Q Value-Based Dynamic Programming with Boltzmann Distribution in Large Scale Road Network

Shanqing Yu∗, Yelei Xu∗, Shingo Mabu∗, Manoj Kanta Mainalī∗, Kaoru Shimada∗, and Kotaro Hirasa was∗

Abstract: In this paper, a global optimal traffic assignment strategy, i.e., Q value-based Dynamic Programming with Boltzmann Distribution is applied to the Kitakyushu City traffic system. The main idea of the proposed traffic assignment strategy is to calculate the expected traveling time for each origin-destination pair and the probability of selecting the next section, then to generate a considerable number of route candidates for the drivers based on the calculated probability. In the simulation, how to select the temperature parameter and the number of the route candidates is discussed in detail. The comparison between the proposed method and the shortest path algorithms indicates that the proposed method could reduce the risk of the traffic congestion occurrence and save the traveling cost effectively. In addition, the computation time is given to reveal the feasibility of the proposed method in large scale networks.

Key Words: Q value-based dynamic programming, Boltzmann distribution, shortest path algorithm.

1. Introduction

As the development of GPS and computer technology, many Intelligent Transportation Systems (ITS), which provide users with not only static traffic information, but also real time traffic situations, have been developed to improve the efficiency of the traffic systems and save the traveling cost [1]–[4]. Route assignment, as one of the most important parts of the vehicle navigation systems in ITS, concerns the selection of routes between origins and destinations. In most of the traditional navigation systems, the greedy strategy which only calculates the shortest paths using conventional algorithms such as Dijkstra algorithm [5], A∗ algorithm [6],[7], and Q value-based Dynamic Programming [8], is adopted to do the route assignment. Along with the wider applications of ITS in modern metropolises, some researchers also figure out that the greedy route assignment may cause the traffic congestion because of some potential negative behavioral phenomena, such as concentration and overreaction [9],[10].

In the recent decades, many researchers have contributed to study the multiple-path algorithms like the k-shortest path algorithms [11]–[13] and the totally disjoint path algorithms [14],[15]. Although these algorithms are successful in searching multiple paths with certain constraints, how to assign the traffic to these selected paths is not discussed. Researchers in [16] calculates k shortest paths for the drivers every operational time interval. However, it just applied to a very simple road network and they just equally distribute the traffic volume to each generated shortest paths. In addition, in order to get the global optimal routes, an additional path checking procedure is to be performed to select the paths with acceptable constraints for the drivers. The reference [17] proposed a Pretrip multipath planning method, in which some partly disjoint alternative paths are calculated offline to render the system less reliant on real-time traffic information and consequently save the computation time during the trip. However, this method only provides alternative paths for the drivers when the original route is not available and it is not aiming at reducing the traffic congestion and improving the traffic system performance in global perspective.

For the purpose of improving the route assignment in the navigation systems, a global optimum traffic routing strategy, that is, quasi-Q value-based Dynamic Programming with Boltzmann Distribution [18]–[20] has been already proposed, where Q value-based Dynamic Programming [21] and Boltzmann distribution [22] are used to minimize the total traveling time of all the vehicles considering the traffic volumes. In this paper, we extend the above original algorithm. The main idea of the proposed method in this paper is to combine Q value-based Dynamic Programming and Boltzmann Distribution to calculate the expectation of the traveling time of each origin-destination pair in the traffic system and the probability of each section to be selected. Based on the calculated probability, a considerable number of route candidates are generated for distributing the vehicles.

Comparing with other multiple-paths algorithms, the proposed Q value-based Dynamic Programming with Boltzmann Distribution has the following three distinguished advantages.

• Unlike other multiple-paths algorithms, the proposed method not only calculates multiple paths, but also shows how many vehicles are assigned to each path in order to get the global optimal routes, because each car has its own route calculated by the probability. Therefore, it is not necessary to do sorting, constraints checking or selection for the paths.

• The computation time of Q value-based Dynamic Programming with Boltzmann Distribution is acceptable even if it is applied to large scale road networks.

• The traffic system performance is less affected by the length...
of the real time traffic information update interval when using Q value-based Dynamic Programming with Boltzmann Distribution [20].

In this paper, the proposed Q value-based Dynamic Programming with Boltzmann Distribution is applied to the static Kitakyushu City traffic system, where the fixed traffic volume is given for each origin-destination pair. The traveling time of each section is updated by the well-known Bureau of Public Roads (BPR) volume-delay function [23],[24], which is described in the next section, in each iteration for converging the traffic volume of each section. In the simulations, it is discussed how to select the temperature parameter and the number of the route candidates in detail. The comparison between the proposed method and the shortest path algorithm indicates that the proposed method could reduce the risk of the traffic congestion occurrence and save the traveling cost effectively. In addition, the computation time is given to reveal the feasibility of the proposed method in large scale networks.

This paper is organized as follows: In the next section, the outline of Bureau of Public Roads (BPR) volume-delay function and the conventional Q value-based Dynamic Programming are reviewed, while Q value-based Dynamic Programming with Boltzmann distribution is described in section 3. The details of the proposed procedure for analysis are described in section 4. In addition, section 5 shows the simulations and section 6 is devoted to conclusions.

2. Overview

In this section, Bureau of Public Roads (BPR) volume-delay function and Q value-based Dynamic Programming are reviewed.

2.1 BPR Volume-Delay Function

In order to obtain the relationship between the traffic volume and traveling time, the following most widely used volume delay function, BPR function [23],[24], is used to calculate the traveling time of each section from its traffic volume in this paper.

\[ t_{ij} = t_{ij}^0 \left( 1 + 0.15 \left( \frac{v_{ij}}{c_{ij}} \right)^2 \right), \] (1)

where,
- \( s_{ij} \): the section from intersection \( i \) to intersection \( j \)
- \( t_{ij}^0 \): free flow traveling time of section \( s_{ij} \)
- \( c_{ij} \): traffic volume capacity of section \( s_{ij} \)
- \( v_{ij} \): traffic volume of section \( s_{ij} \)

2.2 Q Value-Based Dynamic Programming

Q value-based Dynamic Programming is an algorithm used to calculate the optimal traveling time from every intersection to every destination of the road networks.

The Q value, i.e., \( Q_d(i, j) \), which is defined as the minimum traveling time to destination \( d \), when a vehicle bound for destination \( d \) moves to intersection \( j \) at intersection \( i \), is calculated iteratively based on the following equations.

\[ Q_d(i, j) = t_{ij} + \min_{k \in A(i)} Q_d(j, k), \] (2)

\[ Q_d(d, j) = 0, \quad d \in D, \ j \in A(d) \] (3)

\[ Q_d(i, d) = t_{id}, \quad d \in D, \ i \in B(d) \] (4)

where,
- \( i, j \in I \): suffixes of intersections and their set \( D \) of suffixes of destinations and its set \( t_{ij} \): traveling time from intersection \( i \) to intersection \( j \)
- \( A(i) \): set of suffixes of intersections moving directly from intersection \( i \)
- \( B(i) \): set of suffixes of intersections moving directly to intersection \( i \)

3. Q Value-Based Dynamic Programming with Boltzmann Distribution

In the conventional Q value-based Dynamic Programming [21], Q values of the intersection pairs are updated using Eq. (2), in which the optimal route is selected by calculating \( \arg \min_{k \in A(i)} Q_d(i, j) \), that is, the Greedy Strategy. However, when Boltzmann Distribution is considered, which intersection should be selected as the next intersection is determined based on a certain probability. In this case, just using the minimum traveling time is not enough to represent the prospective traveling time to the destination. Therefore, in the new proposed Q value-based Dynamic Programming with Boltzmann Distribution, the expectation of the traveling time for each Origin-Destination pair is calculated based on the following equations instead of the minimum traveling time in the conventional Q value-based Dynamic Programming. Meanwhile, the definition of the Q value, i.e., \( Q_d(i, j) \), also changes to the expectation of the traveling time to destination \( d \), when a vehicle bound for destination \( d \) moves to intersection \( j \) at intersection \( i \).

\[ P_d^{\omega}(i, j) = \frac{e^{-\tau_{ij}^{\omega} / T_d}}{\sum_{j \in A(i)} e^{-\tau_{ij}^{\omega} / T_d}}, \] (5)

\[ Q_d^{\omega}(i, j) = t_{ij} + \sum_{k \in A(i)} P_d^{\omega}(i, j) Q_d^{\omega-1}(j, k), \] (6)

\[ Q_d^{0}(d, j) = 0, \quad d \in D, \ j \in A(d) \] (7)

\[ Q_d^{0}(i, d) = t_{id}, \quad d \in D, \ i \in B(d) \] (8)

where,
- \( Q_d^{\omega}(i, j) \): Q value of \( i, j \) in the nth iteration
- \( P_d^{\omega}(i, j) \): the probability that the vehicle bound for destination \( d \) moves to intersection \( j \) at intersection \( i \) in the nth iteration
- \( \tau_{ij}^{\omega} \): temperature parameter of intersection \( i \) in the nth iteration

Q values and probabilities for all the pairs of adjacent intersections are initialized as follows.
The flow chart of the proposed method is shown in Fig. 1 and the detail of each module will be explained in the following subsections.

The basic idea of the proposed method is to calculate the average traveling time over all the Origin-Destination pairs using Q value-based Dynamic Programming with Boltzmann Distribution and compare it with the one using Q value-based Dynamic Programming, i.e., the greedy optimal route assignment method under the condition that the traffic system is static, where the fixed traffic volume is given for each Origin-Destination pair.

Therefore, in addition to Q iterations another outer iteration is needed for obtaining the converged traffic volume and traveling time of each section as shown in Fig. 1 using the volume delay function.

Furthermore, a number of the route candidates from each Origin-Destination pair is generated using the probability for each section to be selected in order to evaluate the average traveling time over all Origin-Destination pairs. And also the special consideration for determining temperature $\tau_n$ is studied in order to distribute the vehicles in the road network effectively.
4.1 Temperature Parameter

In this subsection, it is described how the temperature parameter of the Boltzmann Distribution in Eq. (6) influences the evaluation of the proposed method. Let us take the road network in Fig. 2 for example. Figure 3 shows the optimal traveling time and the optimal routes from each intersection to destination d calculated by the shortest path algorithm such as Dijkstra algorithm, A* algorithm, and Q value-based Dynamic Programming, while Fig. 4 displays the Q value and probabilities for each section to be selected, which are calculated by using Q value-based Dynamic Programming with Boltzmann Distribution under different temperature parameters. It is obvious that when the temperature $\tau_i$ is a small value, for example $\tau_i = 2$ in Fig. 4a, Q values are close to the optimal traveling time calculated by shortest path algorithms, and the route selection strategy is similar to the greedy strategy. Meanwhile, if the temperature $\tau_i$ is quite high, each section in the road network may have equal opportunities to be selected like Fig. 4c. Therefore, how to determine the temperature parameter plays a very important role in the proposed method.

Generally speaking, the temperature $\tau_i$ should be set according to the volume of the traffic system. Larger temperature should be adopted to distribute the traffic volume when traffic is heavy. However, the numerical range of Q values also influences the selection of $\tau_i$. Even if the same constant $\tau_i$ is used, the sections with a very small numerical range of Q values will be selected fairly randomly, conversely the ones with a large numerical ranges of Q values will tend to have the greedy strategy. Indeed, these kind of phenomena should be considered more carefully in the large scale road network, where the numerical range of Q values of the sections are quite different with each other.

Thus, in order to distribute the vehicles without being influenced by the dynamic range of Q values in the whole road network, the temperature parameter in the proposed method is not a simple constant, but calculated based on the following average Q value over sections.

$$
\tau_i = \tau_0 \left( \sum_{j \in \mathcal{N}_d(i)} \frac{Q^{(0)}_{ij}(i, j)}{|A(i)|} \right)^{\theta},
$$

where,

$\tau_0, \theta$: constant

When $\theta$ equals 0, the temperature parameter of all the intersections in the road network is equal to $\tau_0$, that is, constant. In this case, the intersections far from the destination, where the numerical ranges of Q values are large, will take the greedier strategy than the intersections close to the destination, where the small numerical ranges of Q values are used. Meanwhile, if $\theta$ equals 1, even if a very small $\tau_0$ is adopted, all the routes are likely to be picked up randomly without considering the Q values so much. How to select the value of $\theta$ and $\tau_0$ is carried out in the simulations of this paper.

It seems from Figs. 3 and 4 that Q value-based Dynamic Programming with Boltzmann Distribution is not better than the greedy strategy. It is because the traveling time of sections are fixed, in other words, the traffic volumes of sections are fixed. In this paper, it is studied how the proposed method outperforms the greedy method by changing the traveling times and traffic volumes of sections depending on the optimal route calculations.

4.2 Generate Route Candidates

In the proposed method, a considerable number of route candidates are generated based on the probabilities calculated by Q value-based Dynamic Programming with Boltzmann Distribution for each OD pair. Then, the traffic volume will be equally assigned to these candidates. But, the candidate routes might be the same if necessary. As a result, the routes with small Q values and large probabilities may be selected more than once and consequently attract more traffic volumes.

When the number of candidate routes is large enough, the probability of each section calculated by the candidate routes will be equal to the probability calculated by the proposed method. However, it couldn’t be too large, since too many candidate routes may waste memory spaces and increase the computing time. Meanwhile, too small number of candidate routes are also not encouraged, since it’s necessary to introduce enough alternative routes to distribute the vehicles in the road networks. The reasonable number of candidate routes is discussed in the simulations.

In addition, the cycles in the routes are forbidden.

4.3 Traffic Assignment

The traffic volumes are assigned to the candidate routes generated by the proposed method using the following equations.

$$
tv^o_{ij}(g) \left\{ \begin{array}{ll}
tv^o_{ij} & \text{if } s_{ij} \in \mathcal{R}_d(g) \\
0 & \text{if } s_{ij} \notin \mathcal{R}_d(g)
\end{array} \right.,
$$

$$
tv_{ij} = \sum_{g \in \mathcal{G}} \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} tv^o_{ij}(g),
$$

where,

$s_{ij}$: section from intersection $i$ to intersection $j$.

$v^o_{ij}$: traffic volume of section $s_{ij}$

$od \in \mathcal{O}D$: suffix of OD pairs and its set

$g \in \mathcal{G}$: suffix of candidate routes and its set

$v^o_{ij}(g)$: traffic volume of section $s_{ij}$ when the vehicle of $od \in \mathcal{O}D$ takes the $g$th candidate route

$\mathcal{R}_d(g)$: the $g$th candidate route of $od \in \mathcal{O}D$

5. Simulation

5.1 Simulation Environment

In this paper, the proposed method are simulated using the Kitakyushu City’s map as shown in Fig. 5, which includes 28,973 intersections and 42,284 bidirectional sections with four kinds of road speed limits, i.e., 30 km/h, 50 km/h, 60 km/h, and 80 km/h. 20 origins and 20 destinations (400 OD pairs), which are mainly located in the center of the city, are also displayed in Fig. 5. Static traffic volume is given to each OD pair.

The average traveling time from origins to destinations over all the candidate routes, which is calculated by Eq. (20), is used to evaluate the two methods. In the conventional shortest path algorithm, the number of $|\mathcal{G}|$ will be considered as 1, since only one route is selected for one OD pair. The traveling time of each section is updated by BPR function in Eq. (1), where, the traffic volume capacity of each section is set at 3600 cars/h. What’s
more, Q value-based Dynamic Programming is adopted as one of the shortest path algorithms to compare with the proposed method.

$$E \leftarrow \frac{1}{|OD|} \frac{1}{|G|} \sum_{od \in OD} \sum_{g \in G} \sum_{t \in R_{od}(g)} t_{ij}$$  \hspace{1cm} (20)

5.2 Analysis on $\theta$ and $\tau_0$

In the proposed method, the temperature parameter plays a very important role. The value of $\tau_0$ should be determined according to the traffic volume of the traffic system and the numerical range of the Q values.

As shown in Eq. (17), the parameter $\theta$ and $\tau_0$ are considered to determine $t_j^{(0)}$. In this subsection, four different $\theta$ are simulated with changing $\tau_0$ as shown in Fig. 6. The traffic volume of each OD pair is equal to 120 cars/h in the simulations. It seems that smaller $\theta$ with larger $\tau_0$ could perform as well as larger $\theta$ with smaller $\tau_0$. However, adopting too small $\theta$ with large $\tau_0$ or too large $\theta$ with small $\tau_0$ may make the intersections with small numerical ranges of the Q values lose their guidance, and consequently cause unexpected random selections on these intersections. Therefore, the deviations of $E$ by using the combinations of $\theta$ and $\tau_0$ which performed best in Fig. 6 are calculated in Table 1. It shows the reasonable combinations of $\theta$ and $\tau_0$ not only minimize the average traveling time, but also reduce the deviation, which means that the traffic system is more stable. Based on Table 1, $\theta = 0.2$ is selected in other simulations.

In addition, in order to reveal the relationship between the traffic volume and temperature parameter selection, the proposed method is simulated in three different traffic conditions, where the traffic volume of each OD pair is equal to 30 cars/h, 60 cars/h and 120 cars/h, respectively, changing $\tau_0$ as shown in Fig. 7. $\theta$ is set at 0.2 and the number of candidate routes $|G|$ is set at 100 during the simulation.

It is obvious from Fig. 7 that the best $\tau_0$ is small when the traffic volume of each OD pair is not large, since the distribution of the vehicles is not necessary, while $\tau_0$ becomes larger as the traffic is heavier. Therefore, introducing different $\tau_0$ may improve the efficiency of the proposed method when dealing with various traffic situations.

5.3 Analysis on the Number of Candidate Routes

As we known, introducing unnecessarily large number of candidate routes may sacrifice memory spaces and increase the computing time, while too small number of candidate routes may not include enough alternative routes to distribute the vehicles. Therefore, simulations with gradually changing $|G|$ are carried out in Fig. 8 in order to find an appropriate number of
The traffic volume of each OD pair is 30 cars/h.

The traffic volume of each OD pair is 60 cars/h.

The traffic volume of each OD pair is 120 cars/h.

Fig. 9 Comparison between the proposed method and shortest path algorithm under different traffic conditions.

Table 2 Parameter setting under different traffic conditions.

<table>
<thead>
<tr>
<th>Traffic volume for each OD pairs</th>
<th>30 cars/h</th>
<th>60 cars/h</th>
<th>120 cars/h</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_0$</td>
<td>0.1</td>
<td>1.0</td>
<td>3.0</td>
</tr>
<tr>
<td>$</td>
<td>G</td>
<td>$</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3 Five congestion risk levels.

<table>
<thead>
<tr>
<th>Risk level</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
<th>Level 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_{ij}/c_{ij}$</td>
<td>[0.1, 0.5]</td>
<td>(0.5, 1.0]</td>
<td>(1.0, 3.0]</td>
<td>(3.0, 5.0]</td>
<td>more than 5.0</td>
</tr>
</tbody>
</table>

The traffic volume of each OD pair and the temperature parameter are set at 120 cars/h and $\tau_0 = 3$, respectively. The average of traveling time $E$ is used for the traffic volume and traveling time convergence.

The result indicates that the average traveling time of $E$ becomes smaller and finally converges as the number of candidate routes $|G|$ increases. Therefore, $|G| = 100$, which is large enough to guarantee the performance, is considered as a proper value for common use in the traffic networks.

5.4 Comparison of the Proposed Method and the Shortest Path Algorithm

In order to evaluate the efficiency of the proposed method when it is implemented in the large scale road network, the comparison between the proposed method and shortest path algorithm is carried out under three traffic conditions. The parameters used for each traffic condition is shown in Table 2.

As we can see from Fig. 9, the performance of two methods is similar with each other when the traffic is not heavy, while the differences becomes larger as the traffic volume of each OD pair in the road network becomes larger. The results of the shortest path algorithm is changing alternately between two values, which indicates that the traffic volume is assigned to the shortest path and consequently the traffic jam is caused, and then turns to former routes after the update of the traveling time in the next iteration, which result in the unexpected low traffic volume on the previous optimal route. Contrarily, the proposed Q value-based Dynamic Programming with Boltzmann Distribution seems more stable and effective.

In order to show the traffic distribution more clearly, we divided the congestion risk of the sections into five levels depending on the proportion of $v_{ij}$ and $c_{ij}$ as shown in Table 3. The average numbers of the sections in each level are calculated and shown in Fig. 10. Sections where $v_{ij}/c_{ij}$ is smaller than 0.1 are considered as very small traffic volumes, and the numbers of these sections are counted in Table 4, since they are too large to
be displayed in Fig. 10.

It could be seen from Fig. 10 that although the differences between two methods are not the same in all the traffic conditions, the number of sections in low congestion risk levels of the proposed method is always larger than the shortest path algorithm, while the number of the sections in high congestion risk levels of the proposed method is much smaller than the shortest path algorithm. It is also shown from Fig. 10c and Table 4 that some sections are involved into level 5 when using the shortest path algorithm, while the number of almost unused sections of the shortest path algorithm is much more than the proposed method. Actually, $\tau_{ij}/c_{ij}$ of the section in the worst condition when using the shortest path algorithm is larger than 8, which means that the traveling time of the section is 615 times longer than that of the free flow traveling time. Contrarily, only a few number of sections are involved into level 3 and level 4 when Q value-based Dynamic Programming with Boltzmann Distribution is adopted.

### 5.5 Computation Time of the Proposed Method

In order to reveal the feasibility of the proposed method in large scale networks, the average computation time of Q value calculation by using Q value-based Dynamic Programming with Boltzmann Distribution is carried out in this subsection. The proposed method has been simulated on six different sizes of maps as shown in Table 5 by using Xenon E5310 (1.60 GHz) with 4 GB RAM. Since the temperature parameters also affect the calculation speed, the computation time of different maps with changing $\tau_0$ is described in Table 6. It is the average computation time for the convergence of Q values, which means the time of one iteration in Fig. 9. $\theta$ is set as 0.2 and $G$ is equal to 100 during the simulations. The calculation time of traditional shortest path Dijkstra algorithm (DA) whose time complexity is $O(n^2)$ and the Q value-based dynamic programming (QDP) is shown in Table 7 for comparison. It is shown from Table 6 that the computation time of Q value-based Dynamic Programming with Boltzmann Distribution is acceptable even if the maps with very large sizes are used.

### 6. Conclusion

In this paper, Q value-based Dynamic Programming with Boltzmann Distribution has been systematically studied in the static system using real road networks – Kitakyushu City’s map. It has been seen from simulations that the proposed method could reduce the congestion and save the traveling cost effectively comparing with the shortest path algorithms. In addition, the selection of the temperature parameter and the number of the candidate routes have been discussed. It has been suggested that introducing appropriate temperature parameters for Q value-based Dynamic Programming with Boltzmann Distribution depending on the traffic conditions could improve the efficiency of the proposed method.

However, as for the applications to the real world Vehicle Navigation Systems, the proposed method still remain immature. It should be enhanced and tested in dynamic traffic systems in the future.

### References


136

SICE JCMSI, Vol. 4, No. 2, March 2011


Shanqing Yu
She received the M.E. degrees from the Graduate School of Information, Production and Systems, Waseda University, Japan and the School of Computer Engineering and Science, Shanghai University, China in 2008. Since October 2008, she has been a Ph.D. candidate at the Graduate School of Information, Production and Systems, Waseda University.

Yelei Xu
He received the B.E. degree in Information and Electronic Engineering from Zhejiang GongShang University, China in 2007. Since September 2007, he has been a student of the Master’s Course at the School of Communication and Information Engineering, Shanghai University, China. Since September 2009, he has been a student of the Master’s Course at the Graduate School of Information, Production and Systems, Waseda University, Japan.

Shingo Mabu (Member)
He received the B.E. and M.E. degrees in Electrical Engineering from Kyushu University, Japan in 2001 and 2003, respectively and the Ph.D. degree from Waseda University, Japan in 2006. Since April 2007, he has been an Assistant Professor in the Graduate School of Information, Production and Systems of Waseda University. He is a member of IEEE.

Manoj Kanta Mainali
He received his bachelor degree from Ritsumeikan Asia Pacific University, Japan in 2004 and the M.E. degree from the Graduate School of Information, Production and Systems, Waseda University, Japan in 2009. Since April 2009, he has been a Ph.D. candidate at the Graduate School of Information, Production and Systems, Waseda University.

Kaoru Shimada (Member)
He received his M. Med. Sci. degree from the University of Tsukuba in 1999 and the Ph.D. degree from the Graduate School of Information, Production and Systems, Waseda University in 2007. Since 2007, he has been a Visiting Lecturer at the Graduate School of Information, Production and Systems, Waseda University. He is a member of the Information Processing Society of Japan and IEEE.

Kotaro Hirasawa (Member)
He received the B.E. and M.E. degrees from Kyushu University, Japan in 1964 and 1966, respectively. From 1966 to 1992, he was with Hitachi Ltd., where he served as a Vice President of Hitachi Research Laboratory. From December 1992 to August 2002, he was a Professor in the Graduate School of Information Science and Electrical Engineering of Kyushu University. Since September 2002, he has been a Professor in the Graduate School of Information, Production and Systems of Waseda University. He is a member of ACM and IEEE.