Filtering Order Adaptation Based on Attractor Selection for Data Broadcasting System

Shinya KITAJIMA[†] Takahiro HARA[†] Tsutomu TERADA[‡] Shojiro NISHIO[†]
[†]Dept. of Multimedia Eng., Grad. School of Information Science and Technology, Osaka University
Email: {kitajima.shinya, hara, nishio}@ist.osaka-u.ac.jp
[‡]Dept. of Electrical and Electronics Eng., Grad. School of Science and Technology, Kobe University
Email: tsutomu@eedept.kobe-u.ac.jp

Abstract

Recent spread of various data broadcasting services leads to provide enormous and various data. Since, data that a client needs are a part of them, there has been an increasing interest in information filtering techniques where a client automatically chooses and stores the necessary data. Generally, when a client performs filtering, it applies some filters sequentially and the time required for filtering changes according to the order of filters. On the other hand, in recent years, there have been many studies about attractor selection which is an autonomous parameter control technique based on the knowledge from living organisms. In this paper, in order to reduce the load for filtering, we propose a novel method which adaptively changes the order of filters according to the change in broadcast contents. This method adaptively decides the control parameters for filtering by using attractor selection.

1 Introduction

Recent spread of various data broadcasting services leads to provide enormous and various data. Therefore, there has been an increasing interest in information filtering techniques. In a broadcast system, while the server can broadcast large amount of data to clients typically mobile terminals at a time, the storage of clients are limited[2, 3, 7]. To solve this problem, information filtering to automatically choose and store necessary data on the clients storage is an effective and promising approach.

Generally, when a client performs filtering, it applies some filters sequentially. However, the time required for filtering changes according to the order of filters, since the number of data items that match each filter and the filtering load are different with each other. When the filtering load is high, filtering speed might become slower than the receiving speed of data. Thus, in an information filtering system, how to determine the order of filters is a crucial problem.

On the other hand, in recent years, there have been several studies about *attractor selection* which is an autonomous parameter control technique based on the knowledge from living organisms[4, 5]. By using attractor selection, the system can control parameters depending on the situation autonomously, thus it can cope with changes of the system environment flexibly.

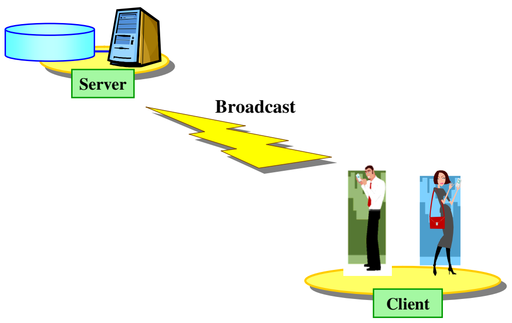


Figure 1. Assumed environment.

In this paper, in order to reduce the load for filtering process, we propose a novel method which adaptively changes the order of filters adapting to the change in broadcast contents. This method adaptively decides the control parameters for filtering by using attractor selection. Furthermore, we show the results of simulation experiments, which we confirm that the proposal method improves the load for filtering process compared with other methods.

The remainder of this paper is organized as follows. Section 2 describes the outline of an information filtering system and Section 3 introduces attractor selection. Section 4 explains our method in details. Section 5 evaluates the performance of our method. Finally, we conclude the paper in Section 6.

2 Information Filtering System

2.1 Mobile Environment

There are several data broadcasting services that have been already available, e.g., those using a surplus band of terrestrial broadcasting, news distributions on the Internet, and bidirectional data services using satellite broadcasting. In such data broadcasting services, the server can send enormous information to a large number of users at a time. However, the data wanted by a user are generally just a small part of the broadcast data.

In this paper, we assume a urban data broadcasting service in town which is thought to be common in the near future. Figure 1 shows a system environment assumed in this paper. In this environment, mobile users equipped with portable devices such as PDAs and smartphones (mobile clients) walk in town, and receive broadcast data via the wireless channel from the nearest server. Some conventional works such as [1] also assume such an information broadcasting system.

Broadcast contents are mainly text data, and the data of various genres such as real time information like news and weather forecast, local store information and event information are broadcast. Since mobile clients have limit for the storage, information filtering to automatically choose the necessary information for users is highly required. We call such system as *information filtering system*.

2.2 Filtering Architecture

In the information filtering system shown in Figure 2, each client stores broadcast data items once into its receiving buffer, then performs filtering operations when the number of the received items reaches the predetermined constant, and stores only the necessary data items on the storage. Here, we show an example of filtering broadcast data. If a user wants to get data on today's news about sports, the system performs filtering operations by using three kinds of filters to get contents whose (i) category is "news", (ii) issue date is today, and (iii) topic is "sports".

There are various kinds of filters. For instance, a filter which gets items that match specified category or keyword, a filter which gets items that are given a timestamp of the particular period of time, a filter which gets items of high relevance by using the cosine correlation between the user's interest and items[6], and so on. The load of applying these filters is different with each other, for instance, the load of calculating the cosine correlation is heavier than that of simple keyword matching.

Moreover, some filters are often applied at the same time as shown in the above example. If the filters do not include ranking operations and do include only selection operations, the order of applying filters does not affect the result of filtering[7].

2.3 Filtering Cost

If there are multiple filters to apply, the order of applying filters influences the *filtering cost* since the number of data items that match each filter and the processing cost of the filter are different among filters. Here, the filtering cost represents the load to perform the filtering operations as a numerical value, and the load to apply a filter is proportional to the processing cost of the filter per data item and the number of data items that are applied the filter.

Figure 3 shows an example that the filtering cost changes according to the order of filters. In this figure, tables (a) and (b) show a case in which there are same five broadcast data

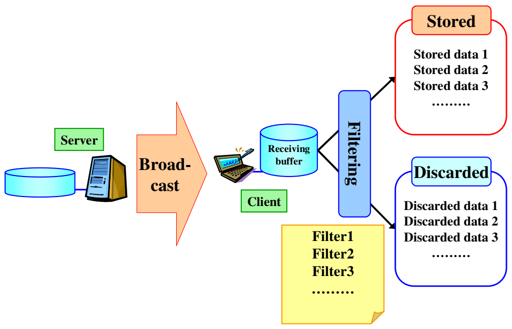
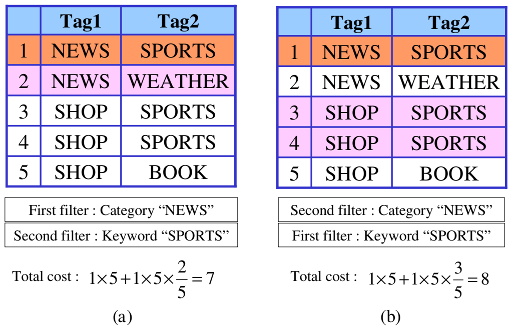


Figure 2. Information filtering system.



	Tag1	Tag2
1	NEWS	SPORTS
2	NEWS	WEATHER
3	SHOP	SPORTS
4	SHOP	SPORTS
5	SHOP	BOOK

First filter : Category "NEWS"
Second filter : Keyword "SPORTS"

Total cost : $1\times 5+1\times 5\times\frac{2}{5}=7$

(a)

	Tag1	Tag2
1	NEWS	SPORTS
2	NEWS	WEATHER
3	SHOP	SPORTS
4	SHOP	SPORTS
5	SHOP	BOOK

Second filter : Category "NEWS"
First filter : Keyword "SPORTS"

Total cost : $1\times 5+1\times 5\times\frac{3}{5}=8$

(b)

Figure 3. Calculation of filtering cost.

items stored in the receiving buffer of a client, but the order of applying filters is different. Here, let us assume that two tags are attached to each data item (Tag1: category, Tag2: keyword) that represent the contents of the item. For instance, item 1 belongs to category "news", and its keyword is "sports".

In table (a), a filter to select data items in category "news" is applied to five items at first, and then another filter to select data items with keyword "sports" is applied to two items that are selected by the first filter. Let us also assume the processing cost of both filters is 1. In this case, the total cost becomes 7. On the other hand, in table (b), the keyword filter ("sports") is applied to the five items at first, and then the category filter ("news") is applied to three items that are selected by the keyword filter. In this case, the total cost becomes 8. In this way, the filtering cost changes according to the number of data items that match each filter and the processing cost of the filter.

If the processing speed of the filters becomes lower than the receiving speed, the receiving buffer overflows since data are continuously broadcast. In addition, users use their mobile terminals not only to receive the broadcast data but also for other services such as a navigation tool and VOD. Therefore, the filtering cost should be as small as possible.

In our assumed environment, there are various broadcast data whose contents dynamically change, such as real time information like news, weather, and local store and event information. Moreover, users' demand also dynamically changes. In such an environment, a method that can adap-

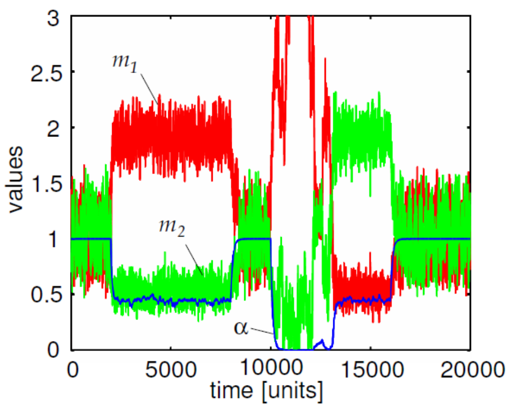


Figure 4. Response of the double feedback loop.

tively and dynamically decide the order of filters is needed.

3 Attractor Selection

3.1 Adaptive Response by Attractor Selection

In this sub section, we describe the outline of the attractor selection mechanism, which has been proposed in [4]. In [4], the authors claim that organisms form a complicated network system having the networks of many hierarchies such as gene, protein, and metabolism which they call the *organism networks*. When different organism networks meet together, they reach to a stable state (*attractor*) while changing their structure and route, and form a organism symbiosis network. This contains many properties such as expansibility, autonomy, toughness, flexibility, adaptability, and variety, which are also needed in an information network. Here, "symbiosis" means that different organisms two or more kinds interact with each other, and live by supplying the properties to others which they do not have mutually.

It is necessary to adapt to a new environment flexibly while two kinds of organisms without having met before process to form symbiosis relations. However, they have not experienced this environment change in the past, so that they cannot prepare for a hereditary program corresponding to it.

By conventional studies, it becomes clear that transition from an original stable state to a new stable state by the reorganization of the gene metabolism network (three classes of networks of gene, protein, and metabolism) and interaction between the cells by the chemical substance are important. Based on this, the authors of [4] suggest a new mechanism called the adaptive response by attractor selection.

In [4], to represent a complicated gene metabolism network simply, a model having double feedback loops is defined as follows:

$$\frac{dm_1}{dt} = \frac{syn(act)}{1+m_2^2} - deg(act)\cdot m_1 + \eta_1, \quad (1)$$

$$\frac{dm_2}{dt} = \frac{syn(act)}{1+m_1^2} - deg(act)\cdot m_2 + \eta_2, \quad (2)$$

$$syn(act) = \frac{6act}{2+act}, \quad (3)$$

$$deg(act) = act. \quad (4)$$

m_1 and m_2 are mRNA densities made by operons 1 and 2, where an operon is one of the functional units existing on a genome. η_1 and η_2 in the third term on the right side in equations (1) and (2) are noises. The activity act changes according to the following equation:

$$\frac{dact}{dt} = \frac{pro}{\left(\left(\frac{Nut\_th_1}{m_1+Nut_1}\right)^{n_1}+1\right)\times\left(\left(\frac{Nut\_th_2}{m_2+Nut_2}\right)^{n_2}+1\right)} - cons\times act. \quad (5)$$

Here, Nut_1 and Nut_2 are the supply densities of nourishment from the outside for operons 1 and 2; and Nut_th_1 and Nut_th_2 are their thresholds. pro and $cons$ are coefficients of production and consumption of the activity, and n_1 and n_2 are appropriate constant numbers.

Figure 4 shows the responses of these double feedback loops. It can be seen that when the outside supply for one nourishment is cut, the attractor making up for the lack is selected. There are two absorption domains in this environment, but only an appropriate attractor is selected. This is because the fluctuation by noises becomes large when the environment becomes worse and activity act becomes low, and then, the system is absorbed by that attractor while it approaches to the attractor and recovers the activity. This behavior is an environmental adaptation by attractor selection.

3.2 Advantage of Attractor Selection

The fully premeditated construction of systems has become impossible with the rapid large-scaling and complexifying of recent information systems. Therefore, in recent years, development of system management techniques to adapt for changes of the environment flexibly and autonomously has become a crucial issue.

For instance, in the field of network design, conventional systems have been fully designed to optimize performance and efficiency for specific (predictable) situations. However, such an approach does not work well in recent complicated systems, because it takes long time or sometimes impossible to recover from a large-scale network failure, especially, unknown type of failures. Therefore, to cope with unpredictable changes of the system environment, e.g., system failures and change of system inputs, another design approach is required, in which each component in the system behaves to maintain a stable state and adapts for the environmental change flexibly and autonomously. By this approach, the system can maintain a stable state and offer a service of good quality in highly dynamic environment, although the performance may not be optimized.

As the system becomes large-scale and more complicated, it becomes impractical to know in advance all events happened in the system and their factors, e.g., reasons and how to deal with them.

Fuzzy reasoningaccumulates human knowledge as a knowledge database with *If-Then-Else* rules, and performs recognition, control, and reasoning by the reasoning engine. While it is based on human experiences, it still requires the advanced construction of rules, thus, the above mentioned problem is not solved.

Neural networksperform pattern recognition by optimizing parameters in neurons. However, since they are based on learning machine, they cannot adapt for a situation that has not ever met and is hard to be predicted.

On the other hand, a genetic algorithmconverts engineering data into the form of gene code, and optimizes the system by imitating heredity processes that occur in organisms such as mutation, recombination, and optimal choice. Since it has both two aspects of random search and optimal choice, it is different from approaches that cannot get away from the local minimum such as the steepest descent methodHowever, it basically assumes well formulated system, thus, it cannot handle unknown changes occurred in the system.

Simulated annealingis an approach that examines multiple neighboring solutions of the current solution randomly, and decides probabilistically which neighboring state to transit. It also has a mechanism to prevent from falling into the local minimum. However, since it searches for the optimal solution heuristically, it generally takes much time to converge, and cannot cope with frequent changes of the environment.

As mentioned above, conventional approaches cannot fully cope with unknown changes in the system flexibly and quickly.

On the other hand, attractor selection has advantages that it can cope with unknown changes and its calculation time is much shorter than conventional heuristic approaches. Since the contents of broadcast data are continuously changing in our assumed system environment, attractor selection is suitable for the problem of determining the order of filters which we address in this paper.

4 Proposed Method

In this section, we propose a method that can adaptively follow the change of broadcast contents and reduce the filtering cost by using attractor selection to control the parameters to decide the order of filters.

In the proposal method, a client determines *filter selection priority* S by using attractor selection, where S consists of a list of filters and defines the order of applying the filters. Here, let us denote n as the number of applying filters, and $S_{i,j}$ as the filter selection priority of applying filter F_i $(i = 1, 2, \ldots, n)$ at j-th position $(j = 1, 2, \ldots, n)$ in S.

In the following, we describe the proposed method in detail.

4.1 Applying Attractor Selection

Calculation of the Filtering Cost We define the ratio of data items which are discarded by applying filter F_i as *decrease ratio* D_i. D_i is calculated by the following equation using the number of data items, d_i, discarded by F_i, and the number of data items, a_i, applied F_i:

$$D_i = \frac{d_i}{a_i}. \tag{6}$$

When a client uses our proposal method in a real environment, it actually applies the filters to the broadcast data items, and uses the elapsed time to perform filtering as the filtering cost. However, in our simulation evaluation, a client cannot apply filters actually since we use pseudo data. Thus, we generalize the filtering cost and define the processing cost of each filter F_j as c_j. c_j represents the processing time per data item when a client applies F_j.

The total cost C for applying n kinds of filters in a certain order to N data items is calculated by the following equation:

$$C = N \sum_{j=1}^{n} \left(c_j \prod_{k=1}^{j-1} D_k \right). \tag{7}$$

Calculation of the Activity In the proposed method, the activity α is defined by using C, since the system performance is considered better when the filtering cost is lower. Here, the minimum value of C among the last x results of filtering is denoted by C_{min}. Then, the activity α is calculated by the following equation, where the activity becomes higher when the filtering cost approaches the minimum cost:

$$\frac{d\alpha}{dt} = \delta \left(\left(\frac{C_{min}}{C} \right)^{\lambda} - \alpha \right). \tag{8}$$

Here, δ and λ are scale factors to control the adaptation rate and the value of activity. Note that α ranges $0 \leq \alpha \leq 1$.

Calculation of the Selection Priority The selection priority $S_{i,j}$ is defined by the following equation, which comes from the approaches in [5]:

$$\begin{aligned}
\frac{d}{dt} S_{i,j} &= \frac{syn(\alpha)}{1 + S_{max,j}^2 - S_{i,j}^2} - deg(\alpha) S_{i,j} + \eta_{i,j}, & (9) \\
syn(\alpha) &= \alpha \left[\beta \alpha^{\gamma} + \phi^* \right], & (10) \\
deg(\alpha) &= \alpha, & (11) \\
\phi(\alpha) &= \frac{syn(\alpha)}{deg(\alpha)}, & (12) \\
\phi^* &= \frac{1}{\sqrt{2}}. & (13)
\end{aligned}$$

Here, $\eta_{i,j}$ is a random number, and β and γ are scale factors (constants). $S_{i,j}$ ranges $0 \leq S_{i,j}$. Note that for $j \geq 2$, $S_{i,j}$ is set as 0 if F_i is already selected, since it is meaningless to apply the same filter more than once.

When the filtering cost is low and the activity is high, the selection priority rarely changes since the first term on the

Table 1. Parameters.

Parameter	Value
Number of times of filtering	50000
Number of filters	5
Number of tags	5
Number of keywords	5
Size of the receiving buffer	5000
Calculation cycle in the optimal method	10000
Calculation cycle in the genetic algorithm	7000
Number of steps in the Runge-Kutta method	10
Band-width of broadcast[Mbps]	10
Size of a data item[KByte]	1

right side in equation (9) is dominant. However, when the filtering cost becomes high and the activity becomes low, the third term, i.e., noise, becomes dominant, and the system tries to another stable state. That is, the system adapts to the change of the broadcast contents.

4.2 Flow Chart

In the proposed method, a client performs following steps every time when N data items are stored in its receiving buffer.

1. Calculate the current activity α using equation (8).
2. Set the union of the selected filters U_f as $U_f = \emptyset$.
3. Perform following steps from (a) to (c) continuously for $j = 1, 2, \ldots, n$.
   (a) Calculate the selection priority $S_{i,j}$ using equation (9) for $i = 1, 2, \ldots, n$ $(F_i \notin U_f)$.
   (b) Find max_i whose $S_{max\_i,j}$ is the maximum among all $S_{i,j}$ calculated in step (a).
   (c) Select F_{max_i} as j-th filter, and add F_{max_i} to U_f.
4. Perform the filtering operations according to the order of filters determined above.
5. Calculate the filtering cost using equation (7).
6. Update C_{min} if necessary.

We define one cycle of the above steps as a unit of filtering.

5 Evaluation

This section evaluates the proposed method using simulation studies. The evaluation criterion is the *average filtering cost*, which is the average of the filtering costs for all filtering processes performed during the simulation time.

5.1 Simulation Environment

In the simulations, the broadcast data and the filtering model are assumed as an information service for mobile clients as described in Section 2. Moreover, we did not use real broadcast data but use pseudo data to represent various situations by changing parameters. To apply multiple different filters, we attach the same number of attributes as filters to the pseudo data and the attribute values (keywords) are used for filtering.

For simplicity, all filters perform selection operations, i.e., only data items that contain keywords specified by the client are stored, and other items are discarded. Thus, the filtering results do not depend on the order of applying filters[7]. The results of our simulations correspond to the results in other settings where filters whose applying order does not affect the filtering results are assumed. We can also easily consider cases where filters whose applying order does affect the filtering result by introducing a mechanism to take dependencies of the applying order into account when determining the order of filters. By setting various values for processing cost c_i of filter F_i $(i = 1, 2, \ldots, 5)$, we can simulate various cases in real environments.

The distribution of keywords attached to the pseudo data is determined according to the Zipf distribution as shown in the following equation:

$$f(k) = \frac{\frac{1}{r_k}}{\sum_{m=1}^{N_a} \frac{1}{m}}. \quad (14)$$

f_k represents the probability that the keyword k is chosen for the attribute in each data item. Here, N_a denotes the number of possible keywords, and r_k represents the rank of k. The larger $f(k)$ is, the higher probability that k is contained in each attribute is.

Table 1 shows the parameters used in the simulations. In the simulations, the rank of each keyword periodically changes randomly. By changing the ranks of keywords, we can simulate the change of the contents of the broadcast data. We call the cycle for changing the rank of keywords as the *keyword-rank-changing cycle*. We change the keyword-rank-changing cycle between 600 to 1400.

For simplicity, for all filters, we basically set the processing cost c_i as 0.6 (millisecond). If we change c_i for each filter, we can simulate arbitrary combination of filters with different loads such as combination of simple keyword matching and cosine correlation. In the simulations, we basically set $\beta = 0.4$, $\gamma = 5.0$, $\delta = 3.0$, $-1.0 \leq \eta \leq 1.0$, and $\lambda = 10$ determined by some preliminary experiments. Furthermore, we set C_{min} as the minimum value of the past 50 times of filtering.

5.2 Comparison Methods

In our simulations, we compared our proposed method with the following four methods.

Minimum Cost Method: In the minimum cost method, a client calculates the filtering cost for each of all possible $n!$ kinds of orders of filters for every filtering process, and adopts the order that gives the minimum cost among them. Note that this method is unrealistic because the computation load is too high to apply it in a real environment. Therefore, we show the performance of this method as a lower bound.

Cyclic Adaptation Method: At a certain calculation cycle, a client calculates the filtering cost for each of all possible $n!$ kinds of orders of filters for the filtering process at the cycle, and adopts the order that gives the minimum cost among them. After this, the client adopts the same order of filters until the next calculation cycle. In this method,

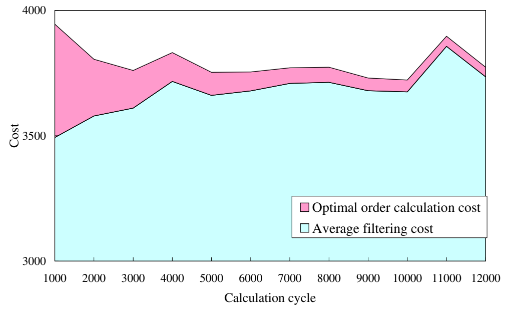


Figure 5. Impact of calculation cycle in the cyclic adaptation method.

the filtering cost at the calculation cycle becomes equal to that of the minimum cost method, but cannot adapt to the change of broadcast contents until the next cycle.

Note that the load at the calculation cycle is high, since the client has to calculate the costs of all $n!$ kinds of orders. We define the value obtained by dividing the filtering cost to perform this method once by the calculation cycle as the *optimal order calculation cost*. Moreover, we call the sum of the average filtering cost and the optimal order calculation cost as the *total cost*.

Genetic Algorithm: In the genetic algorithm, a client periodically searches an appropriate order of filters according to the genetic algorithm so that the filtering cost becomes less.

Random Method: In the random method, a client decides the order of filters at random for every filtering process.

5.3 Simulation Results

Impact of Calculation Cycle in the Cyclic Adaptation Method Fig 5 shows the average filtering cost, the optimal order calculation cost, and the total cost of the cyclic adaptation method when the calculation cycle changes from 1000 to 12000.

The result shows that the average filtering cost when the calculation cycle is 1000 is the lowest. This is because the calculation cycle is short, thus, the server can correspond to the change of broadcast contents rapidly. The average filtering cost basically becomes higher as the calculation cycle gets longer.

On the other hand, the optimal order calculation cost becomes lower as the calculation cycle gets higher. This is because the longer the calculation cycle is, the less the number of times of calculating the costs is.

The total cost, the sum of the average filtering cost and the optimal order calculation cost, is the lowest when the calculation cycle is 10000. This shows that the average filtering cost and the optimal order calculation cost have a trade-off relation, and the system performance is balanced when the calculation cycle is 10000 in this simulation environment. Thus, we chose 10000 as the calculation cycle in the cyclic adaptation method in the following experiments.

Table 2. Comparison between the proposed method and the other methods.

Method	Filtering cost [msec]
Minimum cost method	3343.85
Cyclic adaptation method	3806.26
Cyclic adaptation method (total cost)	3849.39
Genetic algorithm	3898.12
Genetic algorithm (total cost)	3920.73
Proposed method	3464.46
Random method	3824.83

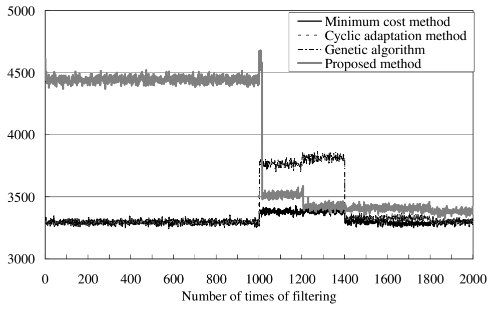


Figure 6. Transition of the filtering cost.

Comparison among the Four Methods Table 2 shows the average filtering costs of the proposed method, the minimum cost method, the cyclic adaptation method, the genetic algorithm, and the random method.

From this result, the average filtering cost of the proposed method is lower than that of the cyclic adaptation method, the genetic algorithm, and the random method. The average filtering cost of the cyclic adaptation method is slightly lower than the random method. However, the cyclic adaptation method requires an extra cost to calculate the order of filters, i.e., the optimal order calculation cost, thus, the total cost is higher than the cost of the random method.

Moreover, the average filtering cost of the genetic algorithm is higher than that of the cyclic adaptation method. In the genetic algorithm, the optimal order calculation cost at the calculation cycle is lower than that in the cyclic adaptation method. However, the filtering cost of filters whose order is determined in the calculation cycle is not always minimum. Thus, the average filtering cost of the genetic algorithm becomes higher than that of the cyclic adaptation method.

Figure 6 shows the transition of the filtering costs of the four methods. Figures 7 and 8 show the transitions of the activity and the selection priority $S_{i,1} (i = 1, 2, \dots, 5)$ in the proposed method, respectively. Due to the limitation of space, we only show the results from the simulation starting time to the time until 2000 times of filtering processes are performed.

Figure 6 shows that the filtering cost of every method except for the minimum cost method changes largely after the time when 1000th filtering process is performed at which

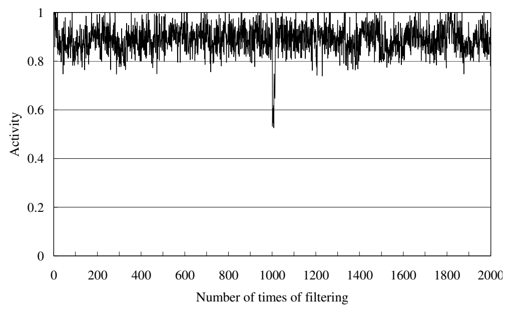


Figure 7. Transition of the activity.

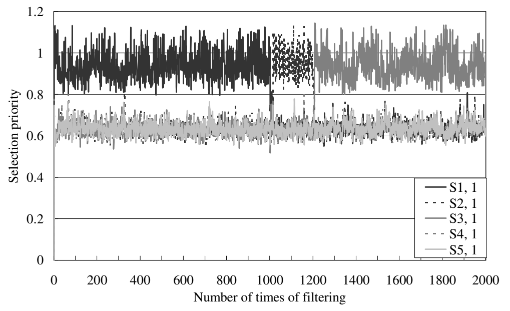


Figure 8. Transition of the selection priority.

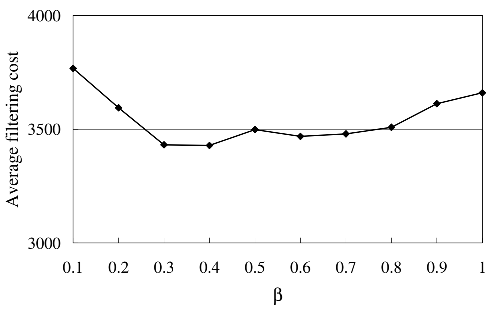


Figure 9. Impact of β

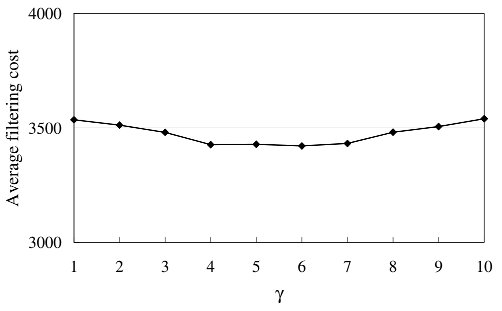


Figure 10. Impact of γ

the broadcast contents change. However, in the proposed method, the filtering cost becomes low soon, which shows that our method can adopt to the change of broadcast contents.

Figure 7 shows that the activity becomes very low when the broadcast contents change and the filtering cost becomes high. Moreover, Figure 8 shows that when the activity becomes low, the random term in equation (9) influences largely, thus, the selection priority goes up and down greatly. After a short time, the system transits into a stable state, and the selection priority of a specific filter rises.

In summary, in the proposed method, the client changes the order of filters adaptively by using attractor selection to control the selection priority when the broadcast contents change and filtering cost becomes high. It confirms us the effectiveness of using attractor selection to adapt to the change of the broadcast contents. Here, in the proposed method, the activity sometimes becomes low and the order of filters is changed even when broadcast contents do not change. This is due to the influence of the random term in equation (9).

Impact of β in the Proposed Method Figure 9 shows the average filtering cost of the proposed method when β changes from 0.1 to 1.0.

From the result, the average filtering cost is low when $\beta = 0.3$ to 0.8. Here, β is a constant that coordinates the influence of the random term in equation (9). When β is very high, the random term influences little, so that the selection priority and the order of filters do not change even when the filtering cost is high and the activity is low. On the other hand, when β is very low, the random term influences largely, and our method acts similar to the random method.

Impact of γ in the Proposed Method Figure 10 shows the average filtering cost of the proposed method when γ changes from 1 to 10.

From this result, the average filtering cost is the low when $\gamma = 4$ to 7. Here, γ is a constant that coordinates the influence of the activity in equation (9). The influence of the activity becomes small when γ is high. However, in our simulations, the average filtering cost is not much influenced by γ.

Impact of δ in the Proposed Method Figure 11 shows the average filtering cost of the proposed method when δ changes from 1 to 7.

The result confirms that the average filtering cost hardly changes even if δ changes, i.e., the impact of δ is very small.

Impact of x in the Proposed Method Figure 12 shows the average filtering cost of the proposed method when x changes from 10 to 100. Here, x is the window size for calculating C_{min}, i.e., our method determines C_{min} as the minimum filtering cost among the last x filtering processes.

From the result, the average filtering cost is low when $x = 30$ to 100. If x is very small, C_{min} is updated frequently, and the activity tends to be unstable according to equation (8). On the other hand, if x is large, C_{min} is rarely updated even when the broadcast contents change, thus, the activity also tends to be unstable.

Impact of c_i Finally, we change c_i as shown in Table 3 to examine the performance of our method where each filter

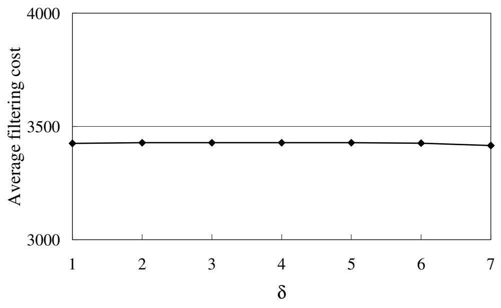


Figure 11. Impact of δ

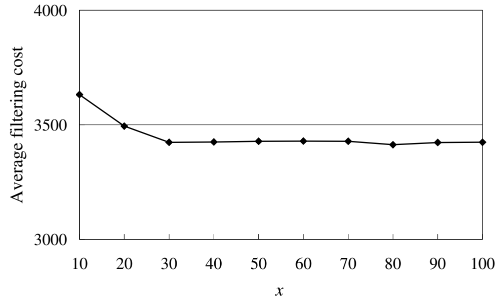


Figure 12. Impact of x

has different processing cost c_i. Table 4 shows the average filtering costs of the proposed method, the minimum cost method, the cyclic adaptation method, the genetic algorithm, and the random method. Note that λ is set as 4, since the average filtering cost in the proposed method is the lowest when $\lambda = 4$ by our preliminary experiments.

This result shows that the average filtering cost of the proposed method is lower than that of the cyclic adaptation method, the genetic algorithm, and the random method even when each filter has different processing cost. The average filtering costs of all methods are lower than those of in Table 2, since the minimum value of c_i is lower.

If all filters have different processing costs, the ratio of C to C_{min} becomes large, since the filtering cost changes largely according to the order of filters and broadcast contents. Therefore, λ should be set a small value to improve the performance of the proposed method. We are currently examining a method to appropriately change λ according to the processing costs of filters. The basic idea of this method is to normalize the ratio of C to C_{min}.

6 Conclusions

In this paper, we proposed a new method that uses attractor selection to control parameters to determine the order of applying filters. With the proposed method, the client adaptively changes the order of filters following the change of broadcast contents to reduce the filtering load. The simulation results confirmed that the proposed method reduced the filtering cost compared with the other methods except for the minimum cost method (lower bound).

Table 3. Change of c_i.

Filter	Processing cost [msec]
c_1	0.6
c_2	0.8
c_3	1.0
c_4	1.2
c_5	1.4

Table 4. Impact of c_i.

Method	Filtering cost [msec]
Minimum cost method	4008.83
Cyclic adaptation method	4836.02
Cyclic adaptation method (total cost)	4851.07
Genetic algorithm	5939.81
Genetic algorithm (total cost)	5971.16
Proposed method	4325.59
Random method	6211.15

As part of our future work, we plan to examine the influence of the change of broadcast contents on the performance of our method in more detail.

Acknowledgments: This research was partially supported by "Global COE (Centers of Excellence) Program" and Special Coordination Funds for Promoting Science and Technology "Formation of Innovation Center for Fusion of Advanced Technologies: Yuragi Project" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] Acharya, S., Alonso, R., Franklin, M., and Zdonik, S.: Broadcast Disks: Data Management for Asymmetric Communication Environments, *Proc. ACM SIGMOD 1995*, pp.199–210 (1995).

[2] Belkin, N. J. and Croft, W. B.: Information filtering and information retrieval: Two sides of the same coin?, *Communications of the ACM*, Vol. 35, No. 12, pp. 29–38, Dec. 1992.

[3] Bell, T. A. H. and Moffat, A.: The design of a high performance information filtering system, *Proc. SIGIR 1996*, pp. 12–20, Aug. 1996.

[4] Kashiwagi, A., Urabe, I., Kaneko, K., and Yomo, T.: Adaptive response of a gene network to environmental changes by fitness-induced attractor selection. *PLoS ONE*, Vol. 1, No. 1, e49, Dec. 2006.

[5] Leibnitz, K., Wakamiya, N., and Murata, M.: Biologically inspired adaptive multi-path routing in overlay networks. *Proc. of IEEE SelfMan 2005*, (CD-ROM) May 2005.

[6] Salton, G. and McGill, M. J.: Introduction to modern information retrieval, *McGraw-Hill Book Co.*, Sept. 1983.

[7] Sawai, R., Tsukamoto, M., Terada, T., and Nishio, S.: Composition order of filtering functions for information filtering, *Proc. of ICMU 2004*, pp. 166-171, Jan. 2004.