed by estimating the nearness of that simulation to the real available traffic data. This real data could be obtained by either manual counting at traffic junctions, or using the probes such as loop inductors, CCTV cameras etc. This distributed computing based approach works to find the best traffic simulation corresponding to the real data in hand, using evolutionary computing technique.

***Keywords*—Distributed computing, Traffic simulation, Evolutionary algorithm, Root Mean Square Error, SUMO.**

I. Introduction

Computers play a major role in our lives today. They are widely used in almost every field of engineering such as mechanical engineering, civil engineering, chemical engineering and so on. Computers have become almost a need for every engineering field and are also playing a significant role in the field medical science. Fast advancements of computer has opened the new possibilities of problems solving, thus taking part in solving complex problems in every professional field. The constantly increasing processor speed and varied memory expansions is allowing to solve large computational problems in less time. Adding to it, the seamless network connectivity between computer systems is also contributing to distributed processing of big problems, thus multiplying the speed of computation. These distributed systems are group of networked computers which have a common goal to solve a large computational problem.

Road traffic problem is ubiquitous these days. Almost every major city in the world is facing the problem of traffic congestion. Traffic congestion creates more problems like air pollution, noise pollution and time wastage in commuting. For better traffic management, traffic congestion should be analyzed and proper measures should be taken as a remedy. Understanding and solving traffic congestion may involve creating and running virtual traffic model on a computer. In order to visualize the traffic behavior dynamically, this traffic congestion can be analyzed by simulating the road traffic model on computer and generating results in the form of traffic flows. Various traffic simulators are available in order to generate virtual traffic on a digital map of the city. SUMO [1], MATSim [2] and DTALite [3] are the examples of free traffic simulators commonly used these days. These simulators require a road network of a city, which could be created on an area of interest. Every single simulation is mainly CPU bound process so it requires dedicated CPU. In this paper, simulation process involves running a test simulation and further comparing simulation results with real observed results. To study heavy traffic, any simulator has to infuse large number of vehicles for analysis. The major problem encountered to simulate this heavy traffic, is the increase in demand of large computing and storage resources from computer. For multiple simulations of whole city the resource requirements multiply, which couldn't be satisfied by a single computer. For analysis through any evolutionary methodology thousands of such simulation has to be run. Distributed computing based methodology addresses these type of the problems.

This paper discusses about the work done to find the best traffic simulation which models the real traffic of the city. This is done by running and filtering the best simulations on a LAN based Distributed system. The best simulation model for the city was achieved after running the algorithm for days on the LAN.

II. Related work

A lot of research has been done on road traffic related areas along with utilizing the distributed computing. Simon [4] has presented that how distributed computing and grid computing approaches can be applied to Modeling and Simulation(M&S). This paper showed how Volunteer computing, a grid computing approach can be used to speed up simulation on bigger models and benefit M&S researchers and practitioners. Tomas [5] has introduced Distributed Urban Traffic Simulator (DUTS), a microscopic parallel distributed version of simulator of road traffic. This simulator was run using cluster of nodes with multicore processors. This paper compared two types of simulation, distributed and parallel/distributed, and proved that parallel/distributed version was efficient. It utilized multithreaded simulation process in order to speed up simulations. Lifeng Xiao [6], proposed that various Transfer Subroutine Module (TSMs) and a distributed computing model can be build for simulation of the product design process. Yijia Xu [7], proposed a distributed programming architecture for operation research studies that evolve optimization model and discrete event simulation model. The architecture is a multi-layer system, which is composed of a middleware layer, the component management layer, component layer and modeling layer from the bottom up. Dr. Christoph [8], presented a combinational approach of material flow simulation and meta-heuristic for an innovative, fast converging procedure for the optimization of simulation models parameters. Based on a specific test setting, Particle Swarm Optimization, Genetic Algorithm as well as their combination are used as an automatic experiment design in a distributed simulation environment. N.Mustafee [9] discussed the readily available grid computational and storage power to the user, making the grid by using desktop PC in the LAN environment. He focused on speeding up the simulation process. P.Fernandes [10], discussed about an approach to increase road capacity by making vehicles move much closer to each other. He explained the studies of models to allow Inter-vehicle communication (IVC), with platooning capabilities. In his research he encompasses the expansion of SUMO features, to allow studies on cooperative behaviors of communication-enabled autonomous vehicles. J.G.Kim [11], presented grid-enabled approach for modeling a three-dimensional, multiphase fluid flow in the grid computational environment. J.Anda [12] proposed Vgrid, an ad-hoc networking and computing grid formed by leveraging inter vehicle and vehicle to roadside wireless communication. His goal was to evolve intelligent transportation system from a centralized to a distributed approach in which vehicle equipped with wireless networking and computer can cooperate and solve vehicular traffic-flow control problem autonomously. He has given the case study of merging the two lanes. Jiankun Wu [13] had presented a simulation architecture named GHA, which implemented the HLA's [14] component as grid service and combined with the agent technology. He implemented a traffic simulation platform based on GHA. Alfired Park [15], has worked on optimistic parallel simulation over public resource computing Infrastructures and Desktop Grids. He used master worker (M/W) approach for these platforms. Lishao Wang [16] presented P2P Grid based Traffic Simulation model (PGTS) which is a grid-based traffic micro simulation model to solve the traffic flow micro-simulation problem for large scale transportation network.

III. TRAFFIC SIMULATION

Traffic simulation of transportation system is the mathematical modeling of the transportation system through the application of computer software. It can be applied to transportation planning, design and operations. Transportation mod-

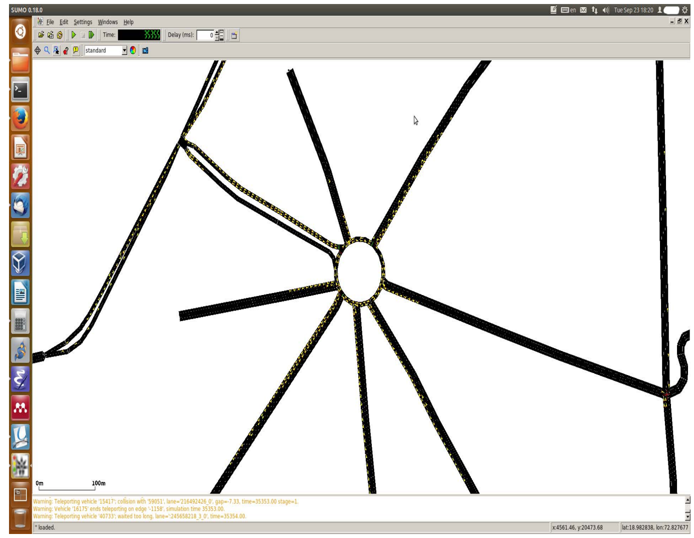

Fig. 1. Traffic Junction: SAAT RASTA, Mumbai

els generally can be classified into microscopic, mesoscopic, macroscopic, and metascopic models. This research has used "Simulation of Urban Mobility" (SUMO) for the simulation purposes. SUMO is microscopic, multi-modal traffic simulator in which each vehicle can be explicitly modeled in the given network of the city. Basic input for the simulation is road/rail network and Origin-Destination (OD) matrix. The OD matrix can be obtained from various sources like traffic authorities or it can be generated from mobility of cellphones. OD matrix are based on flows between the transportation zones representing the number of the people who moves from one zone to another zone at a specific time period. Simulator first decomposes the OD matrix to the single vehicle trip. Once the trips are generated, the microscopic simulation can run, refer Fig. 2. Different utilities are present in SUMO software for various purposes. SUMO simulation software provides *sumo* and *sumo-gui* applications that run the simulation. *sumo* is a command line application for running traffic simulations. *sumo-gui* is a graphical representation of *sumo* which allows to visualize the movement of every single vehicle on the street. Each vehicle is assigned a unique vehicle id. It allows to monitor number of vehicles crossed through any particular junction and also traffic density of the road, refer Fig. 1. *duarouter* is another important utility provided in SUMO package, that computes fastest routes through the network by importing different types of demand description. Through *duarouter*, every vehicle's route can be pre-determined. Among all other types of simulations, microscopic simulation is a time consuming process. For heavy traffic, the simulator takes time to load all the vehicle on the road network. Depending upon the routing algorithm used in the simulator, traffic flows through the network accordingly. Input of the traffic signal timings in simulation are also required in the network that should match with the actual signal timings.

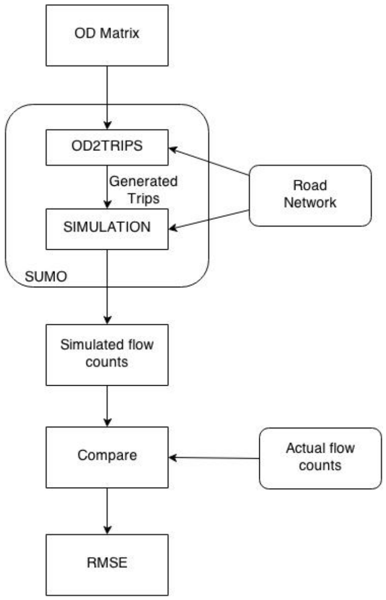

Fig. 2. Traffic Simulation Flow Diagram

A. Simulation Scenario

MUMBAI city, the fifth most populous city in the world having estimated population of 18.4 million has been chosen for simulation purpose. Fig. 3 shows road network of Mumbai city for simulation. OD matrix of the whole mumbai was generated from Call Detail Records(CDR) data with focusing more on south Mumbai, as major commercial activities happen in south Mumbai. OD Matrix was generated for morning peek hours(9.30 to 11.00 AM). All major roads and junctions of the south mumbai were considered for road network creation. A microscopic simulation for 90 minutes was run on this road network. A typical traffic simulation on mumbai city traffic when ran to simulate for 90 minutes traffic on a dual core computer with 4 GB RAM, took more than 20 minutes. Post simulation, every simulation output in the form of flows of traffic was validated by some real data. Real traffic counts were collected at major junctions, from where bigger chunk of traffic moves between different parts of city. The data collection for real traffic counts, was done through manual traffic surveys. Although the data could also be collected by other means like getting sensor values from road side probe devices, loop inductors embedded at key traffic junctions and so on. The data collected by these means would have required extra work to fetch from traffic authorities, calibrate and convert into the counts. Doing manual surveys and counting through pen and paper method had advantage of getting direct counts of the vehicles. Finally, these manual traffic counts acted as input to the simulation validating logic. These traffic counts helped in finding the best simulation depicting the real dynamic traffic behavior. Thousands of random ODs were generated to run as simulation and the whole process repeats until desired optimization is achieved from EA (evolutionary algorithm) [17].

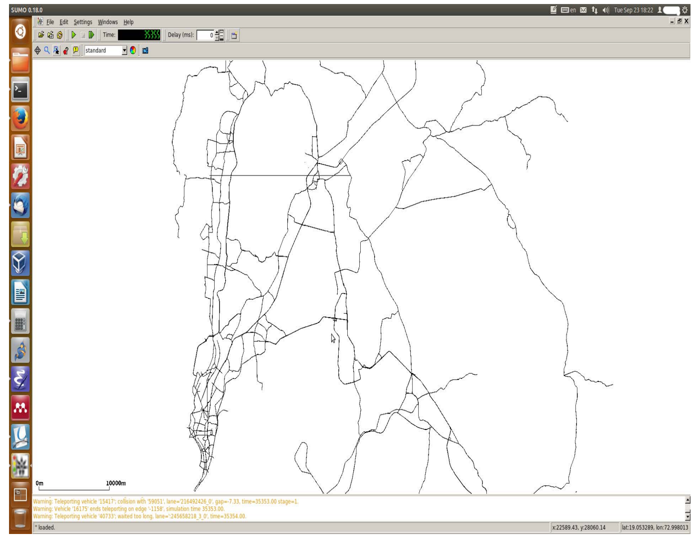

Fig. 3. Mumbai simulation scenario

IV. Distributed Computing

According to Wikipedia, Distributed Computing is a field of computer science that studies distributed systems. A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. A salient feature of distributed computing is the resources are connected to one another through seamless network connectivity. A number of heterogeneous resources can also be used for distributed computing. All the computers connected, can take part in problem solving collectively with a common objective. This allows in breaking down a big computational problem into smaller problems and solve in distributed fashion. Traffic simulation is one such application which requires large number of computing resources to simulate traffic.

Distributed computing inside LAN environment has been used in the work. A group of n computers can be used to construct a distributed system. All the computers should be connected to LAN through Ethernet interface. Any distinguished machine can act as master node. The same machine may or may not be given a role of client, depending upon user's preference. Rest of the n-1 machines act as workers. As a basic requirement of distributed computing, a large computational problem has to be broken down into smaller pieces by client and distributed to the worker nodes. The master node acts as a middle ware between the client and workers. Technically, whole problem is divided into a number of requests. Each worker processes a single request at a time. This result in n-1 workers solving at least n-1 requests parallely. Each single request is processed by a computer which involves, running a number of traffic simulations. After end of every simulation, the simulation results are compared with the actual traffic counts available, and their error is recorded as result. At the end, all results are returned to the client. Client processes these results, and creates a new set of requests to be processed again by workers. Fig. 4 shows block diagram of the distributed system. This whole process repeats a number of times. An evolutionary algorithm technique uses these simulation results to optimize results to best simulation. In the next section we have discussed the methodology to achieve the simulation accuracy.

A. Evolutionary Algorithm

Evolutionary algorithm(EA) methodology is often used to solve complex problems, using mechanisms similar to biological evolution. Genetic algorithm is one popular evolutionary algorithm technique. It mimics the process of natural selection. Like natural selection methodology, genetic algorithm works on chromosomes, a form of candidate solutions for the problem. Genetic algorithm runs for generations which is a form of iteration, where each generation involves crossover, mutation, selection and fitness evaluation on a population of chromosomes. Crossover is a technique which occurs between any two chromosomes, where their contents gets exchanged partially. Mutation technique involves randomly changing some characteristics of chromosomes. Selection as name explains, is done to chose chromosomes for next generation from existing ones. Fitness evaluation is finding nearness of chromosomes to desired outcome in the form of Root Mean Square Error(RMSE) explained in next section. In order to achieve the desired accuracy level, hundreds or sometimes even thousands of such generations need to be run. At first generation, one seed chromosome and rest random chromosomes are evaluated. Evaluation involves running simulation of every problem/chromosome instance and finding out its nearness to the actual desired outcome. Once all the problems are estimated, the chromosomes with better fitness are graduated for next generation. Before the start of next generation, crossover followed by mutation takes place among better candidates. This increases the probability of getting a new better candidate from existing better ones. This complete cycle of crossover, mutation, selection and fitness evaluation keeps running for a significant number of generations until the desired nearness/fitness is achieved.

B. RMSE

Root Mean Square Error(RMSE) was used to find out the nearness/fitness of traffic simulation. It is used to represent standard deviation of the differences between simulated and actual values as shown in Equation (1)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{q}_i - q_i)^2}{n}} \tag{1}$$

where, $\hat{q}_i$ and q_i are the simulated and actual traffic values respectively. In simple words it is used to measure the differences between values predicted by a model and values actually observed.

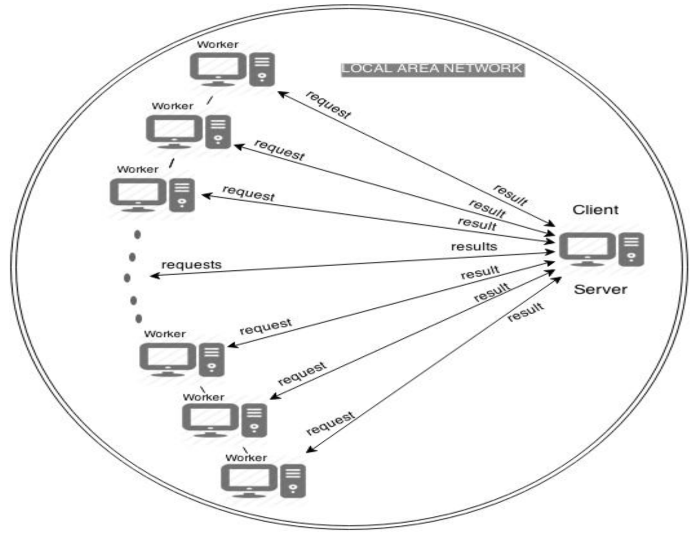

Fig. 4. Distributed System Block Diagram

V. System Configuration

A set of workstations with same configurations was used. A total of 19 workstations were used for setting up LAN system. One distinguished workstation was given the role of both Master and client. The client creates a set of possible candidate solutions and splits it into requests. All the remaining workstations were given role as workers. Master was acting as a middleware between client and worker. Every worker had the job of taking the request and process it. Each request had one or more chromosome to process. The number of chromosomes in each request depended on the number of workers. Specifically, the number of chromosomes in a request was inversely proportional to number of workers. Each chromosome was represented by an OD matrix. Each chromosome was converted to a trip file, and a simulation used to run using that trip file. These days almost every workstation is equipped with multicore CPUs. Exploiting this feature, all the available CPUs in the workstation were used for simulation. As an example, if a workstation had dual core CPU, then 2 simulations were running at a time. Also if a workstation had 8 core CPU, then total of 8 simulations were running parallely at any instant. As previously told, all the workstations(workers) were of same configuration, the number of parallel simulations for every workstation were nearly equal. This helped in utilizing every small to big resource in the LAN. All SUMO simulation models of mumbai traffic were run and ended with the traffic counts at all the junctions. These were simulated values as shown in equation (1) which were compared with actual flow values. A single simulation took on an average 20 to 25 minutes on a typical workstation. This shows distributed system helped in parallelizing the simulation runs.

VI. Results

Out of total 19 deployed, 1 workstation ran as a client and server. Rest 18 played the role of worker. Each worker had octa

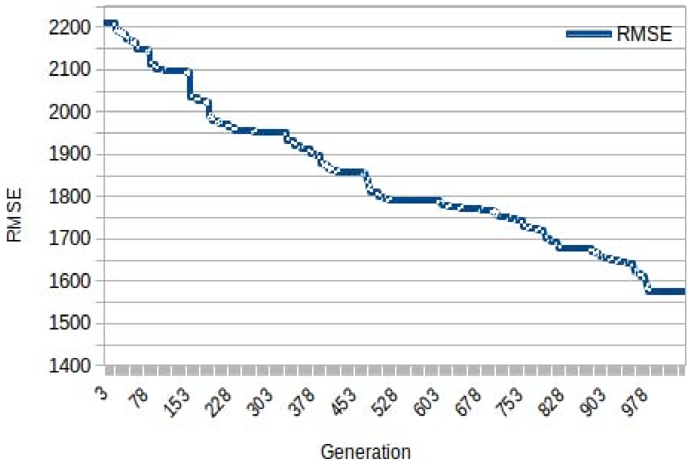

Fig. 5. RMSE optimization using EA run 1

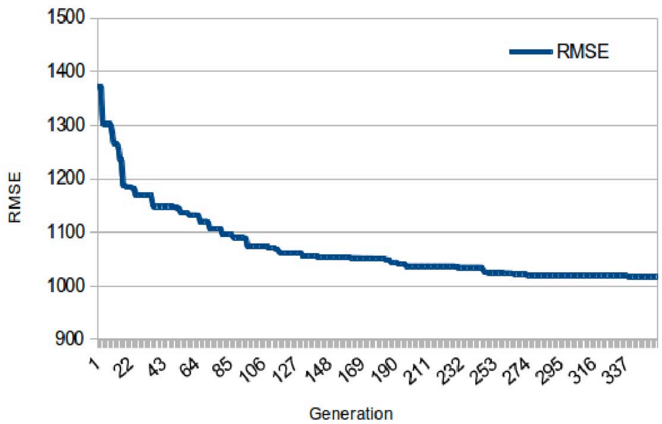

Fig. 6. RMSE optimization using EA run 2

core CPU(8 cores). A population of 144 chromosomes was taken for every generation. This calculates to 8 chromosomes per worker node. Due to running parallel simulation on this system all 144 simulations were able to run in average 20 minutes time. This way 72 generations were able to run in 24 hours. A total count of 10368 simulations were able to run in 24 hours. Multiple runs were executed by using evolutionary algorithm methodology, out of which we have displayed result of two important EA runs in Fig. 5 and Fig. 6. These two figures shows improvements in error rate in the form of downward trend for RMSE. Fig. 5 are results after merging multiple EA runs, which shows the RMSE improving from 2200 towards 1500 in total 1049 generations. Similarly, Fig. 6 shows the reduction of RMSE from 1300 level to 1000 in 356 generations. Further EA runs can be executed to bring RMSE further down.

VII. Conclusion

Distributed computing is an efficient methodology for problems which require massive calculations. Traffic simulation optimization problem is chosen to understand city traffic demand. Distributed computing helped in archiving the desired traffic simulation scenario corresponding to mumbai traffic flows. Future work will involve setting up and using grid computing mechanisms for simulation optimization of further new cities.

Acknowledgment

The authors would like to thank Centre for Development of Advanced Computing (C-DAC) and Department of Electronics & Information Technology (DeitY) for providing resources in carrying out this research. Authors would also thank ITS Planners and Engineers Pvt. Ltd (ITSPE) for helping to accomplish this research.

References

[1] SUMO, "Simulation of Urban MObility," http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/.

[2] Matsim, "Multi agent transport simulation," http://www.matsim.org/.

[3] DTALite, "dynamic traffic assignment model," https://sites.google.com/site/dtalite/.

[4] S. J. E. Taylor, M. Ghorbani, T. Kiss, D. Farkas, N. Mustafee, S. Kite, S. J. Turner, and S. Straburger, "DISTRIBUTED COMPUTING AND MODELING & SIMULATION:SPEEDING UP SIMULATIONS AND CREATING LARGE MODELS," *Proceedings of the 2011 Winter Simulation Conference*, pp. 161–175, 2011.

[5] T. Potuzak, "Distributed-Parallel Road Traffic Simulator for Clusters of Multi-core Computers," *16th International Symposium on Distributed Simulation and Real Time Applications*, pp. 195–201, 2012.

[6] L. Xiao and Y. Zhang, "Design of Distributed Computing Model in Multidisciplinary Collaborative Simulation Environment," in *Computer Science and Information Engineering, 2009 WRI World Congress on*, 2009.

[7] Y. Xu and S. Sen, "A distributed computing architecture for simulation and optimization," in *Simulation Conference, 2005 Proceedings of the Winter*, 2005.

[8] D. C. Laroque and A. Klaas, "Fast converging, automated experiment runs for material flow simulations using distributed computing and combined metaheuristics ," in *Simulation Conference (WSC), Proceedings of the 2012 Winter*, 2012.

[9] N. Mustafee, A. Alstad, B. Larsen, and S. J. E. Taylor, "Grid-enabling FIRST: Speeding Up Simulation Applications Using WinGrid," in *Distributed Simulation and Real-Time Applications, 2006. DS-RT'06. Tenth IEEE International Symposium on*, 2006.

[10] P. Fernandes and U. Nunes, "Platooning of Autonomous Vehicles with Intervehicle Communications in SUMO Traffic Simulator," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 2010.

[11] J. G. Kim, K. W. Cho, and H. W. Park, "Toward a Grid-Based Simulation of Multiphase Fluid Flow in Porous Media," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, 2004.

[12] J. Anda, J. LeBrun, D. Ghosal, C.-N. Chuah, and M. Zhang, "VGrid: Vehicular AdHoc Networking and Computing Grid for Intelligent Traffic Control," in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 2005.

[13] J. Wu, L. Huang, J. Cao, M. Li, and X. Wang, "Research on Grid-Based Traffic Simulation Platform," in *Services Computing, 2006. APSCC '06. IEEE Asia-Pacific Conference on*, 2006.

[14] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Frame and Rules [S], IEEE Std," Sep.2000, pp. 1516–2000.

[15] A. Park and R. Fujimoto, "Optimistic Parallel Simulation over Public Resource-Computing Infrastructures and Desktop Grids," in *Distributed Simulation and Real-Time Applications, 2008. DS-RT 2008. 12th IEEE/ACM International Symposium on*, 2008.

[16] L. Wang, B. Mao, and S. Chen, "A P2P Computational Grid-Based Parallel Traffic Micro-Simulation Model for Large Scale Transportation Networks," in *Distributed Simulation and Real-Time Applications, 2008. DS-RT 2008. 12th IEEE/ACM International Symposium on*, 2008.

[17] EA, "Evolutionary algorithm," http://en.wikipedia.org/wiki/Evolutionary_algorithm.