

Research Article

Parallel Evolutionary Peer-to-Peer Networking in Realistic Environments

Kei Ohnishi

Graduate School of Computer Science and System Engineering, Kyushu Institute of Technology, Kawazu 680-4, Iizuka, Fukuoka 820-8502, Japan

Correspondence should be addressed to Kei Ohnishi; ohnishi@cse.kyutech.ac.jp

Received 6 July 2016; Accepted 28 December 2016; Published 26 January 2017

Academic Editor: Shyi-Ming Chen

Copyright © 2017 Kei Ohnishi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the present paper we first conduct simulations of the parallel evolutionary peer-to-peer (P2P) networking technique (referred to as P-EP2P) that we previously proposed using models of realistic environments to examine if P-EP2P is practical. Environments are here represented by what users have and want in the network, and P-EP2P adapts the P2P network topologies to the present environment in an evolutionary manner. The simulation results show that P-EP2P is hard to adapt the network topologies to some realistic environments. Then, based on the discussions of the results, we propose a strategy for better adaptability of P-EP2P to the realistic environments. The strategy first judges if evolutionary adaptation of the network topologies is likely to occur in the present environment, and if it judges so, it actually tries to achieve evolutionary adaptation of the network topologies. Otherwise, it brings random change to the network topologies. The simulation results indicate that P-EP2P with the proposed strategy can better adapt the network topologies to the realistic environments. The main contribution of the study is to present such a promising way to realize an evolvable network in which the evolution direction is given by users.

1. Introduction

The recent growth of information and communication technologies, for example, the progress of communication speed, storage capacity, and processing speed, is remarkable. Under this quantitative technological progress, the degree of dependence of humans on networks becomes much higher. For instance, the number of users of the Internet in the world reaches around three thousand and two hundred million in 2015, while the number in 2000 is just around four hundred million. While the quantitative technological progress is being made continuously, the quantitative characteristics of humans as a leading part of the network society, such as calculation speed, have not changed so far and will not change in future basically. Thus, the difference in quantitative characteristics between humans and networks is becoming larger, so that it would be needed for sustainable growth of the network society to consider the relationship between humans and networks deeply.

One technique that considers the relationship between humans and computers, between which quantitative

difference is also becoming larger, is interactive evolutionary computation [1, 2]. Interactive evolutionary computation combines a computer, which executes fixed procedures quickly, and a human, which judges things not quickly but from various points of view. Concretely, this technique uses a computer for quickly executing an evolutionary algorithm (referred to as EA hereinafter) as an optimization method and also uses a human as an objective function of an optimization problem to be solved. This technique enables us to optimize parameters of a system whose outputs can be evaluated only by humans. This is one form of cooperation between humans and a computer. Meanwhile, there are few techniques considering a relationship between humans and a network at this moment.

The cooperation between a network and humans indicates that interactions between a network and its users make a network service on the network better for the users. To realize such a cooperation, in the first place, a network implementing a service is needed to be able to interact with users. The interaction between a network and users indicates that network's behaviors influence humans' ones and vice

versa. One of networks that are capable of such interactions is a peer-to-peer (P2P) network [3]. More precisely, an unstructured P2P network is the one. An unstructured P2P network is logically built on a physical network such as the Internet, and in an unstructured P2P network, nodes can be both servers and clients and provide services for each other with direct connection between them. In some applications of unstructured P2P networks such as file sharing, nodes can basically be regarded as users and logical links between nodes equivalent to users are determined by users' own will. That is to say, an unstructured P2P network topology is formed logically and freely by users. If an unstructured P2P network could receive demands from its users and change how to link between nodes (users) based on the users' demands, such an unstructured P2P network would be one form that realizes our desired cooperation between a network and humans.

In our previous study, we thought that for sustainable growth of the network society, a network adapting to demands of humans who are a leading part of the network society is needed. That is to say, we thought that the above-mentioned cooperation between a network and humans is a must for sustainable growth of the network society. The study in the present paper is also based on the same thought. Then, in our previous study, we proposed the evolutionary P2P networking technique that evolutionarily reconstructs topologies of a P2P network based on fitnesses given by nodes (users) in an on-line manner, which is called EP2P hereinafter [4].

However, the basic evaluation of EP2P in [4] did not consider a large size of network as the real P2P networks. For example, according to the investigation by some company in 2014, the accumulative number of users of BitTorrent, which is one of the representative P2P networks for file sharing, for a month was around three hundred million. Since EP2P needs a special node called a super node that plays a role of collecting fitnesses from nodes (users) and executing EA to adaptively change the P2P network topologies, it can happen that the super node is overloaded as the number of nodes increases and then the P2P network stops working due to that. That is the problem of EP2P. Although we did not estimate the exact load of the super node caused by the increasing nodes in our previous study [4], this problem is basically equal to the risk of a single point of failure and we have to solve it.

So, as a solution of the problem of EP2P, we then proposed the parallel evolutionary P2P networking technique, which is called P-EP2P hereinafter, that first divides an entire P2P network into several smaller networks to avoid the overload of a super node and then applies EP2P to each of the small networks to make the entire network adaptive [5]. However, even P-EP2P has a problem. The problem is that search failure rate becomes higher as the number of divided node groups increases. More precisely, in P-EP2P, timing of evaluating and reconstructing network topologies is the same for all node groups, and the simultaneous evaluation and topology reconstruction causes inefficient adaptation of network topologies to users' demands.

Then, as a solution of the problem, we further proposed a new method for evaluating network topologies in P-EP2P. The new method makes all node groups evaluate their

network topologies not simultaneously but sequentially. In the study, it was shown through simulations that P-EP2P using the new method yields better search failure rate than P-EP2P using the previous method when the number of node groups becomes larger [6].

However, in a series of our above-mentioned studies [4–6], we considered only a special evaluation scenario. The special scenario is that all nodes look for only one particular node. To examine if P-EP2P is practical, we need to evaluate our proposed techniques and methods using more realistic evaluation scenarios.

For the purpose of examining the practicality of P-EP2P, in the present paper, we first make more realistic evaluation scenarios for P-EP2P and evaluate a few types of P-EP2P that we proposed in our previous studies using the realistic evaluation scenarios made. The realistic evaluation scenarios assume realistic environments in which various users in terms of what they have and want in the network exist. Then, based on the discussions of the first evaluation results, we propose a new strategy for P-EP2P to better adapt network topologies to the realistic environments. Finally, we evaluate P-EP2P with the new strategy to show the practicality of P-EP2P. In fact, the first evaluation results show that P-EP2P cannot evolutionarily adapt network topologies to some realistic environments. For those environments, random construction of network topologies is considered to be suitable. Therefore, the new strategy that we propose in the present paper first judges if evolutionary adaptation of the network topologies is likely to occur in the present environment, and if it judges so, it actually tries to achieve evolutionary adaptation of the network topologies. Otherwise, it brings random change to the network topologies. The simulation results of P-EP2P including the strategy indicate that it can better adapt the network topologies to the realistic environments.

The main contribution of the present study is to present a promising way to realize an evolvable network in which the evolution direction is given by humans (nodes of an unstructured P2P network), that is, P-EP2P with the proposed strategy. Since there is no alternative way for comparison with P-EP2P at this moment, we just evaluate P-EP2P with the new strategy by comparing it with variants of P-EP2P. However, P-EP2P with the new strategy is shown to be the most practical among them through simulations, and we can expect from the results that P-EP2P with the new strategy works well even in the real world.

The remainder of the present paper is organized as follows. In Section 2, we describe P-EP2P that is a target to be evaluated and to be improved, which we previously proposed. Section 3 explains two types of methods for evaluating and reconstructing network topologies that we previously proposed. Section 4 first describes realistic evaluation scenarios that we design and then shows the simulation results of evaluation of P-EP2P with the methods explained in Section 3 using the realistic evaluation scenarios made. In Section 5, based on the simulation results and discussions in Section 4, we propose a new strategy for P-EP2P and evaluate P-EP2P including the new strategy with an evaluation scenario assuming a dynamic environment as well as with the same simulation scenarios as in Section 4. Section 6

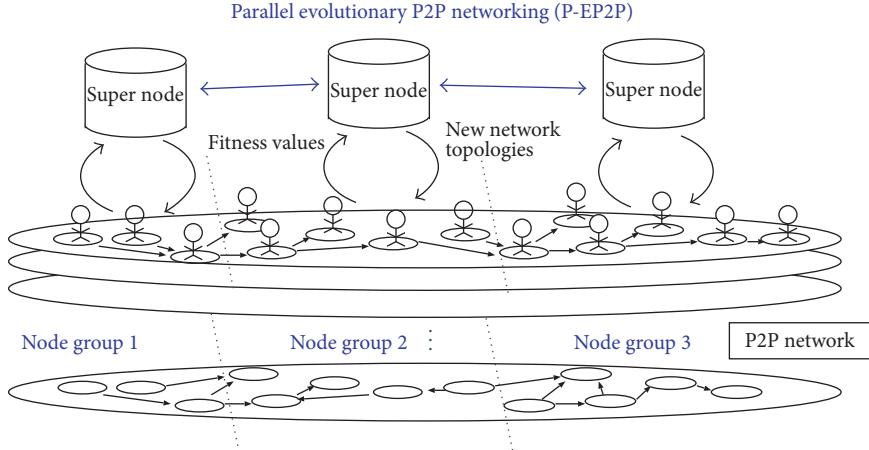


FIGURE 1: Overview of the parallel evolutionary P2P networking technique.

describes related work and conclusions are presented in Section 7.

2. Parallel Evolutionary P2P Networking Technique

The parallel evolutionary P2P networking technique, P-EP2P, divides all of nodes into multiple node groups and then applies the evolutionary P2P networking technique, EP2P, to each group. The overview of P-EP2P is shown in Figure 1. The method for evaluating and reconstructing network topologies explained in Section 3 is related to a timing at which network topologies are evaluated and reconstructed when applying EP2P to each node group.

2.1. Network Composition. As shown in Figure 1, a network using P-EP2P is composed of a P2P network that includes several network topologies, in which all of the nodes are included at the same time, and multiple super nodes, in which EA is used to optimize the topologies. P-EP2P first divides an entire network into several smaller networks. Let N_G be the number of node groups, which are obtained by dividing the entire network, and G_k be the number of nodes in the k th node group ($k = 1, 2, \dots, N_G$). Then, one super node is assigned to each node group.

The actual role of the super node is (1) to determine links for a node in its node group that joins the network for the first time, (2) to reconstruct the network topologies of its node group by executing the EA, and (3) to manage which nodes in its node group join the network at each moment. The super node does not manage which services each node in its node group can provide to other nodes. For example, in a P2P file-sharing network, the super node does not manage which files each node holds. However, the P2P nodes communicate their joining and leaving the network to the super node. Thus, the super node can determine which P2P nodes in its node group have joined the network and whether these nodes are currently in the network.

2.2. Joining and Leaving Nodes. The P2P node communicates its joining and leaving of the network to the super node of its node group. Thus, the super node can determine which P2P nodes in its node group have joined the network and whether these nodes are still in the network.

When a node joins the P2P network for the first time, the super node randomly determines the nodes to which the joining node will link from among all of the P2P nodes present in the node group at that moment. Since there are several network topologies, the super node determines the links for all of the topologies for the joining node. Next, when the joining node leaves the network, the joining node first informs the super node that it will leave the network and then informs the super node when it rejoins the network. However, the target nodes linked to the node are the same as those to which the node was linked before leaving the network, although the target nodes may no longer be in the network.

2.3. Fitnesses Assigned by Nodes. In P-EP2P, nodes give fitnesses to network topologies. The fitness is not explicit evaluation from a user oneself. However, it is herein assumed that network topologies achieving reliable searches are useful, and under this assumption, nodes that are considered to be equivalent to users automatically provide fitnesses for network topologies. Even in this case, users can be considered to be an objective function of the evolutionary P2P networking problem.

A P2P node in each node group uses all network topologies to which it is included for time period T and then assigns a fitness to each of the topologies. The fitness of each network topology is set to zero initially and at every time interval T . Otherwise, each network topology basically increases the fitness by being used by the nodes.

When a P2P node searches the P2P network for P2P nodes that can provide the desired service, this P2P node uses all of the P2P network topologies in which it is included for the search. Therefore, it is possible that within a given allowed number of hops, H_{\max} , the P2P node can find the desired service in some topologies while not being able to find the

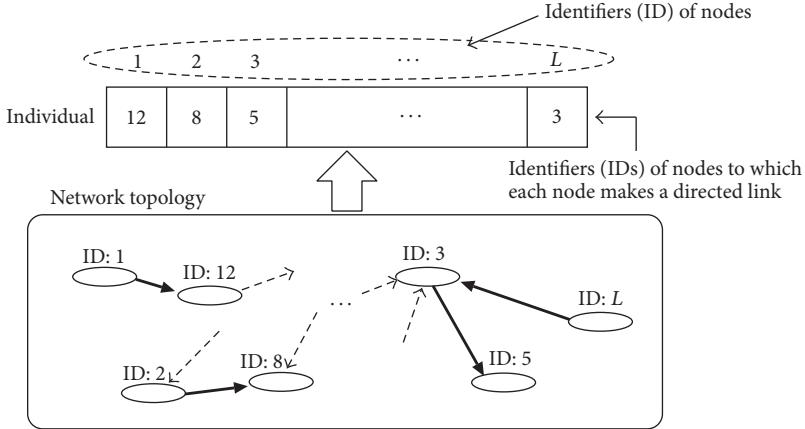


FIGURE 2: Representation of a P2P network topology in the EA (EA individual) when $N_C = 1$.

service in other topologies. If the desired service is not found in a certain topology, the fitness of the topology is increased by one. Otherwise, the fitness does not change.

If the above-mentioned search and assignment of fitnesses are conducted for a period of time T , each topology will be assigned a certain fitness in each node group. Then, the topologies with smaller fitnesses are regarded as better in the EA used herein.

In P-EP2P, one super node is assigned to each of N_G node groups and gathers fitnesses only from G_k ($k = 1, 2, \dots, N_G$) nodes that belong to its node group. All nodes simultaneously belong to N network topologies and are divided into N_G node groups. Therefore, it can happen that the node groups assign different fitnesses to the identical network topology.

2.4. Representations of Network Topologies. In EA, a solution candidate for an optimization problem is represented in an alternative form. This alternative form is designed by a person who is attempting to solve the problem using the EA and is referred to as a genotype or an individual. Meanwhile, a solution candidate itself is referred to as a phenotype in the EA. In P-EP2P, the P2P network topology is an object of optimization and an individual is an alternative form of a P2P network topology.

Suppose that a P2P network consists of L nodes. The P2P network topology assumed herein is generated by having each of the L nodes make N_C directed links to other nodes. Therefore, an individual is an internal representation of this network topology in the EA.

Each node is assigned a serial number as its identifier, and when $N_C = 1$, the identifier corresponds to the index of the vector representing the individual. When $N_C \geq 2$, the identifier corresponds to the index representing each chunk of N_C elements. An element value of the individual represents an identifier of the node to which a focus node makes a directed link. A direction represented by a directed link indicates that a search query can be forwarded only in that direction. Thus, if flooding is used as a query forwarding method, a search query generated at some node is forwarded

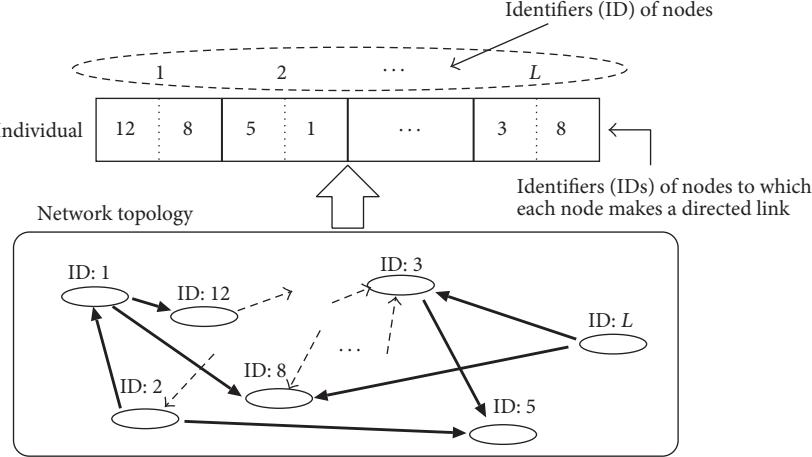
node by node in the direction represented by the directed links, and the paths for forwarding the query (flooding tree) are then determined accordingly. However, when data, such as a file, is found during this search, the node having the object transmits the data to the node making the query by means of a direct communication.

In fact, the most reliable way to search a network for a desired content is to form a full-mesh network topology in which every node links to all other nodes and uses flooding for search query forwarding in which a node issuing a search query forwards the query to all its neighboring nodes. By the flooding on the full-mesh network topology, a user issuing a search query can surely find a desired content if the desired content exists in the present network. However, the flooding on the full-mesh network topology causes heavy network traffic. In addition, in case that the user needs to confirm if each node holds the desired content, the user would feel much fatigue. Therefore, we adopt the evolutionary P2P networking in which a limited small number of times of query forwarding is performed on the evolutionarily optimized network topologies.

Figures 2 and 3 show representations of individuals of the EA used here when $N_C = 1$ and $N_C = 2$, respectively.

2.5. Evolutionary Operators. Evolutionary operators are applied to the set of individuals mentioned in Section 2.4, which is referred to as a population, in order to generate a new set of individuals, which is referred to as the new population. The number of individuals held in the EA, that is, the population size, is N . Evolutionary operators generally include a selection operator, which is inspired by natural selection in Darwinism, a recombination or crossover operator, which models genetic recombination, and a mutation operator, which models gene mutation. The evolutionary operators used in P-EP2P are explained below.

2.5.1. Selection. The selection operator used herein is a tournament selection with a tournament size of K . The tournament selection randomly selects K individuals from the EA population and selects the individual with the best

FIGURE 3: Representation of a P2P network topology in the EA (EA individual) when $N_C = 2$.

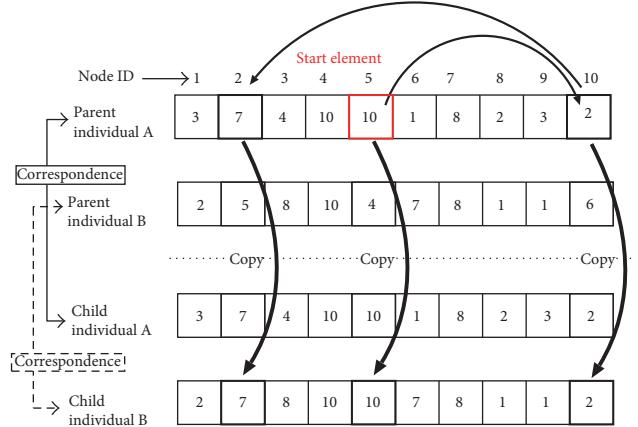
fitness among the K individuals. This selection procedure is repeated until N individuals have been selected. N is a population size and an even number.

In each node group, the tournament selection is conducted using fitnesses assigned to individuals encoding network topologies. Since it can happen that the node groups provide different fitnesses for identical individuals, selected individuals can be different in the node groups.

2.5.2. Crossover. The crossover operator used herein is *node linkage crossover* (NLX) that we previously proposed in [4]. NLX has been shown to form better network topologies than conventional one-point and uniform crossover operators under the evaluation scenario that all nodes belong to only one node group and search the network for only one particular node.

NLX is applied to the selected individuals by the tournament selection in each node group as follows. A range within which NLX is applied in each node group are the vector elements (the loci) that are correspondent to the nodes of that group.

- (1) N individuals selected by the selection operator are divided into $N/2$ pairs of individuals. The selected individuals become parent individuals.
- (2) The crossover operator is applied to each pair of parent individuals with probability p_c , which is referred to as the crossover rate. Child individuals generated from each pair of parent individuals are identical to the parent individuals before the crossover operator is applied. Each parent individual has a corresponding child individual.
- (3) For each pair of parent individuals to which the crossover operator is applied, one element is randomly selected from among the $G_k \times N_C$ ($k = 1, 2, \dots, N_G$) elements of the individual. Recombination is conducted for the selected element with probability p_e .
- (4) For the element to which the recombination is to be applied, which child individual corresponding to

FIGURE 4: Example of node linkage crossover (NLX) when $N_C = 1$.

one parent individual receives the element values of the other parent individual to be copied on itself is decided randomly.

- (5) After deciding which parent individual provides the element values for recombination, the node (element value) linkage generated by directed links between nodes is copied to the target child individual.

Figures 4 and 5 show examples of NLX with $N_C = 1$ and $N_C = 2$ when $N_G = 1$, respectively. For example, in Figure 5, the third node has been selected as the initial node of the linkage. However, since each node makes two directed links, the third node has two elements that can be referred to by NLX, which, in this example, are 10 and 1. Then, NLX randomly chooses one of the two possible elements and refers to the value of the selected element, which is 10. Next, since the second node of the linkage, which is the tenth node, also has two elements, NLX randomly chooses one of the elements and refers to the value of the selected element, which is 2. In this way, the node

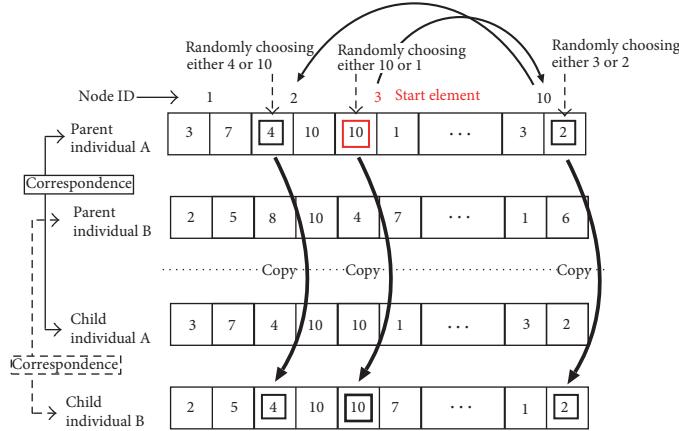


FIGURE 5: Example of node linkage crossover (NLX) when $N_C = 2$.

linkage is formed. Generally, when $N_C \geq 2$, NLX is performed in the same manner.

However, when there are multiple node groups ($N_G \geq 2$), the way becomes complicated. For example, suppose that a node a_1 in a node group 1 of focus makes a directed link to a node b_2 in another node group 2 and the node b_2 makes a directed link to a node c_3 in another node group 3. Then, we will consider a copy of the linkage among these nodes, $a_1 \rightarrow b_2 \rightarrow c_3$. In this case, the linkage from the node a_1 , which is in the node group 1 of focus, to the node b_2 , which is outside the node group 1, is copied, but the linkage from the node b_2 to the node c_3 , which is also outside the node group, cannot be copied.

In one attempt of NLX, when the number of times of recombination has not reached N_L yet and a linkage from a node in a node group of focus to a node in another node group appears, a linkage from the node in that other group to some node cannot be copied, as mentioned above. In this case, one new vector element is selected from all of the vector elements in the node group of focus to be copied. The node linkage again starts from that selected element (node). One attempt of NLX is finished when the total number of the vector elements copied becomes N_L .

- (6) Repeat Steps 3 through 5 $N_C \times G_k$ ($k = 1, 2, \dots, N_G$) times.

2.5.3. Mutation. The mutation operator used herein is such that the value at each element (the gene) on the N individuals obtained after NLX is randomly changed to some other possible value with probability p_m , which is referred to as the mutation rate. The element value of the individual represents an identifier of a node to which the node corresponding to the element position is linked, so that the mutation operator changes a node to which the node of focus is linked. Since all the super nodes exchange information on which nodes are present in their networks among them, every element value of the individual can become one of the identifiers for all of the nodes in the entire network by the mutation operator.

This mutation operator is introduced mainly for bringing novel genes that did not appear in the initial population. In addition, if we set the mutation rate to be higher, the P-EP2P approaches to a random method.

2.6. Transforming EA Population to Network Topologies. The EA population obtained after applying the evolutionary operators is transformed into a new set of P2P network topologies on the super node mentioned above, and the nodes to which each node must make directed links are then communicated to each node in the network. The nodes then make directed links to other nodes according to this information. Nodes that are not present in the network at this moment obtain information on nodes to which they must link upon joining the network.

3. Simultaneous and Sequential Evaluation Methods for Network Topologies

In our two earlier studies [4, 5], we considered that EP2P or P-EP2P forces all node groups to simultaneously conduct collection of fitness values to network topologies from the nodes and evolutionary reconstruction of the network topologies based on the fitness values at the same time. We refer to this method for evaluating and reconstructing network topologies as *the simultaneous topology evaluation method*.

In our most recent study [6], it was suggested from simulation results that when using the simultaneous topology evaluation method, change in network topologies of some node group can cause bad effect on fitness values of other node groups; that is, evolution of network topologies that each node group manages does not occur harmoniously. In P-EP2P using the simultaneous topology evaluation method, all node groups can give a different fitness value to one network topology, and then each node group can select and modify pieces of different network topologies for the next generation. Then, the entire network topologies are formed by randomly combining such pieces of different network topologies that all node groups selected. There is no basis that this random

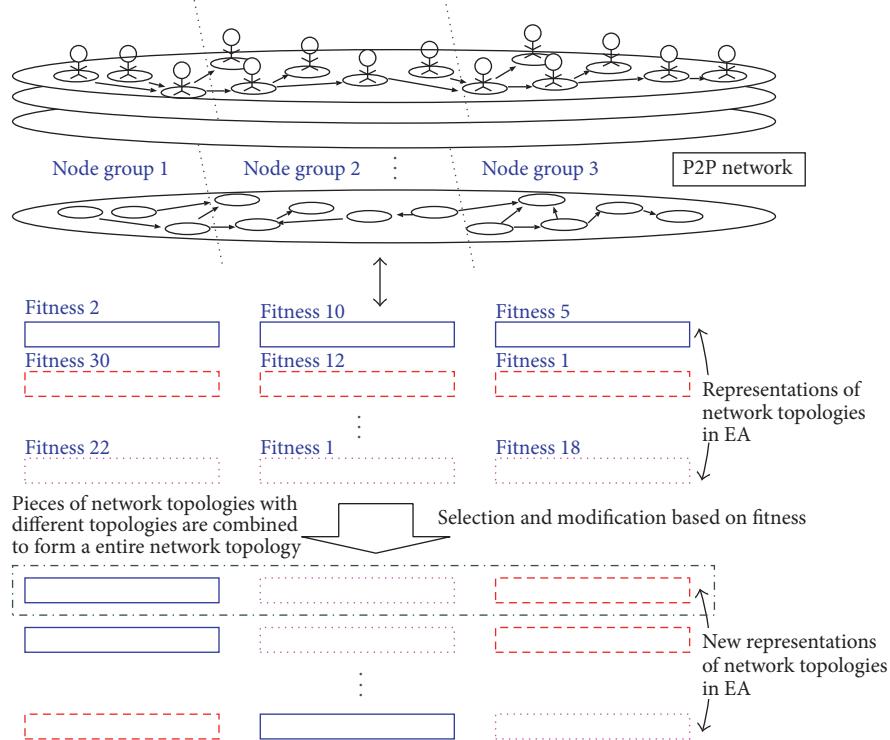


FIGURE 6: Problem of the parallel evolutionary P2P networking technique.

combination of pieces of different network topologies yields better network topologies. The problem of P-EP2P mentioned above is also shown in Figure 6. So, we proposed a method that evaluates and reconstructs network topologies one by one in a fixed order (see Figure 7). This method allows a node group to reconstruct its own network topologies under the condition that network topologies of other node groups are fixed. We refer to this method as *the sequential topology evaluation method*.

The procedures of P-EP2P using the simultaneous topology evaluation method are as follows: (1) all nodes are divided into N_G node groups, (2) nodes in each node group use the network topologies for a fixed time period of T , (3) a super node in each node group gathers fitness values from its nodes, (4) all node groups evolutionarily reconstruct its network topologies at the same timing, and (5) return to (2).

The procedures of P-EP2P using the sequential topology evaluation method are as follows: (1) all nodes are divided into N_G node groups, (2) nodes in each node group use the network topologies for a fixed time period of T , (3) a super node in each node group gathers fitness values from its nodes, (4) only a node group that is now at its turn evolutionarily reconstructs its network topologies, where order of the topology reconstruction for all node groups is determined in advance, and (5) return to (2). For example, suppose that two node groups, A and B, exist. Then, only the node group A first evolutionarily reconstructs its network topologies at time T using fitness values that were obtained from the real use of the network topologies by its nodes from time 0 to T . Next, only the node group B reconstructs its

network topologies at time $2T$ using fitness values gathered from time T to $2T$. Then, the turn comes to the node group A again. This procedure is repeated.

4. Simulations

We herein assume simulation scenarios that are more realistic than those used in our previous study [6] and evaluate P-EP2P under the assumed simulation scenarios. The simulator used in the simulations is of our own making in C programming language.

4.1. Configurations of the Parallel Evolutionary P2P Networking Technique. Table 1 shows configurations of P-EP2P used herein. The configurations are the same as those used in our previous study [6]. In the simulations, as for node joining and leaving at every time, we consider two scenarios. One is that all L nodes always stay in the network. This scenario is called “the all-in scenario.” The other is that a probability randomly determined in between 60% and 100% is assigned to each node and each node participates in the network with the assigned probability at every time. A unit of time in a simulation is defined as a time period needed for the same number of searches as the number of nodes that participate in the network at the beginning of the time period to be done. A node that conducts a search is randomly chosen from the nodes that participate in the network at the beginning of the time period. This scenario is called “the partially-in scenario.” The behaviors of nodes that dynamically join and leave the network are often called node churn [7, 8].

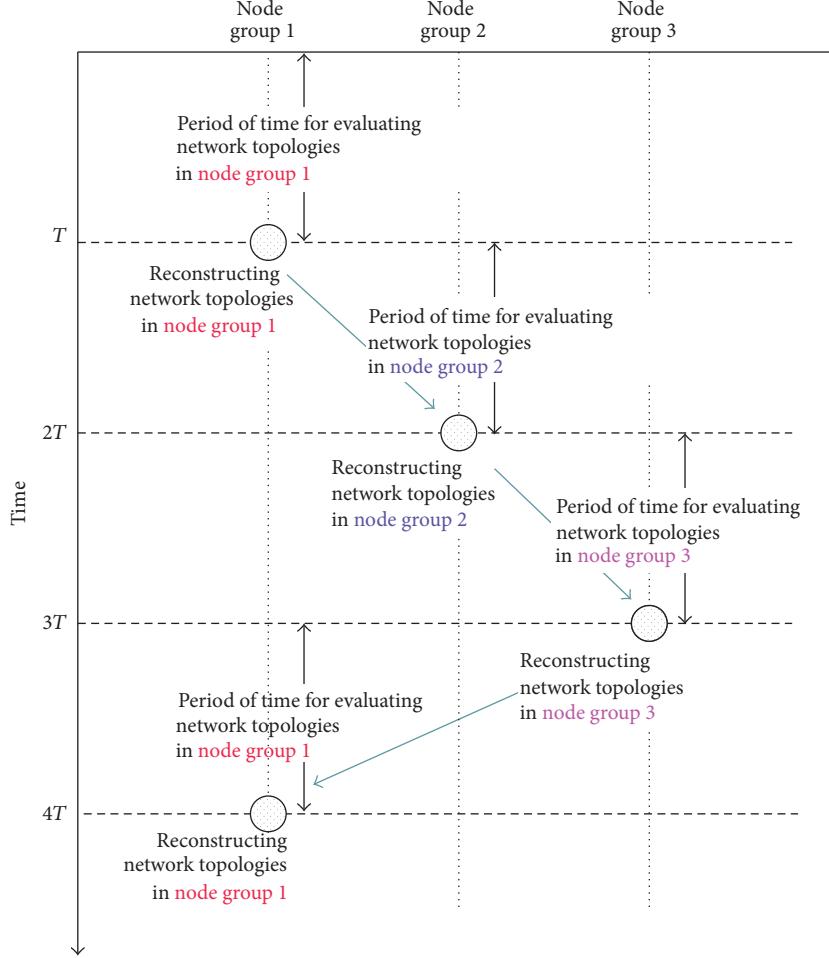


FIGURE 7: Timing to reconstruct the network topologies by the sequential topology evaluation method. The number of node groups, N_G , is three.

TABLE 1: Parameters values of P-EP2P used in the simulations.

Parameter	Description	Value
L	Number of nodes (genes)	10^3
N	Number of P2P network topologies (individuals)	30
T	Time period for which generated topologies are used	20
H_{\max}	Allowed number of hops for one search	5
D	Number of directed links generated by a node in one topology	1
K	Tournament size for the tournament selection	2
p_c	Crossover rate	100%
p_e	Probability with which recombination is conducted for a selected element in NLX	10%
N_L	Length of node linkage in NLX	5
p_m	Mutation rate	0.5%
N_G	Number of node groups	1, 2, 5, 10
G_k	Number of nodes in the k th node group	L/N_G

As mentioned above, the unit time in the simulations is defined as a time period needed for all present nodes to conduct a search. Similar to this, the unit time in the real execution of P-EP2P should be defined as a time period needed for a fixed number of searches to be done. Ideally,

the unit time in the real execution of P-EP2P should be the same as the unit time in the simulations because the present network topologies are fairly evaluated by all present nodes. In addition, it is desirable that the environment is not changed during several units of time in order that the evolutionary

topology reconstruction works effectively. However, it is hard for us to know the velocity of variation of the environment in advance. So, we will consider a way to adaptively define the unit time in the future work.

In the all-in scenario, all L nodes participate in the network at every time. Meanwhile, in the partially-in scenario, we confirmed through simulations that approximately $0.75 \times L$ nodes participate in the network at every time.

4.2. Simulation Scenarios. We have already mentioned simulation scenarios with respect to node joining and leaving above. We herein design simulation scenarios from other aspects and merge them into the scenarios on node joining and leaving, that is, the all-in scenario and the partially-in scenario.

In the simulations, we assume a P2P network for streaming of a motion video [9], where replication of motion videos for streaming is not conducted and each node searches the network for a desired motion video when necessary. Replication in a P2P network stands for creating copies of original contents in other nodes. A P2P file-sharing network often employs a replication strategy [10].

Simulation scenarios are produced by changing how to decide streaming contents held by each node and how to decide streaming contents retrieved by each node. The total number of kinds of streaming contents held by all nodes in the network is 200. Each node holds one kind of content and retrieves one kind of content by a search. The 200 kinds of contents are assigned serial numbers from 1 to 200.

We use two ways to decide a content held by each node. One is to randomly decide it, which is called *random holding*, and the other is to decide it according to Zipf's law [11], which is called *Zipf's holding*. Zipf's law in the context of information retrieval indicates that when the order of popularity of a content is x ($x = 1, 2, \dots, 200$), probability with which the content is retrieved by a node is proportional to $x^{-\alpha}$ ($\alpha > 0$). Here we consider not probability of retrieval of a content but probability with which a node holds the content. The parameter α adjusts the deviation of popularities among the contents. The larger the value of α is, the larger the deviation is. We set the alpha to be 1 for all simulations in the paper.

We use three ways to decide contents for which nodes search the network. One is to randomly decide them, which is called *random searching*. Another is to decide them according to Zipf's law, which is called *Zipf's searching*. This way is basically the same as the way to decide contents held by nodes according to Zipf's law (*Zipf's holding*) but considers probability of searching for contents. When contents held by nodes and contents searched for by nodes are both decided according to Zipf's law, popularities of the contents are identical. That is to say, a content of the serial number i is held and searched for by a node with the same probability proportional to i^{-1} . Another is that all nodes search the network for only one kind of content, which is called *particular searching*.

As mentioned in Section 2, a super node in P-EP2P does not manage what contents each node holds. Therefore, it is basically impossible to provide a list of contents distributed in the network for nodes. Then, a search query issued by a node cannot be a form of a content name. We here ideally assume

that a search query takes a form of keywords and nodes can guess keywords to retrieve a desired content somehow.

We prepare six evaluation scenarios by combining the two ways to decide contents held by nodes and the three ways to decide contents for which nodes search the network. In addition, for reference, we use the same evaluation scenario used in our previous study [6]. This scenario is that not one kind of content but one node is searched the network for by all nodes, which is called *one node searching*.

4.3. Simulation Results. We observed the time transition of search failure rate every 20 unit times as a simulation result. We show the simulation results of P-EP2P using the simultaneous and sequential topology evaluation methods when all nodes always stay in the network in Figure 8 and when each node participates in the network with the probability randomly assigned in between 60% and 100% at every time in Figure 9. A simulation result of P-EP2P for each evaluation scenario is the average result over 30 independent simulation runs. In all evaluation scenarios, the same set of 30 initial network topologies were used for 30 independent simulation runs.

In Figures 8 and 9, the simulation result of the simultaneous topology evaluation method is labeled as "simultaneous" and that of the sequential topology evaluation method is labeled as "sequential." In addition, the number of node groups is represented by " N_g ". For example, the number of node groups is two, the simulation result is labeled as " $N_g = 2$ ".

First, comparing Figures 8 and 9, we can confirm that P-EP2P with the identical configuration has a similar tendency on the time transition of search failure rates between when node joining and leaving did not occur and when it occurred, though the search failure rates in the case that node joining and leaving occurred were larger than those in the case that node joining and leaving did not occur in most evaluation scenarios. Therefore, we discuss the tendencies of time transitions of search failure rates for the two cases together in the followings.

Comparing the simulation results of P-EP2P using the simultaneous and sequential topology evaluation methods, we can observe that, for all evaluation scenarios and for all number of groups, N_g , P-EP2P using the sequential topology evaluation method is better than or equal to that using the simultaneous topology evaluation method. The smaller the number of kinds of contents as search targets become, the larger the performance difference between them is. That is to say, the performance difference becomes larger in the order of one node searching, particular searching, Zipf's searching, and random searching.

Meanwhile, we can observe that search failure rates increased right after the simulation start for all evaluation scenarios. The more diverse search targets of contents were, the more significantly search failure rates increased. The reason for this would be that although search targets of contents were diverse, the evolutionary method, especially, the network topology selection produced network topologies that were specialized for finding particular contents among diverse desired ones. As a result, it would become hard to

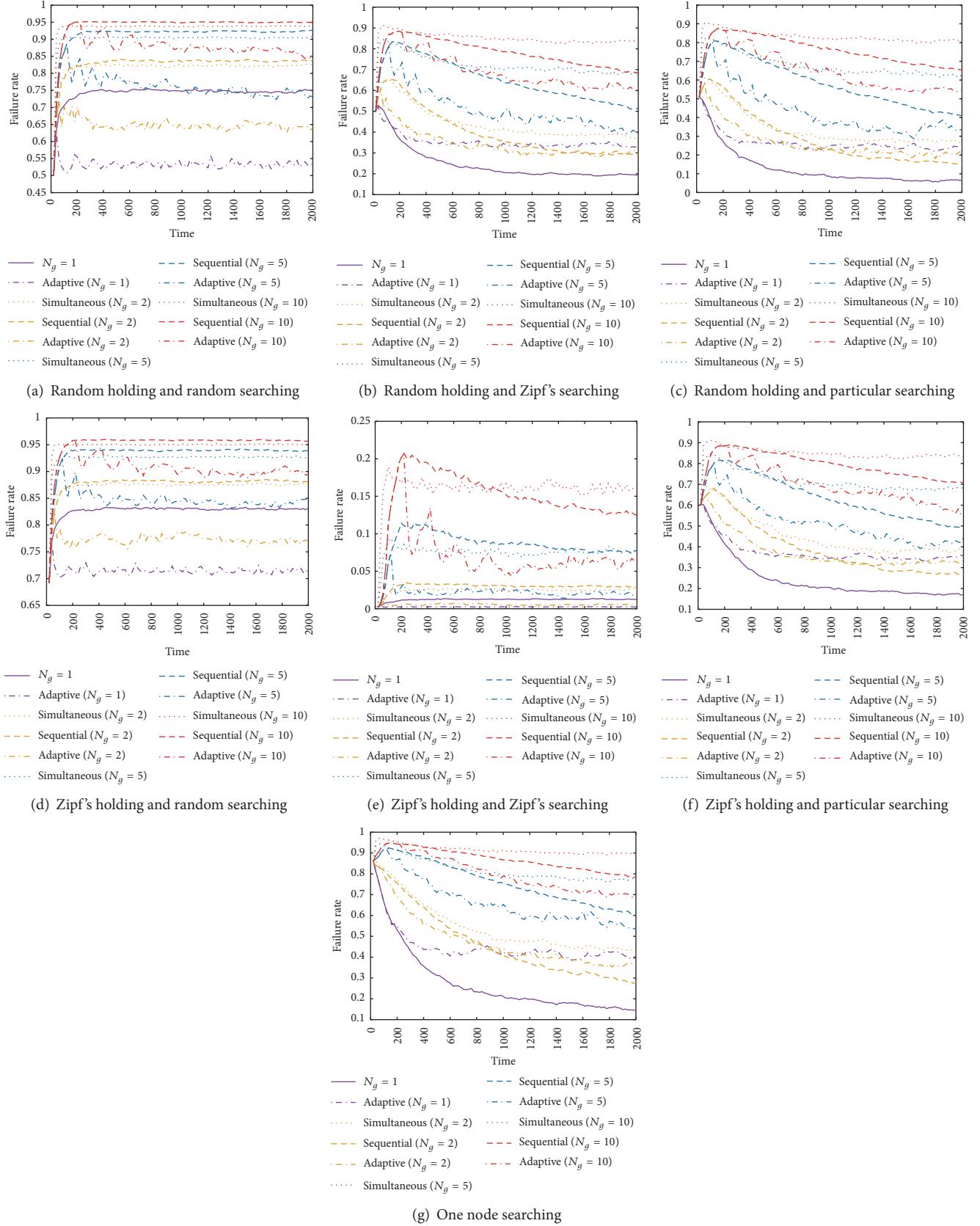


FIGURE 8: Simulation results of P-EP2P using the sequential and simultaneous topology evaluation methods when all nodes always stay in the network.

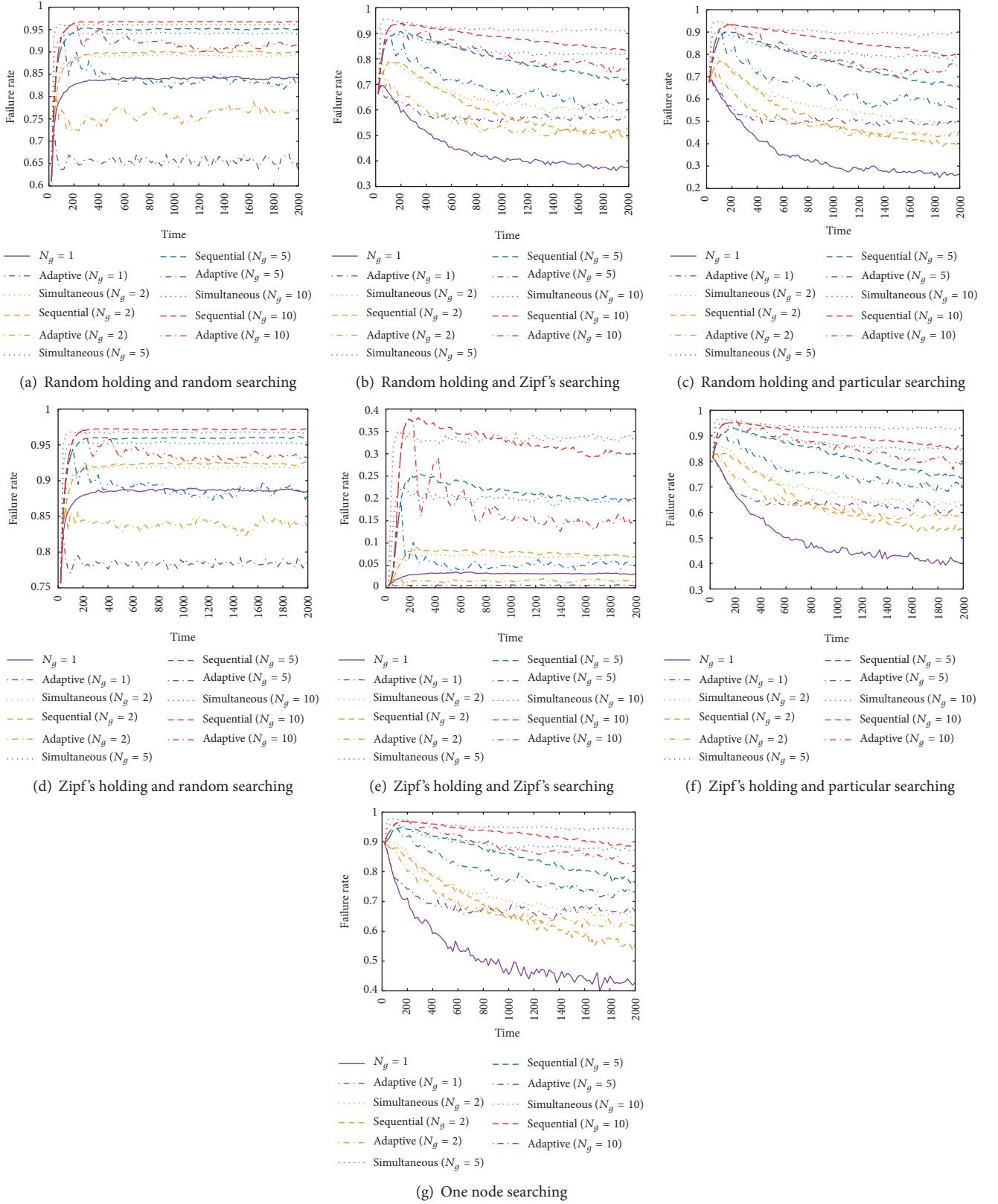


FIGURE 9: Simulation results of P-EP2P using the sequential and simultaneous topology evaluation methods when each node participates in the network with the probability randomly assigned in between 60% and 100% at every time.

find contents except the particular contents using the evolved network topologies, and therefore, the search failure rates would significantly increase. When search targets of contents were particular ones, search failure rates also increased by evolving network topologies specialized for some search targets of contents among the particular ones. However, the increase temporarily occurred. Since the number of the contents as search targets was small in the first place, the crossover and mutation operators used could evolve the network topologies that were suited for finding all those particular contents. Then, as mentioned above, P-EP2P using the sequential topology evaluation method was better.

From the above-mentioned simulation results, we can conclude that P-EP2P causes the significant increase in search failure rates right after evolutionarily reconstructing random network topologies to new ones. This is just a transitional state and it would be a matter of how a state after the transitional state is like. As we mentioned above, when search targets of contents are diverse, it seems to be hard to improve search failure rates even after passing the transitional state. That is to say, when search targets of contents are diverse, not evolutionary network topology reconstruction, which causes convergence of network topologies, but random network topology reconstruction, which diversifies network topologies, would be appropriate. Thus, we need a method that first judges which is a suitable way for the present situation, evolutionary network topology reconstruction, or random reconstruction and then applies the suitable way to the present network topologies.

5. Strategy for Adapting Network Topologies to Realistic and Dynamic Environments

As mentioned at the end of Section 4, P-EP2P needs to judge which way is suitable for the present situation, evolutionary network topology reconstruction, or random reconstruction somehow. It is evident from the simulation results obtained in the last section that the more particular search targets for nodes in a network become, the more suitable an evolutionary way is, and in the opposite situation, a random way is more suitable. However, in reality it is hard to know what objects nodes search a network for in a real-time manner because of the nature of a P2P network as a distributed system. We need an alternative way to solve this problem. Here an evolutionary way stands for a way to change network topologies by applying the selection, crossover, and mutation operators explained in Section 2 to them, and a random way stands for a way to change them by applying only the mutation operator with the mutation rate of 0.25 ($p_m = 0.25$) to them.

So, we propose the following adaptive strategy to select a suitable way for network topology reconstruction. In each node group, at the t th time of network topology reconstruction, if a search failure rate obtained by a currently adopted way, which is the evolutionary way or the random way, $f_r(t)$, is better than the search failure rate obtained at one time before the present time, $f_r(t - 1)$, the currently adopted way is kept using until the next time of network topology reconstruction, that is, $(t + 1)$ th time. On the other hand,

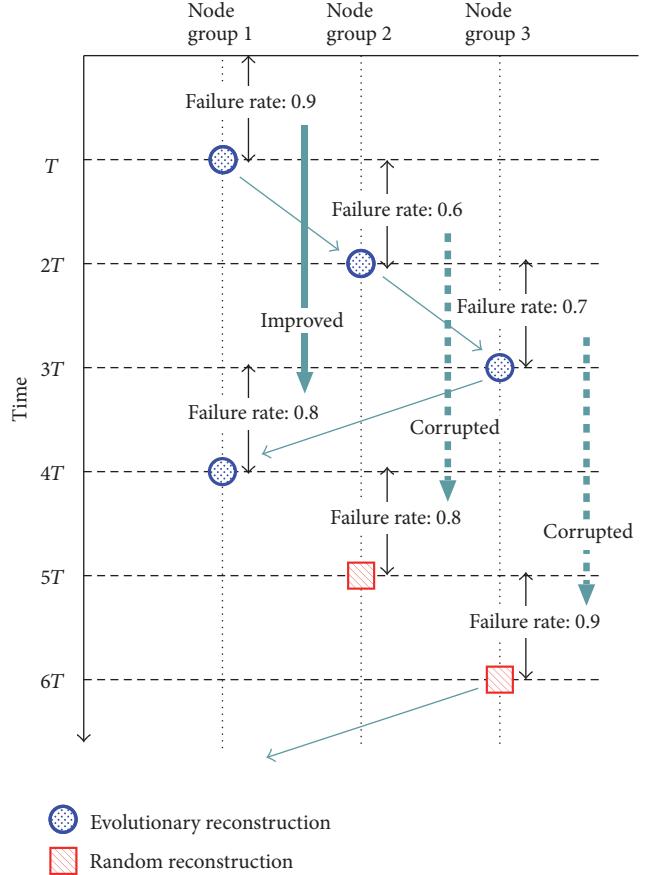


FIGURE 10: The proposed adaptive strategy for network topology reconstruction.

if the search failure rate is worse than the previous one, the currently adopted way is changed to the other way. In any case, at the next time of network topology reconstruction, that is, $(t + 1)$ th time, an obtained search failure rate, $f_r(t + 1)$, is compared to $f_r(t)$. The proposed adaptive strategy is illustrated in Figure 10.

We combined P-EP2P using the sequential topology evaluation method with the adaptive strategy proposed here. We refer to this P-EP2P as P-EP2P with the adaptive strategy hereinafter. Then, we conducted simulations of P-EP2P with the adaptive strategy using the same evaluation scenarios presented in Section 4. We drew the simulation results together in Figures 8 and 9, which include the simulation results of Section 4. A simulation result of P-EP2P for each evaluation scenario is the average result over 30 independent simulation runs. In all evaluation scenarios, the same set of 30 initial network topologies were used for 30 independent simulation runs and they are identical to those used in the simulations of Section 4. In Figures 8 and 9, the simulation result of this P-EP2P is labeled as “adaptive.” In addition, the number of node groups is represented by N_g .

First, similar to P-EP2P using the simultaneous and sequential topology evaluation methods, comparing Figures 8 and 9, we can confirm that P-EP2P with the adaptive strategy that had the identical configuration has a similar

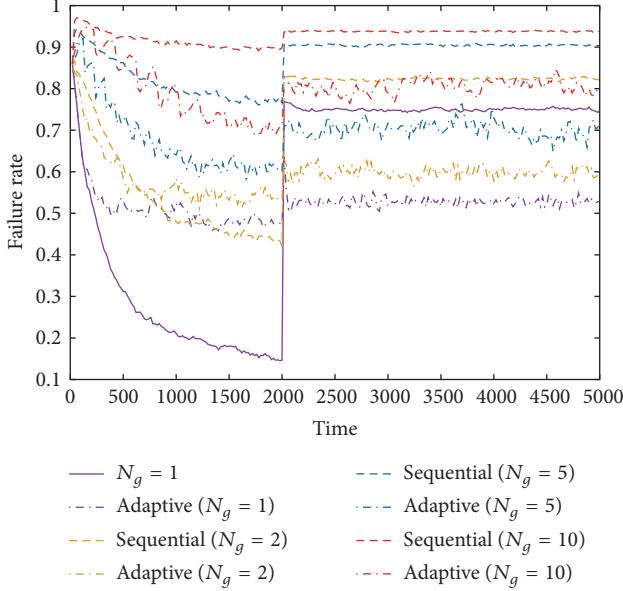


FIGURE 11: The simulation results of P-EP2P using the sequential topology evaluation method (labeled as “sequential”) and P-EP2P with the adaptive strategy (labeled as “adaptive”) for the evaluation scenario including dynamic environmental change, where all nodes always stay in the network.

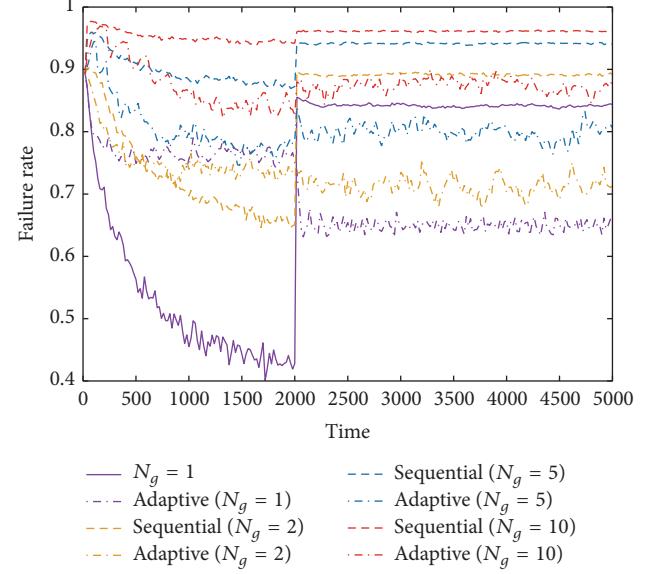


FIGURE 12: The simulation results of P-EP2P using the sequential topology evaluation method (labeled as “sequential”) and P-EP2P with the adaptive strategy (labeled as “adaptive”) for the evaluation scenario including dynamic environmental change, where each node participates in the network with the probability randomly assigned in between 60% and 100% at every time.

tendency on the time transition of search failure rates between when node joining and leaving did not occur and when it occurred. Therefore, we discuss the tendencies of time transitions of search failure rates for the two cases together in the following.

Comparing the simulation results of P-EP2P using the sequential topology evaluation method, P-EP2P using the simultaneous topology evaluation method, and P-EP2P with the adaptive strategy, we can observe that except the case that search targets of contents were particular ones and the number of node group, N_G , was one, P-EP2P with the adaptive strategy improved search failure rates more significantly than the others after the initial rapid increase of search failure rates had occurred. As expected, we can think that when search targets of contents were diverse, the random topology reconstruction way was indeed selected and introducing randomness into network topology reconstruction contributed to the improvement of search failure rates.

Moreover, to examine the effect of the adaptive strategy, we consider an evaluation scenario including dynamic change in the environment. In the evaluation scenario, the evaluation scenario of “random holding and one node searching” is used from time 1 to time 2,000 and then is suddenly switched to the evaluation scenario of “random holding and random searching” at time 2,001 and the scenario continues up to time 5,000. This dynamic evaluation scenario is an extreme example of scenarios in which the situation that the number of nodes holding desired search objects is quite small is shifted to the one that the number of nodes holding desired search objects is quite large. However, as the dynamic evaluation scenario used here, it would be a general

situation that users search a network for a few popular search objects that only a few nodes hold for a short period of time, but the search objects fall in popularity, and then, the users search a network for various search objects that many nodes separately hold.

We applied P-EP2P using the sequential topology evaluation method and P-EP2P with the adaptive strategy to the above-mentioned evaluation scenario including the dynamic environmental change. We show the simulation results of the two types of P-EP2P when all nodes always stay in the network in Figure 11 and when each node participates in the network with the probability randomly assigned in between 60% and 100% at every time in Figure 12. We can guess from Figures 11 and 12 that when only evolutionary topology reconstruction was carried out, the network topologies almost converged to particular network topologies. Consequently, the search failure rates would rapidly increase at time 2,001 and thereafter. Meanwhile, P-EP2P with the adaptive strategy could restrain the rapid increase of the search failure rates. These results suggest the effectiveness of the adaptive strategy in dynamic environments.

Furthermore, we observed which topology reconstruction way was taken by each node group in P-EP2P using the adaptive strategy at every time, together with the time variation of search failure rates. The obtained results are shown in Figures 13 and 14.

We expected before executing this simulation that the evolutionary topology reconstruction way is mainly selected from time 1 to 2000 in which the scenario of “random holding and one node searching” is employed and the random topology reconstruction way is mainly selected from time

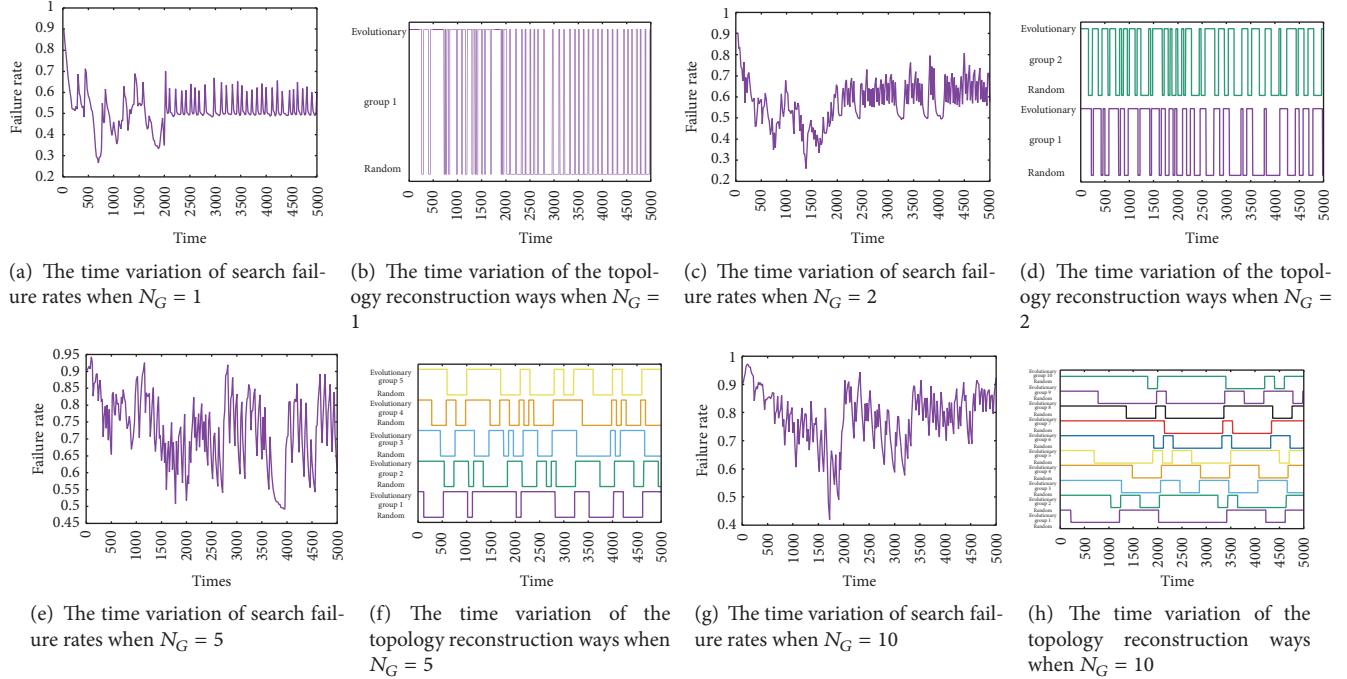


FIGURE 13: The result of one simulation run when using the simulation scenario that all nodes always stay in the network and the environment dynamically changes. The time variations of search failure rates are drawn in the subfigures (a), (c), (e), and (g) and the time variations of the topology reconstruction ways are drawn in the subfigures (b), (d), (f), and (h). The time variation of the topology reconstruction ways is represented by a sequential line graph, in which drawing a line on the upper side stands for the use of the evolutionary reconstruction strategy and drawing a line on the lower side stands for the use of the random topology reconstruction.

2001 to 5000 in which the scenario of “random holding and random searching” is employed. We can observe from Figures 13 and 14 that the expectation is roughly true no matter whether node churn exists.

However, in case of any number of node groups, we can observe all over the simulation period of time that the search failure rate first decreased to a certain extent and then increased and again decreased. Furthermore, since a way to reconstruct the network topologies varies during such a fluctuation of search failure rates, the fluctuation of search failure rates is thought to be caused by the change in topology reconstruction ways. For instance, when the number of node groups is one, the search failure rates can originally keep decreasing by selecting the evolutionary topology reconstruction way until time 2,000. However, the adaptive strategy sometimes selected the random topology reconstruction way, and the search failure rates stopped decreasing accordingly. This result suggests that the effect of the change in topology reconstruction ways on search failure rates is large. We need to further investigate and adjust the sensitivity of the adaptive strategy to environmental changes.

6. Related Researches

Our present study deals with a technique that optimizes P2P network topologies based on fitness obtained from nodes in an on-line manner. So, we here describe researches on optimization of P2P network topologies by computational intelligence techniques.

First, except our previous studies [4–6], there is only one study on the evolutionary P2P networking technique (EP2P) [12]. This study dealt with not the parallel evolutionary P2P networking technique (P-EP2P) but EP2P in which only one node group exists and proposed two methods for maintaining diversity of network topologies as a population of the evolutionary algorithm.

The first method proposed in [12] is to devise a fitness function to maintain network topologies that include various links between nodes in a population. The method introduces an index that measures degree of overlap of links between network topologies, which was named “repetition factor,” and if multiple network topologies providing the same search success rate exist, the method preferentially takes network topologies with lower degree of the link overlap.

Furthermore, under the situation that the first method maintains diversity of a population, an elite strategy is introduced, which makes better a half of the present population occupy a half of the next generation population. This elite strategy is the second method. An extreme elite strategy often causes the premature convergence in evolutionary search, but the elite strategy used here effectively works as a method to maintain useful links under the use of the method for maintaining the diversity of network topologies. EP2P that used these two methods were shown to be better than EP2P that did not use them in terms of search failure rate as well as the speed of convergence to better network topologies for all nodes under the evaluation scenario that all nodes search the network for only one node, which is the same as “one node

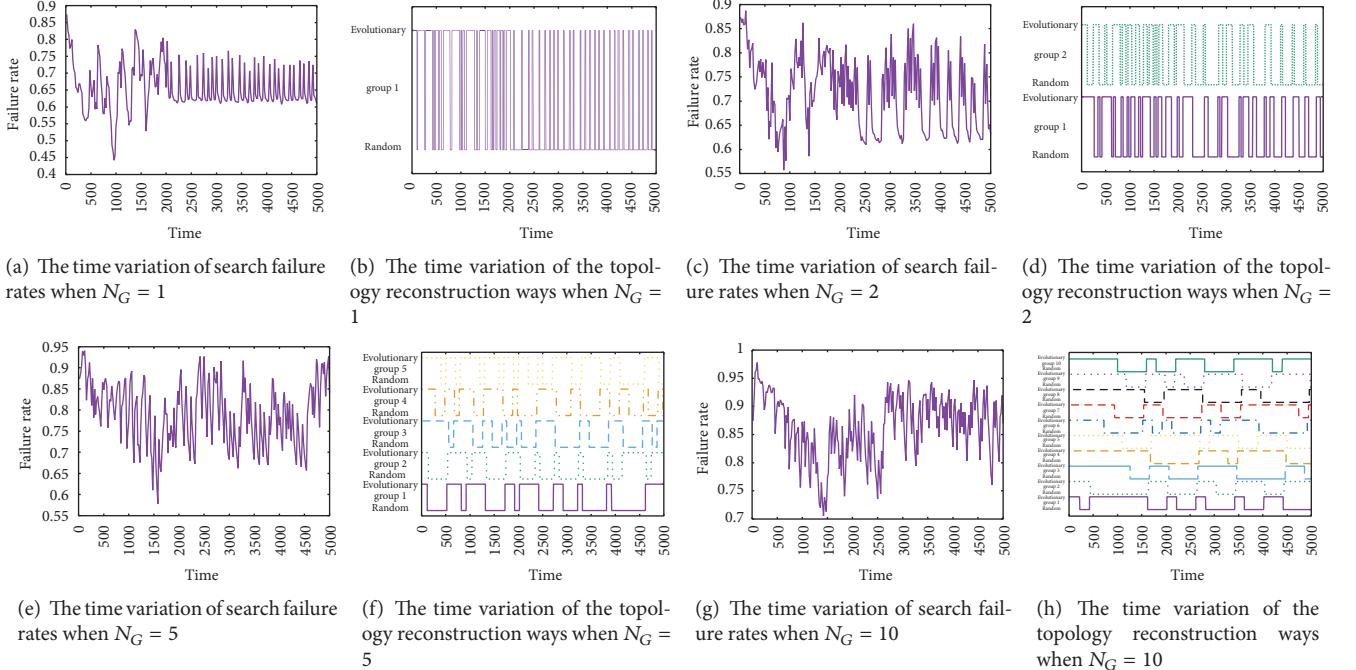


FIGURE 14: The result of one simulation run when using the simulation scenario that each node participates in the network with the probability randomly assigned in between 60% and 100% at every time and the environment dynamically changes. The time variations of search failure rates are drawn in the subfigures (a), (c), (e), and (g) and the time variations of the topology reconstruction ways are drawn in the subfigures (b), (d), (f), and (h). The time variation of the topology reconstruction ways is represented by a sequential line graph, in which drawing a line on the upper side stands for the use of the evolutionary reconstruction strategy and drawing a line on the lower side stands for the use of the random topology reconstruction.

searching” scenario used in Sections 4 and 5, in static and dynamic environments. The static environment here means that all nodes always stay in the network and the dynamic environment means that incoming and outgoing nodes exist in the network.

Moreover, this study also assumed the evaluation scenario that popularity of contents as search targets follow the Zipf’s law as in our present paper and showed that, in two cases where replication was used and not used, only EP2P that used these two methods was better than or equal to a P2P network using randomly generated network topologies in terms of search failure rate. In case of not using a replication method, losing diversity of links between nodes by an evolutionary way would be linked to the increase in search failure rates. Therefore, the two methods that can maintain diversity of good links worked effectively in the case. We proposed the adaptive strategy for the same purpose, that is, maintaining diversity of links, in Section 4. In case of using a replication method that produces a replica of a content in a node that searched for and obtained the content, as we showed in our previous study [4], an evolutionary way does not work effectively because a network topology useful for a node to obtain a desired content at one moment becomes a useless one for the node as the node has already held the desired content and also content distribution varies with time due to contents replication.

A genetic algorithm was used for determining neighboring nodes in a P2P file-sharing network [13]. Also, a

particle swarm optimization (PSO) was used for determining them [14]. In these studies, it was assumed that a file was divided into fragments and fragments for constructing the entire file were downloaded from many nodes, and the goal was to achieve the shortest download time to download the fragments for the entire file. These studies are similar to our present study in optimizing a network topology by a meta-heuristics technique, but the optimization in [13, 14] can be thought to be off-line optimization, though there was no description on that, and it is a significant difference from our present study.

For each node in a P2P network to find desired resources quickly reliably, a method for determining links of each node to others was proposed [15]. The method executes a neural network to determine them in each node. This method, as our P-EP2P, changes a network topology every time a fixed amount of search queries were generated in an on-line manner. However, weights between neurons in a neural network have been trained before using the neural network in a P2P network and that is different from our approach in which learning itself, more precisely reinforcement learning itself, is executed in an on-line manner.

A genetic algorithm was used for constructing a P2P network topology that includes all existing nodes and minimizes a defined cost [16, 17]. However, these studies conducted off-line optimization that utilized information of the whole network. Meanwhile, our approach conducts on-line optimization.

A genetic algorithm was used to optimize a topology of an overlay network including a P2P network in an on-line manner [18]. This study evaluated the presented approach using the evaluation scenario in which some links of a physical network as the basis of overlay networks have failed and then the approach tries to reconstruct an overlay network topology to enable efficient data communication between given two overlay nodes. This study is similar to our studies with respect to evolutionary on-line optimization. However, optimization target of this study is a single network topology, while that of our studies is a set of network topologies. In addition, this study evaluated a network topology based on results of routing between two nodes, while our studies evaluated network topologies based on results of searching for desired contents by many nodes (users).

7. Conclusions

In the present paper we further extended our series of studies on the parallel evolutionary peer-to-peer networking technique (P-EP2P). We evaluated if P-EP2P can adapt the network topologies to realistic environments through simulations and showed that there are situations hard for P-EP2P to adapt them to. Then, based on the simulation results, we proposed an adaptive strategy to enable P-EP2P to adapt the network topologies to the realistic environments and showed that the adaptive strategy is actually able to enhance the adaptability of P-EP2P to the realistic environments. Although there is no other technique for comparison with P-EP2P and we compared variations of P-EP2P with each other, we could demonstrate the ability of the framework of P-EP2P and can expect from the results that P-EP2P with the new strategy works well even in the real world. We think that the main contribution of the present study was to present such a promising way to realize an evolvable network in which the evolution direction is given by humans. Nodes of an unstructured P2P network can be considered to be equivalent to humans depending on applications. For instance, nodes of an unstructured P2P content-sharing network are basically equivalent to users.

The final goal of our series of studies on P-EP2P is to realize an actual P2P network implementing P-EP2P and to examine if the realized P-EP2P can adapt network topologies to users' (nodes') demands. For example, by using WebRTC (Web Real-Time Communication) which is a technology enabling us to realize a P2P communication through web browsers, we would be able to construct a P2P network for video-streaming that implements P-EP2P as mentioned in the present paper as an example of its real applications. If we construct a real P2P network implementing P-EP2P, we can obtain knowledge on user behaviors in the P2P network implementing P-EP2P, and such knowledge will be utilized for simulation studies for developing new techniques.

In addition, in the future work, we will consider a method for dividing an entire network into small networks. This procedure is needed to implement P-EP2P. However, in the present paper we assumed that the network division has been done in advance. For this network division, we will focus on granular computing [19, 20]. More concretely, we will

consider applying a data clustering method based on granular computing to the parallel P2P networking problems. Granular computing is a computational framework that builds a complex system by organizing granules such as clusters and intervals and has recently attracted attention. Meanwhile, P-EP2P includes P2P nodes (users), small networks that consists of nodes, and an entire network that consists of small networks. If we consider nodes and small networks as granules, clustering nodes to build small networks and an entire network match the concept of data clustering based on granular computing. For example, we can consider gathering nodes (users) who have similar interest in the same small network to facilitate network services related to the shared interest.

Competing Interests

The author declares that they have no competing interests.

Acknowledgments

This work was supported partly by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (C) (25330289) and partly by the Cooperative Research Program of Information Initiative Center, Hokkaido University.

References

- [1] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.
- [2] M. Sathe, *Interactive Evolutionary Algorithms for Multi-Objective Optimization*, VDM, 2008.
- [3] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.
- [4] K. Ohnishi and Y. Oie, "Evolutionary P2P networking that fuses evolutionary computation and P2P networking together," *IEICE Transactions on Communications B*, vol. 93, no. 2, pp. 317–328, 2010.
- [5] K. Ohnishi and Y. Oie, "Parallel evolutionary P2P networking for realizing adaptive large-scale networks," in *Proceedings of the 11th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT '11)*, pp. 532–537, Munich, Germany, July 2011.
- [6] K. Ohnishi and M. Koppen, "Parallel evolutionary P2P networking technique evaluating network topologies sequentially," in *Proceedings of the IEEE 37th International Computer Software and Applications Conference Workshops (COMPSACW '13)*, pp. 22–26, IEEE, July 2013.
- [7] D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing unstructured overlay topologies in modern P2P file-sharing systems," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 267–280, 2008.
- [8] C. Vassilakis and I. Stavrakakis, "Minimizing node churn in peer-to-peer streaming," *Computer Communications*, vol. 33, no. 14, pp. 1598–1614, 2010.
- [9] R. S. Cruz and M. S. Nunes, "A P2P streaming architecture supporting scalable media," *Peer-to-Peer Networking and Applications*, vol. 8, no. 5, pp. 758–776, 2015.

- [10] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and replication in unstructured peer-to-peer networks,” in *Proceedings of the 16th International Conference on Supercomputing*, pp. 84–95, New York, NY, USA, June 2002.
- [11] L. A. Adamic and B. A. Huberman, “The nature of markets in the world wide web,” *Quarterly Journal of Electronic Commerce*, vol. 1, no. 1, pp. 5–12, 2000.
- [12] E. Pérez-Cortés and H. Sato, “On the impact of path redundancy awareness in evolutionary P2P networking,” *The Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 17, no. 6, pp. 872–882, 2013.
- [13] S. G. M. Koo, K. Kannan, and C. S. George Lee, “On neighbor-selection strategy in hybrid peer-to-peer networks,” *Future Generation Computer Systems*, vol. 22, no. 7, pp. 732–741, 2006.
- [14] A. Abraham, H. Liu, and A. E. Hassanien, “Multi swarms for neighbor selection in peer-to-peer overlay networks,” *Telecommunication Systems*, vol. 46, no. 3, pp. 195–208, 2011.
- [15] A. Auvinen, T. Keltanen, and M. Vapa, “Topology management in unstructured P2P networks using neural networks,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 2358–2365, Singapore, September 2007.
- [16] P. Merz and S. Wolf, “Evolutionary local search for designing peer-to-peer overlay topologies based on minimum routing cost spanning trees,” in *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN '06)*, pp. 272–281, Reykjavik, Iceland, September 2006.
- [17] Y. Li, J. Yu, and D. Tao, “Genetic algorithm for spanning tree construction in P2P distributed interactive applications,” *Neurocomputing*, vol. 140, pp. 185–192, 2014.
- [18] A. J. Ramirez, D. B. Knoester, B. H. C. Cheng, and P. K. McKinley, “Plato: a genetic algorithm approach to run-time reconfiguration in autonomic computing systems,” *Cluster Computing*, vol. 14, no. 3, pp. 229–244, 2011.
- [19] A. Bargiela and W. Pedrycz, *Granular Computing: An Introduction*, vol. 717 of *The Springer International Series in Engineering and Computer Science*, Kluwer Academic Publishers, Boston, Mass, USA, 2003.
- [20] W. Pedrycz, A. Skowron, and V. Kreinovich, Eds., *Handbook of Granular Computing*, Wiley-Interscience, 2008.

