Urban Traffic Signal Control Based on Distributed Constraint Satisfaction

Kazunori Mizuno
Department of Computer Science
Takushoku University
815-1, Tatemachi, Hachioji, Tokyo 193-0985, Japan
mizuno@cs.takushoku-u.ac.jp

Yukio Fukui Seiichi Nishihara
Department of Computer Science
University of Tsukuba
Tsukuba, Ibaraki 305-8573, Japan
{nishihara, fukui}@cs.tsukuba.ac.jp

Abstract

Urban traffic problems including traffic accidents and traffic congestion or jams have been very serious for us. Urban traffic flow simulation has been important for making new control strategies that can reduce traffic jams. In this paper, we propose a method that can dynamically control traffic signals by equivalently representing as the constraint satisfaction problem, CSP. To solve local congestion in each intersection, we define the whole system as multiagent systems where the represented CSP is extended to distributed CSP, DCSP, in each of which variable is distributed among each intersection agent. Each intersection agent determines some signal parameters by solving the DCSP. The proposed method is implemented on our separately developed agent-oriented urban traffic simulator and applied to some roadnetworks, whose experimental simulations demonstrated that our method can effectively reduce traffic jams even in the roadnetworks where traffic jams are liable to occur.

1. Introduction

Recent remarkable progress of our traffic environments including the increase in cars and improvement of roadnetworks have given us the good results industrially and economically. In another aspect, however, urban traffic problems, including traffic jams and accidents, and environmental problems, including noise and air pollution, have been serious for us. In particular, for traffic jams, which are inveterately happening in urban areas, it can be expected to reduce traffic jams by taking measures to meet the situation.

On the other hand, due to improvement of the computer performances and technologies, there have been many studies on urban traffic flow simulations within a PC[1, 4, 5, 6, 7, 10, 11, 16, 17, 19], enabling us to try to verify the effective measures for reducing traffic jams in advance.

To reduce traffic jams, controlling traffic signals can be one of the most effective measures[2, 13, 24, 26]. The present control methods in Japan, based on selection out of some patterns that are made by the experts beforehand, may be difficult to control traffic signals dynamically and flexibly. It also seems to need much costs for the increase in the number of intersections because of the centralized control method.

In this paper, we propose the method that can dynamically control traffic signals according to the current traffic situation. We represent determining the parameters for controlling as the constraint satisfaction problem, CSP. Assuming that traffic jams can reduce by solving local congestion in each intersection, we also define the whole system as the multi-agent systems[12, 18] where the represented CSP is extended to the distributed CSP, DCSP, in each of which variable is distributed among intersection agents. Each intersection agent dynamically determines the parameters of its own traffic signals by solving the CSP while receiving congestion information from road agents connected to itself. The proposed method is implemented in our separately developed agent-oriented urban traffic simulator, where we experimentally demonstrate that our method can reduce traffic jams.

In section 2, after describing traffic signals and their parameters for controlling, we refer to the present control method in Japan. In section 3, the proposed control method is given: we first introduce our developed traffic simulator. Then, we represents controlling signals as the CSP and give the procedure for constraint satisfaction processes. In section 4, we show experimental results where our method is applied to some roadnetworks. In section 5, we briefly review related works for controlling traffic signals and the intelligent transportation system, ITS, in Japan.

1530-1605/08 $25.00 © 2008 IEEE

2. Distributed constraint satisfaction and urban traffic signal

2.1. Distributed constraint satisfaction

A constraint satisfaction problem, CSP, involves finding values for problem variables which are subject to constraints specifying the acceptable combinations of values. Such combinatorial search problems are ubiquitous in artificial intelligence and pattern analysis, including scheduling and planning problems. A CSP is defined as (V, D, C), where V is a set of variables, D is a set of values to be assinged to variables, and C is a set of constraints required among variables. A distributed CSP, DCSP, is the CSP in which variables and constraints are distributed among multiple agents[8]. It consists of the following:

- a set of *agents*, $1, 2, \cdots, k$,
- a set of CSPs, $P_1, P_2, \cdots, P_k$, such that P_i belongs to agent i and consists of
  - a set of *local variables* whose values are controlled by agent i,
  - a set of *intra-agent constraints*, each of which is defined over agent i's local variables,
  - a set of *inter-agent constraints*, each of which is defined over agent i's local variables and other agents' local variables.

Various application problems in multi-agent systems, e.g., the distributed resource allocation problem[3], the distributed scheduling problem[21], the distributed interpretation task[15], and the multiagent truth maintenance task[9] can be represented as DCSPs.

2.2. Traffic signals and parameters for controlling

A traffic signal signs one of three colors, *green*, *yellow*, and *red*, and also works cyclically at all times. To control traffic signal, it is necessary to fix three parameters, *cyclelength*, *split*, and $offset$, after revelation of signals is determined.

- *cyclelength*: time length where the color of the signal goes around once,
- *split*: time distribution of green time length,
- $offset$: time lag of the green time period between signals at the adjacent intersections.

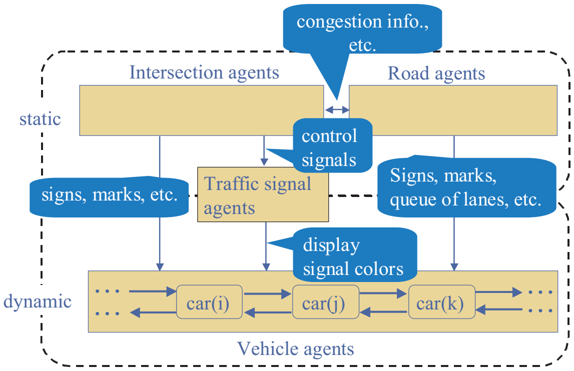


Figure 1. The model of our traffic flow simulator.

2.3. The present control method

In Japan, two methods, the pattern selection method and the MODERATO method, are mainly used to control traffic signals. The former selects one of several patterns fixed by the experts which investigate traffic situations in advance. The latter fixes *cyclelength* and *split* by calculating a pattern that minimizes a delay time in which cars pass through intersections. The $offset$ parameter selects a pattern same as the former.

These methods have some drawbacks to dynamically control signals. It is difficult to flexibly cope the traffic conditions that are not assumed since parameter patterns have to be fixed in advance. Constructing the patterns also needs an enormous cost and advanced knowledge for traffic. Since traffic flow tends to be changeable over time, the patterns should be adjusted and reconstructed periodically.

3. Constraint-based traffic signal control

3.1. Basic ideas

Solving local congestion at each intersection can make traffic jams reduce in the entire roadnetwork. We therefore propose the method for controlling signals at each intersection. The basic ideas of the proposed method are as follows.

i) We define the problem for fixing the signal parameters described in 2.2 as the CSP. The CSP is extended to DCSP, in each of which local variable is distributed to each intersection agent.

ii) The proposed method is implemented in the our separately developed traffic simulator, each of which intersection agent receives congestion information from road agents connected to the intersection.

```
**Intersection_agent**{
/* common data */
  $P$; (coordinates)
  $Sig$; (No. of signal agents belonged to)
  $S_j$; (signal agent ID, $j = 1, \cdots, Sig$)
  $Cl$; (No. of road agents connected to)
  $Cid_k$; (connected road agent ID, $k = 1, \cdots, Cl$)
  $p\_link_l$; (list of pairlink, $l = 1, 2$)
  $Wmax, Wmin$; (congestion degree in connected roads)
  $sline$; (existence of stop line)
  etc...
  /* Traffic signal agent */
  **Signal_agent**{
    $cid$ (intersection ID belonged to);
    $color$; (current color indication, $red, yellow, green$)
    etc...
}
**Road_agent**{
  $N_j$; (connecting intersection ID, $j = 0, 1$) ;
  $Tp$; (kind of road)
  $Wd$; (breadth)
  $Ct$; (kind of the center line)
  $Ln$; (No. of lanes)
  $Sn$; (No. of road signs)
  $cqueue_k$; (queue of each lane, $k = 1, \cdots, Ln$)
  $J$ (amount of congestion)
  etc...
}
```

Figure 2. An example of data sets for intersection and road agents

iii) Since each intersection agent has multiple local variables, the search algorithm to solve the DCSP is based on the algorithm described in [25].

3.2. Our agent-oriented traffic flow simulator

We have developed a agent-oriented traffic flow simulator[22]. In this system, four elements, i.e., vehicles, intersections, traffic signals, and roads, of which traffic flow is composed, are modelled as agent sets to be cooperated, that is, interaction among these agents causes several traffic phenomena.

The system is defined as a triple, $Sys =< Car, Intersection, Road >$, where $Car (= \{c_1, \ldots, c_N\})$ is the set of vehicle agents that directly make up traffic flow. $Intersection$ and $Road$ are sets of intersection and road agents, respectively, in each of which element provides various types of information in deciding actions in vehicle agents. A roadnetwork is given as a graph structure, where $Intersection$ and $Road$ correspond to sets of vertices and edges respectively. Fig. 2 gives data sets that intersection and road agents should hold. $Intersection$, which has information, e.g., existence of signal agents and pairlink (See the below definition and Fig. 3), etc., can get congestion information from connected road agents. According to the degree of congestion, signal parameters are fixed, being sent to signal agents. Signal agents sign colors of signals.

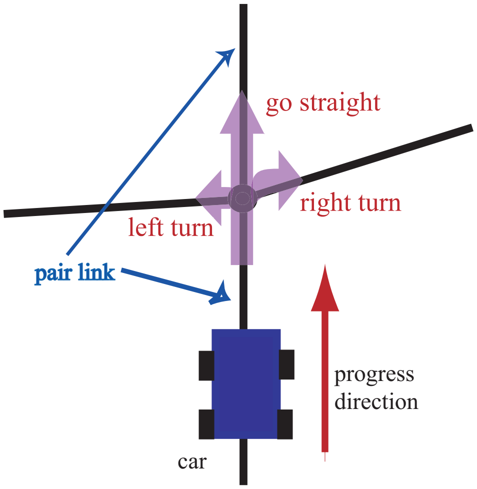


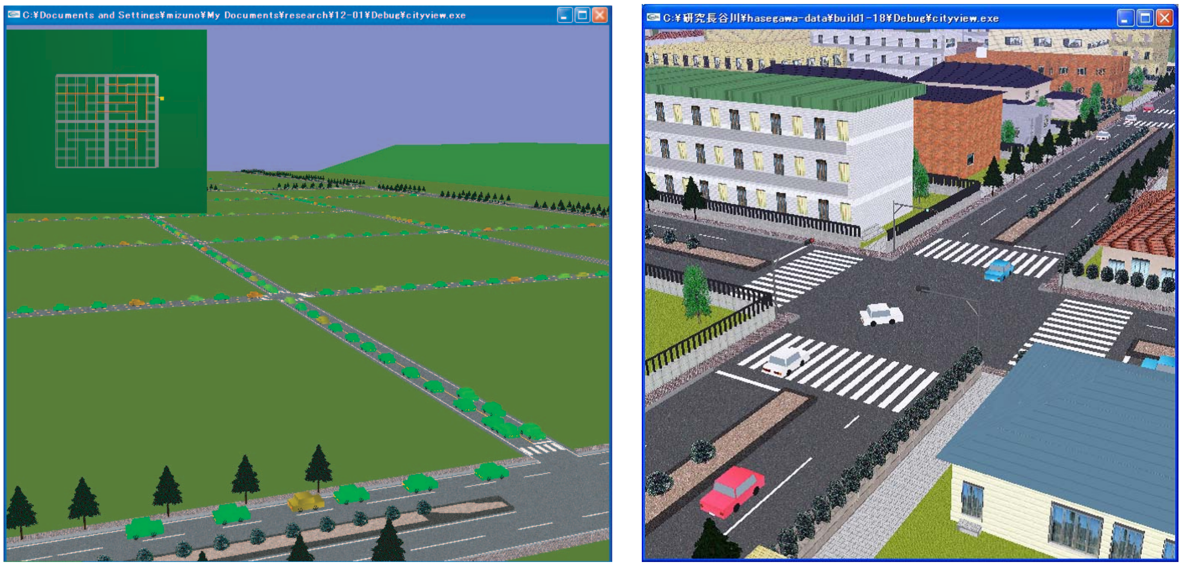
Figure 3. An example of a pairlink.

Definition 1 ***pairlink***
A pair of roads, or links, out of four roads which are connected to the same intersection, where passing through is allowed by the green signal. In the intersection without signals, a pair of roads both of which are as straight as possible.

Road provides several information , e.g., road marks and signs, bus stops, and lane queues, etc., that should be recognized by vehicle agents.

Car, or vehicle agent sets, can update their position, direction and speed by deciding its own next action by using two types of knowledge, or basic rules and individual rules[22].

We have developed our traffic flow simulator as a 3D animation system to visualize more realistic traffic phenomena, where we integrate our simulator with our separately developed virtual city generation system[23]. Fig. 4 shows execution examples of our simulator, where the example (b), in particular, gives a result where traffic flow appears in a virtual city.



(a) Simple mode.

(b) Virtual city mode.

Figure 4. Example screens on our developed urban traffic flow simulator.

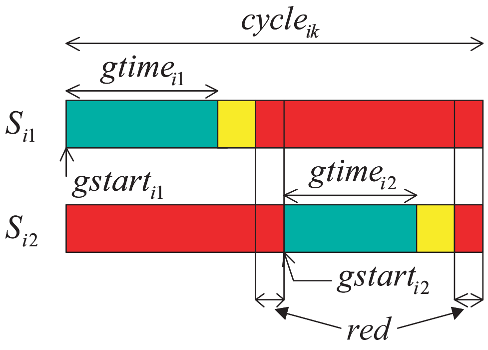


Figure 5. Variables which the intersection agent *i* has.

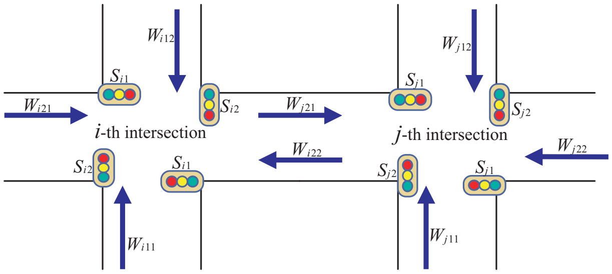


Figure 6. An example of adjacent signals and the degree of congestion.

3.3. Representation

Fixing signal control parameters is represented as a triple $P = (V, D, C)$.

$$\begin{aligned}
P &= (V, D, C), \\
V &= \{cycle_{ik}, gstart_{ik}, gtime_{ik}\}, (i = 1, \ldots, n,\ k = 1, 2) \\
D &= \{D_{cycle}, D_{gstart}, D_{gtime}\}, \\
C &= \{C_{in}, C_{adj}, C_{road}\},
\end{aligned}$$

where V is a set of variables, in which elements, $cycle_{ik}$, $gstart_{ik}$, and $gtime_{ik}$, correspond to time length of one cycle, start time period of the green sign in the cycle, and time length of the green sign, respectively, in the k-th signal belonging to the i-th intersection agent, as shown in Fig. 5. D is a set of values to be assigned to variables, where $D_{cycle} = \{40, \ldots, 140\}$ and $D_{gstart} = D_{gtime} = \{0, \ldots, 140\}$. Therefore, a signal has cycle time length from 40 to 140 and

```
procedure constraint_check()
  for (all intersections with signals)
    check constraints in $C_{road}$;
  end for
  while (all constraints are not satisfied)
    check constraints in $C_{in}$ according to the priority value;
    check constraints in $C_{adj}$ according to the priority value;
  end while
  apply the arranged values to signals;
end.
```

Figure 7. The constraint satisfaction process.

green time length from 0 to 140. C is a set of constraints composed of subsets, C_{in}, C_{adj}, and C_{road}.

(1) C_{in}: constraints required inside one intersection agent

C_{in} is the subset of constraints to be satisfied between signals, S_{i1} and S_{i2}, belonging to the same intersection i. The constraints are very hard, which must be satisfied not to make traffic interrupt. Constraints in C_{in} are as follows.

- Cycle time length of S_{i1} and S_{i2} must be same, that is,
  $cycle_{i1} = cycle_{i2}$,
- S_{i1} and S_{i2} must not be the green sign in the same time, that is,
  $gstart_{i2} = gstart_{i1} + gtime_{i1} + yellow + red \times 2$,
  $gstart_{i1} = gstart_{i2} + gtime_{i2} + yellow + red \times 2$,
- Both green time length of S_{i1} and S_{i2} should be length from 20% to 80% of cycle time length, that is,
  $cycle_{ik} \times 0.2 < gtime_{ik} < cycle_{ik} \times 0.8$.

(2) C_{adj}: constraints on signals between the adjacent intersection agents

C_{adj} is the subset of constraints on signals, S_{ik} and S_{jk}, between adjacent intersections, i and j. The constraints are soft, which are desirable to be satisfied to make traffic smooth.

Definition 2 ***adjacent signals***
When two intersections, i and j, with signal agents, S_{ik} and S_{jk} are adjacent in the sense of graph structures and length of the link (i, j) is less than a constant value, S_{i1} and S_{j1} at the same direction are adjacent signals.

Constraints in C_{adj} are as follows.

- Cycle time length of adjacent signals may be same, that is,
  $cycle_{ik} = cycle_{jk}$,
- Start time period of the green sign of S_{ik} may be shifted from that of S_{jk}, that is,
  $gstart_{ik} = gstart_{jk} + Dist_{ij}/LV_{ij}$,
  where $Dist_{ij}$ and LV_{ij} denote the distance and speed limit at the road between i and j.

(3) C_{road}: constraints decided by congestion information from road agents

The constraints can play the role in resolving local congestion, in which values are reassigned to variables according to congestion information from road agents. The degree of congestion for each road is defined by $W = CL/RL$, where CL denotes length of car rows and RL length of the road. The maximum and minimum degrees, $Wmax_{ikl}$ and $Wmin_{ikl}$, of congestion are also calculated for each lane, l, of roads approaching to intersections, where $Wmax_{ikl}$ is the degree of congestion when the color of signal is just changed to *green* from *red* and $Wmin_{ikl}$ is one when the color is just changed to *red* from *yellow*. We define the priority value, P_value_{ik} of signal, S_{ik} for priorly assigning values to variables in CSPs, calculated as $P\_value_{ik} = Wmin_{ikl} + Wmax_{ikl}$.

According to W and P_value, assignment of values to variables is reassigned subject to the following constraints.

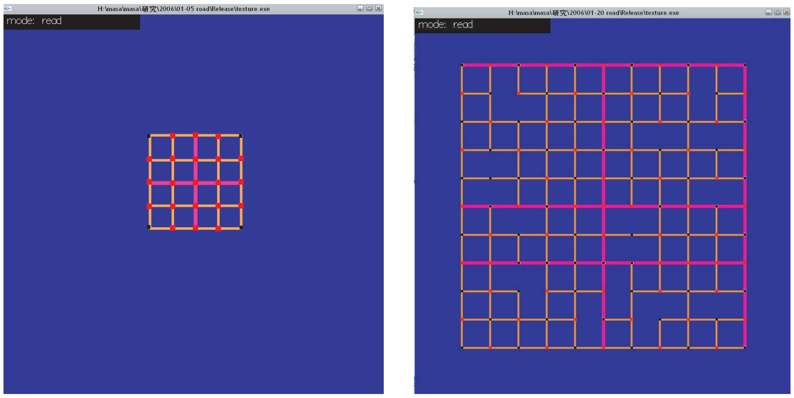
- If there are many cars in roads connected to the intersection i when the signal is just changed to *yellow* from *green*, cycle time length of the signal is made longer, that is,
  $Wmin_{ikl} > 0.4 \Rightarrow cycle_{ik} + = C$, where C is a constant value,
- if there are few cars in four roads connected to the intersection i when the signal is just changed to *yellow*, cycle time length of the signal is made shorter, that is,
  for all l, $Wmin_{ikl} < 0.4 \Rightarrow cycle_{ik} - = C$, where C is a constant value,
- the offset value is taken account if roads are congested, that is,
  $Wmax_{ikl} > 0.4 \Rightarrow gstart_{ik} = gstart_{jk} + Dist_{ij}/LV_{ij}$,
- green time length of each signal is decided accoring to its road congestion, that is,

$$gtime_{ik} = \frac{Wmax_{ik1} + Wmax_{ik2}}{\sum_{k=1}^{2} \sum_{l=1}^{2} Wmax_{ikl}}$$

.

3.4. Algorithm

Fig. 7 gives the procedure for solving CSP defined in section 3.3. The procedure is basically executed for each



(a) The roadnetwork A.

(b) The roadnetwork B.

Figure 8. Roadnetworks used in experiments.

Table 1. Conditions of experiments.

roadnetwork	No. of pairs of signals	No. of cars	execution time
roadnetwork A	21	100, 200	60*min*
roadnetwork B	60, 113	1000, 1300	30*min*

group of adjacent signals. Parameters fixed by the procedure are applied to each signal after the current cycle is terminated. In each intersection agent, checking constraints in C_{road} is first executed while congestion information is received from connected road agents. Then, checking constraints in C_{in} and C_{adj} are executed according to the priority value, P_value_{ik} for each signal, S_{ik}.

(1) check constraints in C_{road}

Each intersection agent receive the degrees, W, of congestion from connected road agents whenever one cycle is terminated. In the intersection agent i, $Wmax_{ikl}$, and $Wmin_{ikl}$ are fixed by W and the priority value, P_value_{ik} is calculated from $Wmax_{ikl}$ and $Wmin_{ikl}$. The intersection agent i changes values assigned to its local variables, $cycle_{ik}$, $gstart_{ik}$, and $gtime_{ik}$, using $Wmax_{ikl}$ and $Wmin_{ikl}$.

(2) check constraints in C_{in}

In this phase, for each intersection agent, constraints defined within the intersection agent i are first checked according to P_value_{ik}. When the k-th signal, S_{ik}, of the agent i has higher P_value_{ik}, the pairlink k has more congested roads. Thus, values of variables in the signal S_{ik} with higer P_value_{ik} are precedently fixed such that constraints in C_{in} are satisfied. Then, those in S_{ik} with lower P_value_{ik} are automatically changed subject to S_{ik} with higher P_value_{ik}. When all constraints within the intersection agent i are satisfied, lower P_value_{ik} is changed to the same value as higher P_value_{ik} because constraints in C_{adj} are checked in the next phase. The intersection agent i sends values assigned to each variable, $cycle_{ik}$, $gstart_{ik}$, and $gtime_{ik}$, and the priority value, P_value_{ik}, to adjacent intersection agents.

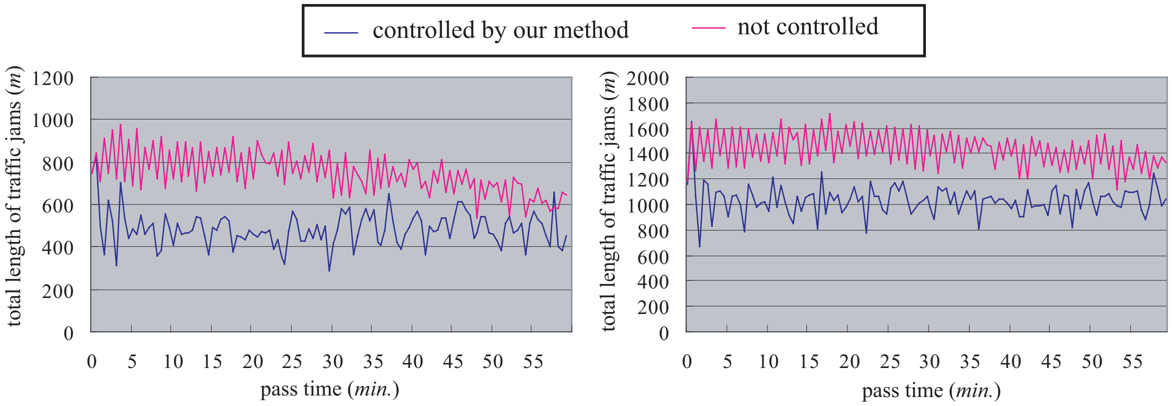
(3) check constraints in C_{adj}

As well as the previous phase, this phase takes precedence over S_{ik} with higher P_value_{ik}. If values assigned to $cycle_{ik}$ and $cycle_{jk}$ in the adjacent intersection agent, i and j, are different, the value of $cycle$ in the signal with lower P_value is changed to the same value as that in the signal with higher P_value. the constraint for setting offset is satisfied by taking account of traffic in the most congested roads, based on the degree of congestion, $Wmax_{ikl}$.

4. Experiments

We implement the proposed control method in our developed traffic simulator and conduct some experiments. In experiments, we use two roadnetworks as shown in Fig. 8, where the roadnetwork A contains 25 nodes, or intersections, and 40 edges, or roads and the roadnetwork B contains 121 nodes and 202 edges. The distance between intersections is 100 meters in both roadnetworks. Table 1 summarizes each experimental condition. As for evaluation, we measure length of traffic jams in the entire roadnetwork for each 30 second for the execution time in Table 1. We define length of traffic jams as total length of cars with a speed of 0 kilometers per hour, which approach to intersections.

Fig. 9 gives the experimental result in the roadnetwork A. The more cars make traffic jams increase. However, applying our method for controlling signals can make traffic



(a) result on 100 cars.

(b) result on 200 cars.

Figure 9. Experimental result on the roadnetwork A.

case	No. of cars	average length (*m*)	
		our method	not controlled
Fig. 10	1000	4076	4797
(60 signals)	1300	6041	6869
Fig. 11	1000	4342	6070
(113 signals)	1300	5931	7752

Table 2. Comparison of average length of traffic jams between Fig. 10 and Fig. 11.

jams reduce. Average lengths of traffic jams are 486 and 1034 meters for each case of 100 and 200 cars when our method is applied, although there are 748 and 1434 meters for the same case in applying no methods.

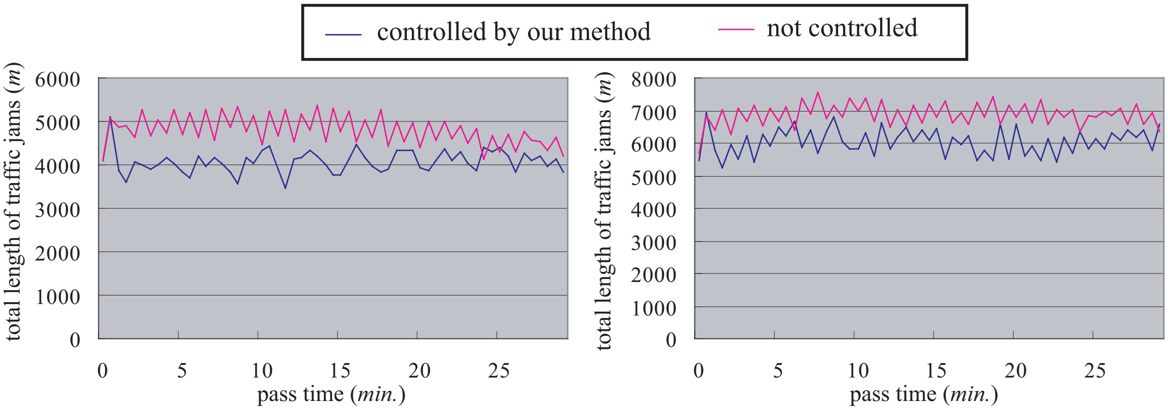
Figs. 10 and 11 give the experimental results in the roadnetwork B. As well as Fig. 9, our method can also reduce traffic jams. All results show that it is effective to apply our method for resolving traffic jams. Comparing Fig. 10 which has 60 pairs of signals with Fig. 11 which has approximately doubled pairs of signals, 113, an increase in the number of signals tends to increase traffic jams when no control methods are applied. In contrast, by applying our method as a control strategy, length of traffic jams seems not to increase, although it can be supposed that an increase in the number of signals causes traffic jams. In particular, compared the case of 113 pairs of signals with that of 60 pairs when our method is applied to the case of 1300 cars in Table 2, traffic jams for 113 pairs of signals can be reduced on average rather than those for 60 pairs.

5. Related work

As for traffic signal control methods adopting intelligent approaches, there has been some studies. In [2], as well as our study, controlling signals is defined as multi-agent systems, each of which intersection agent determines $cyclelength$ and $split$ by making use of some rules. It may be difficult for this method to fix $offset$ because the intersection agent is not communicated to the other one. In [24], genetic algorithms, GA, are used to fix the parameters. This method puts states, or colors of signals, of each time period of all signals into a chromosome code and tries to minimize the number of cars which is stopping. Therefore, increasing in the number of signals causes inefficient search performance because of the enormous size of the chromosome.

On the other hand, in Japan, due to development and introduction of intelligent transportation systems, ITS[14], advanced traffic management by getting more precise traffic data has been becoming feasible. It may also be easy to control traffic by universal traffic management systems, UTMS[20]. UTMS, which is composed of some subsystems, is a kind of ITS systems and can communicate with individual cars through sensors or radio beacon put at roads, not only providing traffic information to drivers but also aggressively managing travel and distribution information.

Usual methods to control traffic signals are conducted based on data obtained for a few minutes ago. Therefore, these methods can be inconvenience because there is difference between control measures to be conducted and current traffic conditions, i.e., delays of control happen to a greater or lesser extent. UTMS can be available to solve the problem of delays. The characteristics of controlling signals using UTMS are 'prediction of traffic demands', 'real-time



(a) result on 1,000 cars.

(b) result on 1,300 cars.

Figure 10. Experimental result on the roadnetwork B, where there are 60 pairs of signals, i.e. 60 intersections have signals.

control', and 'distributed-type decision of control'. The control methods using UTMS are discussed and trially applied to some areas in Japan[13].

6. Conclusion

We have proposed the method that dynamically control urban signals according to the congestion situation in roadnetworks. Our method represents adjusting control parameters as the CSP. The CSP is extended to distributed CSP, in which intersection agents have some local variables. The intersection agent receives congestion information from road agents connected to it and changed values from adjacent intersection agents. Then, while taking account of these information and the priority value, the agent attempt to change or reassign values subject to constraints. Constraints in C_{road} and the constraint for setting offset in C_{adj} especially play the most important role in resolving local congestion, resulting in reducing entire jams. We implemented our method to the agent-oriented traffic flow simulator, in which experimental simulation demonstrated that our method can reduce traffic jams. In particular, our method have a great effect even for environments where traffic jams liable to increase, i.e., increased number of signals.

Our future works should consist in applying our mehotd to more realistic roadnetworks and extending our method by including deciding revelation of signals. It should be important to make our method cooperate UTMS or ITS systems.

Acknowledgement

This research was supported in part by the Minisry of Education, Culture, Sports, Science and Technology of Japan, Grant-in-Aid for Exploratory Research, No. 18650017, 2006–2009.

References

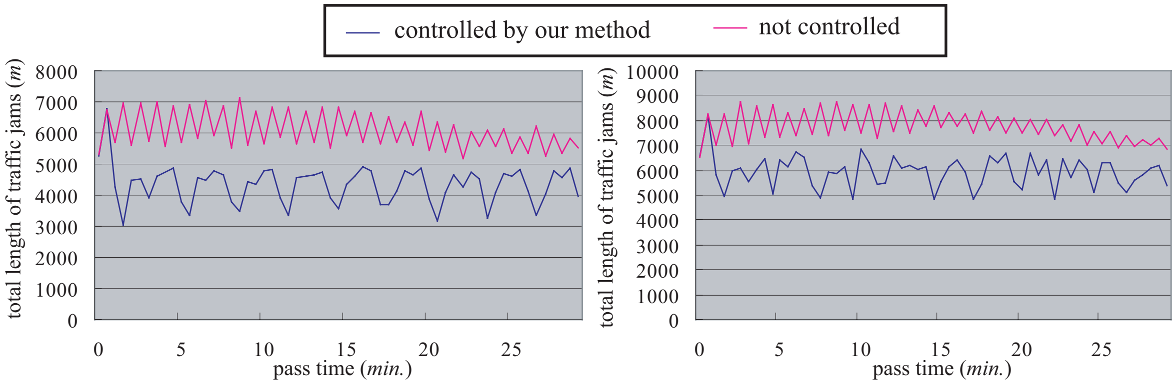
[1] Barrett, C. L. *et. al.*, TRANSIMS (TRansportation ANalysis SIMulation) volume0 – Overview, *Tech. Rep., LA-UR-99-1658* (1999).

[2] Chen, R. S., Chen, D. K., and Lin S. Y., ACTAM: Cooperative Multi-Agent System Architecture for Urban Traffic Signal Control, *IEICE Trans. Inf. & Syst.*, Vol. E88–D, No. 1, pp. 119–126 (2005).

[3] Conry, S. E. *et. al.*, Multistage negotiation for distributed constraint satisfaction, *IEEE Trans. Systems, Man, and Cybernet.*, Vol. 21, No. 6, pp. 1462–1477 (1991).

[4] Dia, H., An agent-based approach to modelling driver route choice behaviour under the influence of real-time information, *Transportation Research Part C*, Vol. 10, pp. 331–349 (2002).

[5] Ehlert, P. A. M., and Rothkrantz, L. J. M., Microscopic traffic simulation with reactive driving agents, *2001 IEEE Intelligent Transportation Systems Conference Proceedings*, pp. 861–866 (2001).



(a) result on 1,000 cars.

(b) result on 1,300 cars.

Figure 11. Experimental result on the roadnetwork B, where there are 113 pairs of signals, i.e. 113 intersections have signals.

[6] Fernandes, J. M., and Oliveira, E., TraMas: Traffic Control through Behaviour-based Multi-Agent System, *Proc. PAAM'99*, pp. 457–458 (1999).

[7] Helbing, D. *et. al.*, Micro- and Macro-Simulation of Freeway Traffic, *Mathematical and Computer Modelling*, Vol. 35, pp. 517–547 (2002).

[8] Hirayama, K. and Yokoo, M., The distributed breakout algorithms, *Artficial Intelligence*, Vol. 161, pp. 89–115 (2005).

[9] Huhns, M. N. and Bridgeland, D. M., Multiagent truth maintenance, *IEEE Trans. Systems, Man, and Cybernet.*, Vol. 21, No. 6, pp. 1437–1445 (1991).

[10] KLD associates, Inc., TRAF NETSIM USER GUIDE (1995).

[11] Li, M. *et. al.*, A Cooperative Intelligent System for Urban Traffic Problems, *Proc. IEEE Symp. Intelligent Control*, pp. 162–167 (1996).

[12] Mackworth, A. K. *et. al.*, A Formal Approach to Agent Design: An Overview of Constraint Based Agents, *Constraints*, Vol. 8, No. 3, pp. 229–242 (2003).

[13] Metropolitan Police Department Website (*in Japanese*), `http://www.npa.go.jp/koutsuu/kisei/index.html`

[14] Ministry of Land, Infrastructure and Transport Japan Website, `http://www.mlit.go.jp/road/ITS/`

[15] Mason, C and Johnson, R., DATMS: a framework for distributed assumption based reasoning, *Distributed Artif. Intell.*, Vol. 2, pp. 293–318 (1989).

[16] Nagel, K. *et. al.*, Traffic at the edge of chaos, *Artficial Life IV*, pp. 222–235 (1994).

[17] Nagel, K. *et. al.*, TRANSIMS for Urban planning, *Tech. Rep., LA-UR-98-4389* (1999).

[18] Nareyek, A., *Constraint-Based Agents*, LNAI-2062, Springer (2001).

[19] Raney, B. *et. al.*, Large Scale Multi-Agent Transportation Simulations, *42nd ERSA (European Regional Science Association) Congress* (2002).

[20] Universal Traffic Manegement Society of Japan Website, `http://www.utms.or.jp/english/index.html`

[21] Sycara, K. P. *et. al.*, Distributed constrained heuristic search, *IEEE Trans. Systems, Man, and Cybernet.*, Vol. 21, No. 6, pp. 1446–1461 (1991).

[22] Yamada, M., Mizuno, K., Fukui, Y., and Nishihara, S., An Agent-Based Approach to Urban Traffic Flow Simulation, *Proc. International Workshop on Advanced Image Technology 2006 (IWAIT2006)*, pp. 502–507 (2006).

[23] Yamada, K., Samamaki, M., Mizuno, K., Fukui, Y., and Nishihara, S., Automatic Generation of Building Shapes for City Views, *Proc. International Workshop*

on Advanced Image Technology 2006 (IWAIT2006), pp. 484–489 (2006).

[24] Yamamoto, N. and Morishita, S., Simulation of Traffic Flow by Cellular Automata and its Control (*in Japanese*), *Journal of The Japan Society of Mechanical Engineers (C)*, Vol. 65, No. 637, pp. 3553–3558 (1999).

[25] Yokoo, M. and Hirayama, K., Distribute Constraint Satisfaction Algorithm for Complex Local Problems (*in Japanese*), *Journal of Japan Society for Artificial Intelligence*, Vol. 12, No. 2, pp. 348–354 (2000).

[26] Wiering, M. *et. al.*, Intelligent Traffic Light Control, *Tech. Rep., UU-CS-2004-029* (2004).