# Parallel Simulation Algorithm for Traffic Systems

M. Namekawa[a], A. Satoh[b], H. Mori[c], N. Honda[d] and H. Suda[e]

[a] *College of Management and Economics, Kaetsu University, Tokyo, Japan (namekawa@kaetsu.ac.jp)*
[b] *Faculty of Engineering, Toyo University, Saitama, Japan (akira@eng.toyo.ac.jp)*
[c] *Faculty of Engineering, Toyo University, Saitama, Japan (mori@mo.cs.toyo.ac.jp)*
[d] *Faculty of Electro-Communications, The University of Electro-Communications, Tokyo, Japan (honda@fs.cas.uec.ac.jp)*
[e] *Faculty of Business and Commerce, Keio University, Tokyo, Japan (ssuda@hc.cc.keio.ac.jp)*

**Abstract:** In recent years, the parallel processing as well as distribution processing has been in wide use in many fields. Various parallelisms are attempted for a purpose of reducing execution time. The parallel simulation technique has attracted researchers' attentions recently as a method which enables to perform simulation in a system which requires the execution of simulation in a large-scale and complex system which has been considered impossible and which improves the processing efficiency of the simulation and shortens the execution time.The parallel simulation extracts the parallel property at the time of executing the simulation and improves the simulation-processing efficiency by distributing the workload among the multiple processors. However, almost all the conventional parallel simulation is based on discrete-events and a continuous-event parallel simulation is hardly in use. This is because the parallel simulation for the continuous-event requires, in principle, the synchronization processing per simulation clock so that the processing efficiency due to the parallel processing implemented is deteriorated to a greater degree.We classify simulation modes into two systems. One is classified as a spatial flow-type system which is expressed by a model which reproduces the situation where the fluid flows in the three-dimensional space. Another is classified as a transaction flow-type system in which a certain kind of transaction moves in the space. Traffic systems are classified into the transaction flow type system. We propose a new synchronization algorithm which realizes a desirably efficient simulation, because a certain synchronization is omitted by means of transport prediction of the transaction in the transaction flow-type system. It can simulate almost all general social system.In this paper, we describe the parallel simulation algorithm for traffic system to improve the processing efficiency of each processor. We also carried out experimentation there of and presents its verification.

*Keywords:* Parallel simulation; Parallel/distributed algorithms; Continuous-changed model; Traffic system

## 1. INTRODUCTION

The parallel simulation technique has attracted researchers' attentions recently as a method which enables to perform simulation in a system which requires the execution of simulation in a large-scale and complex system, and which improves the processing efficiency of the simulation and shortens the execution time.

The parallel simulation extracts the parallel property at the time of executing the simulation and improves the simulation-processing efficiency by distributing the workload among the multiple processors. However, almost all the conventional parallel simulation is based on discrete events [Ayani and Rajaei, 1992; Galluscio and Douglass, 1995; Konas and Yew, 1995; Mabry and Gaudiot, 1994; Misra, 1986; Takai, 1996], and a continuous-event parallel simulation is hardly in use. This is because the parallel simulation for the continuous-event requires, in principle, the synchronization processing per simulation clock so that the processing efficiency due to the parallel processing implemented is deteriorated to a greater degree.

We classify simulation modes into two systems. One is classified as a spatial flow-type system which is expressed by a model which reproduces the situation where the fluid flows in the three-dimensional space. Another is classified as a transaction flow type system in which a certain kind of transaction moves in the space. In this paper, we propose a new synchronization algorithm which realizes a desirably efficient simulation, because a certain synchronization is omitted by means of transport prediction of the transaction in the transaction flow type system [Namekawa et al., 1995; Namekawa et al., 1999]. It can simulate almost all general social system.

Then, we will justify our algorithm and verify its effectiveness.

We describes parallel simulation method at section 2, synchronization algorithm for road traffic system at section 3, evaluation method for it at section 4, evaluation of simulations at section 5 and finally concluding remarks at section 6.

## 2. PARALLEL SIMULATION METHODS

In general, in the parallel simulation, there exist two methods. In one method, a specific function in common is made separate and independent so as to parallel-process the function. In another method, a system as an object for the simulation is divided spatially, so that the process is divided for each divided sub-space.

(a) Function distributed method.

(b) Space distributed method.

The former method is defined to be a function distributed method while the latter is defined to be a space distributed method.

In the parallel simulation, necessitated is the process which avoids the logical conflicts and which correctly holds the order of simulation clock between processors where the workload is distributed among them. As a result, its overhead time becomes obstacle in terms of process efficiency.

The overhead time in the function distributed method depends on the number of access to the functions in common as well as on its processing. The overhead time in the space distributed method depends on neighboring sub-space as well as on the transfer of transaction. Vehicles in a road traffic system are expressed as transactions. It is to be noted that as far as the processors have the identical function and performance in the parallel processing, the processing time for simulation alone is invariant in the space distributed method.

Therefore, in the parallel simulation according to the space distributed method, it is necessary to add the preparation time, communication time, waiting time and so on for consistency processing as well as synchronization processing, of the simulation time.

## 3. SYNCHRONIZATION ALGORITHM FOR TRAFFIC SYSTEM

In this section, a new synchronization algorithm of the parallel simulation for traffic systems will be described.

### 3.1 Traffic System and Distribution Method

Simulation, in which vehicles are expressed as microscopic models, such as a road traffic system simulation is considered to be a typical continuous-event simulation [Nakanishi et al., 1995; Satoh et al., 1995; Yikai et al., 1997]. A characteristic of the continuous-event simulation is that the change of the system is mainly dependent on continuous change (that is, continuous movement of a vehicle). Thus, the movement of a vehicle is simulated in a manner which is described as a continuous-change model.

The road traffic system is considered as one of the space distributed method, which we call it here a local distribution method so as to emphasize its two-dimensionality.

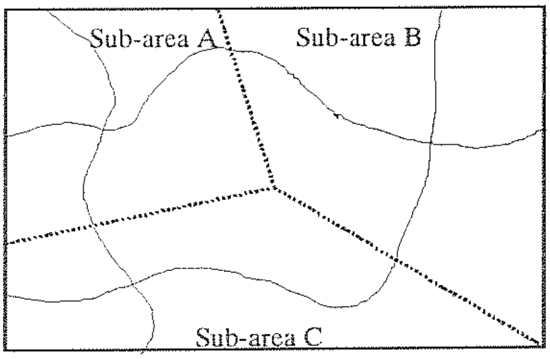


**Figure 1.** Divided area.

In the parallel simulation implementing this local distribution method, a sub-area to be simulated is partitioned into disjoint sub-areas as shown in Figure1. A processor is assigned for each divided

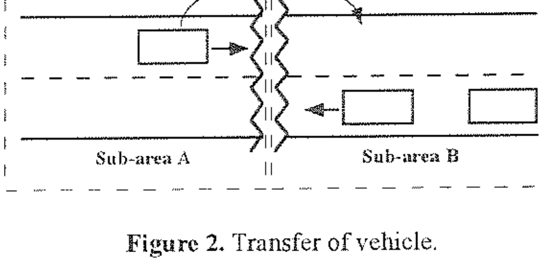


**Figure 2.** Transfer of vehicle.

sub-area, so that simulation corresponding to each sub-area can be carried out. Thus, a vehicle which moves through the simulation area moves from one sub-area to another, that is, from one processor to another as shown Figure 2. We define this movement as 'transfer'. However, it is to be noted that these transfers do not necessarily occur per simulation clock though its occurrence depends on the density of traffic. In other words, it is found that the transfer of vehicle between sub-areas is of a discrete event.

As for a continuous-event model, the parallel simulation synchronization itself is expressed as a discrete-like event. We are trying to come up with a parallel simulation method by which the transfer of a vehicle between processors gives out the same result as if it is obtained by a simulation performed by a single processor.

### 3.2 Transfer Forecast for Road Traffic System

In order that the transfer of a vehicle between sub-areas is performed consistently at a same timing, it is indispensable that the time of the transfer be accurately known in advance. However, since the transfer is generally is not deterministic, it is impossible to carry out the transfer forecast in a smooth and efficient manner. Hence, we attempted to achieve such a smooth and efficient transfer forecast by performing the transfer forecast as precisely as possible.

#### 3.2.1 Transfer forecast

The forecast which turns out to be earlier than the actual transfer time results in the increase of the execution time. However, since the forecast is not behind the actual transfer, consistent synchronization process is guaranteed.

The transfer from one sub-area to another can be roughly predicted by taking the current location, transfer direction and transfer speed into consideration so that the arriving time and destined sub-area can be known. Since the vehicle transfer changes in a random manner, it is necessary to carry out the transfer forecast based on each term such as the current location, transfer direction and transfer speed.

In order that this transfer is carried out smoothly, the simulation clocks between two neighbouring sub-areas must be identical. Therefore, it is indispensable that the transfer forecast is ahead of the actual transfer as shown in Figure 3. If the forecasted is behind the actual transfer, this method is no longer valid.

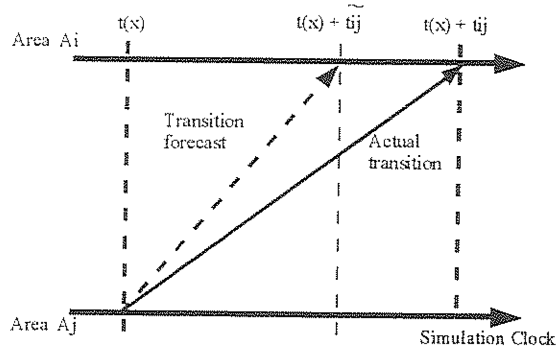


**Figure 3.** Predicted as well as actual transfer.

#### 3.2.2 Repetition of the forecast

If the first forecast does not match with the actual one, several more predictions can lead to a smooth transfer as shown in Figure 4. If the synchronization as depicted in Figure 4 is carried out, the simulation can be performed independently at each sub-area alone without other sub-areas concerned, up to the synchronization time. As a result, the synchronization process at each simulation clock is omitted. Thus, the communication time is decreased and thereby the total of simulation execution time is also reduced.

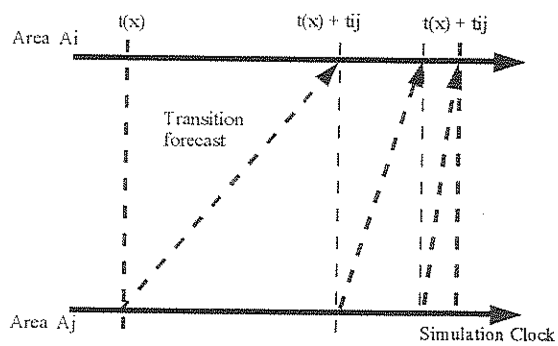


**Figure 4.** Repetition of the event forecast.

## 4. EVALUATION METHOD FOR SYNCHRONIZATION ALGORITHM

In order to verify our proposed algorithm so as to evaluate it, two phases of simulation systems through various steps are developed and each simulation is carried out as follows.

### 4.1 Specific Simulation

One is a specific simulation for the road-traffic system by means of the microscopic model in which a certain sub-area is simulated. Performed are the single-unit simulation by a single processor and the simulation by two processors where the workload is distributed and parallel-processed (we call them continuous-event parallel simulations). These two types of simulations are compared so that each algorithm is verified.

### 4.2 Evaluation Simulation

Another is a simulation for confirming and evaluating the effectiveness of the algorithm for its parallelism (we call it evaluation simulation). The structural data on the overhead time in the parallel simulation and various data required for executing road-traffic simulation with the micro-model are obtained from the continuous-event parallel simulation described at section 4.1. Then, the evaluation simulation with multiple processors employed is performed virtually on a single processor. In this simulation, the effect of the parallelism can easily be evaluated by varying parameters. For example, the number of processor is specified by a parameter, so that the distributed parallel environment corresponding to the number of processor is automatically adapted and the simulation to be evaluated is executed.

### 4.3 The Transaction Vehicle Distribution

The parallel simulation is performed with a certain area being distributed. When the transactions are evenly distributed among each distributed processor, the effect on the parallelism will be of a greater extent. However, in general, the distribution of the transaction is not well balanced since the movement of transaction occurs at random. Thus, in order to confirm generality of the proposed algorithm, the simulation is performed on two kinds of cases; one is a case where the transaction is almost evenly distributed and another is a case where the transaction is not evenly distributed.

Suppose there is a circular road whose circumference is 60 km long. Then, the road is uniformly divided and parallel-processed. The vehicle's running speed is set to 40 plus or minus 5 km/hour in a uniform manner as an attributed value for the vehicle. The vehicle is generated in a uniformly distributed manner and runs on the circular road in the case where the transaction is almost evenly distributed. On the other hand, an environment in which the transaction to be processed contains twice as much as the not-evenly-distributed transaction in a specific area is set up in the case where the transaction is not evenly distributed.

### 4.4 Evaluation Items

The rate of speedup improvement in executing the simulation is a general method in evaluating the parallelism effect. In our study, as a means to confirm the effect of the algorithm, the number of communication responsible for the overhead time as well as the waiting time for the synchronization in addition to the rate of speedup improvement are analyzed and evaluated.

## 5. EVALUATION OF SIMULATIONS

We evaluate the simulation results to verify our synchronization algorithm as follows;

### 5.1 Evaluation of New Algorithm Against Full Synchronization Algorithm Type

Figure 5 shows a result of a case where the workload is almost evenly distributed. In order to verify the effectiveness of the our algorithm, the result obtained is compared to an algorithm which is always synchronized at every simulation clock. We call it a full synchronization algorithm type. As the number of parallelism (the number of processor) increases, the effectiveness is improved. For example, the

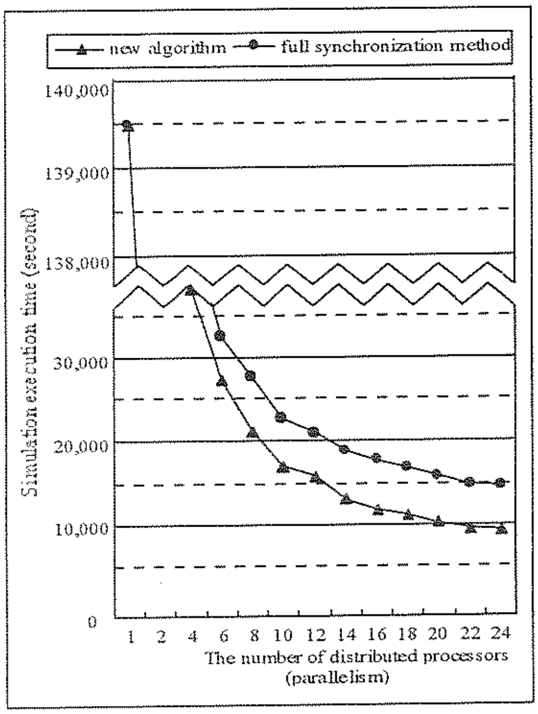


**Figure 5.** Simulation results (almost evenly distributed).

execution time for the simulation is 53.3 % as the number of processor is increased from one to two. That is, the effectiveness is almost directly proportional to the number of processor.

On the other hand, the execution time for the simulation of the full synchronization algorithm type is 56.4% when the number of processor is increased from one to two as well as from two to four. That is, the our algorithm is more effective compared to the full synchronization algorithm type. This tendency becomes further notable as the number of processor increases.

### 5.2 Improvement Rate

To see the effectiveness of the new algorithm, Figure 6 indicates the rate of improvement of the new algorithm against the full synchronization algorithm type. Referring to Figure 6, we can see that the rate of improvement improves as the number of processor increases.

### 5.3 Evaluation for Transaction Distribution

Figure 7 shows a result of the simulation in which the transaction is not evenly distributed. In this case too, the our method is more effective than the full synchronization algorithm type. Compared Figure 5 with Figure 7, it is evident that the new algorithm is further effective in the case where the transaction is almost evenly distributed.

### 5.4 Speedup Evaluation of New Algorithm

Figure 8 shows the rate of speedup improvement for new algorithm itself. We can clearly see that the rate of speedup improvement improves as the number of processor increases from Figure 8.

## 6. CONCLUSIONS

In our study, the increased quality is guaranteed in the continuous-event parallel simulation, and a new synchronization algorithm which enables to perform the simulation with improved processing efficiency is incorporated into the congested road-traffic system. Then, the experiments are carried out in the continuous-event parallel simulation environment so as to verify the effect of the processing efficiency.

The synchronization algorithm in this new continuous-event parallel simulation is one for road traffic system. It is a system of the transaction flow type which can be found almost everywhere in the society in general. For example, the system includes

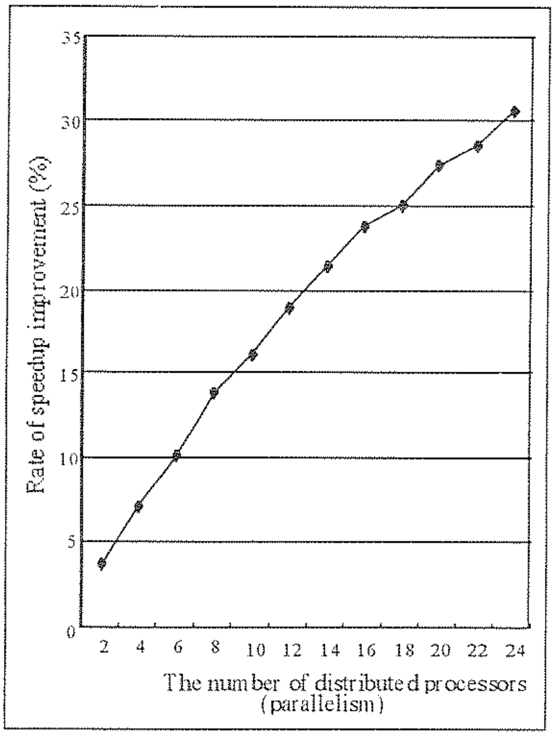


**Figure 6.** Improvement of new algorithm against full Synchronization algorithm type.

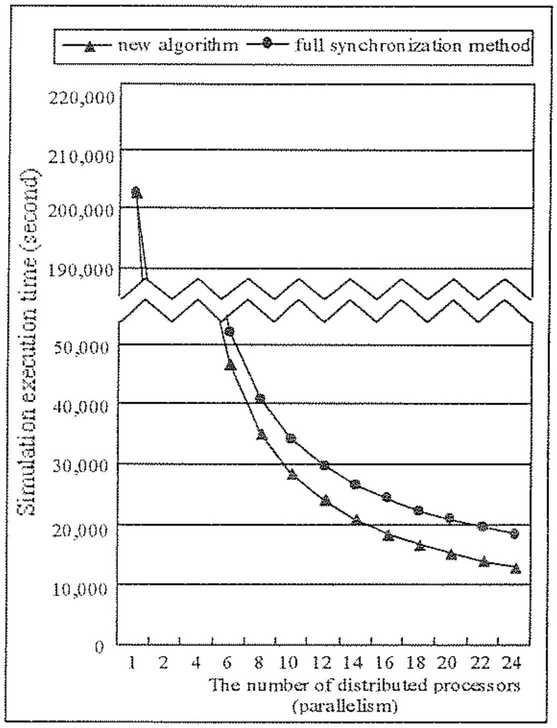


**Figure 7.** Simulation results (not evenly distributed).

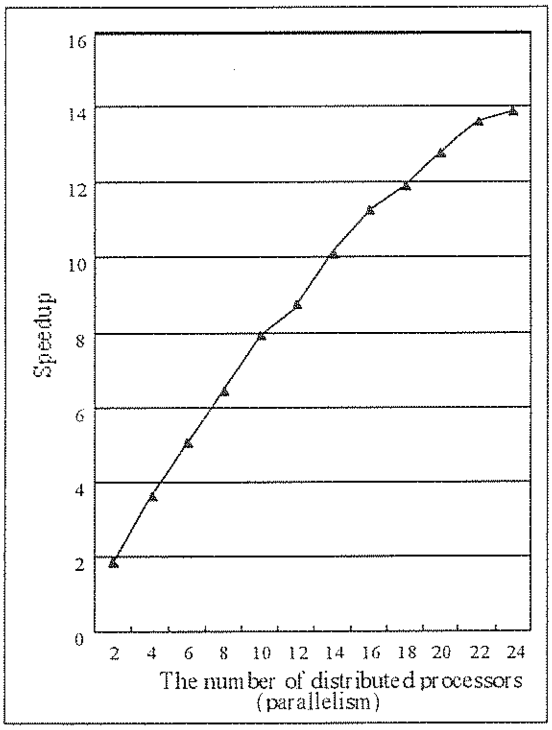


**Figure 8.** Speedup rate of new algorithm itself (almost evenly distributed).

a railroad transport system and telecommunication system.

In the future, we will examine possible applications to other various systems, in the hope that we can analyze them in order to gain its general-purpose use and efficiency with further efficient synchronization method.

## 7. REFERENCES


Ayani, R. and H. Rajaei, Parallel simulation using conservative time windows, Proceedings of the 1992 Winter Simulation Conference, 709-717, 1992.

Galluscio, A. P. and J. T. Douglass, A comparison of two methods for sdvancing time in parallel discrete event simulation, Proceedings of the 1995 Winter Simulation Conference, 650-657, 1995.

Konas, P. and P. C. Yew, Processor self-scheduling in parallel discrete event simulation, Proceedings of the 1995 Winter Simulation Conference, 682-689, 1995.

Mabry, S. L. and G. L. Gaudiot, Distributed parallel object-oriented environment for traffic simulation (POET), Proceedings of the 1994 Winter Simulation Conference, 1093-1100, 1994.

Misra, A., Distributed discrete-event simulation., ACM Comp., surveys. Vol.18, 39-65,1986.

Nakanishi, T., K. Yikai, J. Satoh and A. Satoh, The development of a road traffic simulation system in broad areas, *Mathematics and Computers in Simulation*, 39(3-4), 207-212, 1995.

Namekawa, M., A. Satoh, K. Tsuchida, K. Yikai and T. Nakanishi, A parallel simulation based on a road traffic network system, Proceedings of the International Congress on Modelling and Simulation, 331-336, 1995.

Namekawa, M., A. Satoh, H. Mori, K. Tsuchida, K. Yikai and T. Nakanishi, Clock synchronization algorithm for parallel road-traffic simulation system in a wide area, *Mathematics and Computers in Simulation*, 48(4-6), 351-359, 1999.

Satoh, A., J. Satoh, I. Miyoshi and M. Takahashi, Data Input Subsystem for the Simulation through MITRAM, Proceedings of the 11th Int. Congress on Modelling and Simulation, Vol.4, 325-330, 1995.

Takai, M., T. Yamashiro and S. Narita, Evaluation of conservative synchronization methods in parallel discrete event simulation. *Journal of Information Processing Society of Japan*, 32(11), 2010-2019, 1996.

Yikai, K. and A. Satoh, Verification for road model of road traffic simulation system with microscopic vehicle model, *Journal of Japan Simulation Society*, 18(3), 52-61, 1999.