Evolutionary P2P Networking for Realizing Adaptive Networks

Kei Ohnishi and Mario Köppen and Kaori Yoshida and Yuji Oie

Abstract Recently, a peer-to-peer (P2P) network model became very popular. This model is different from a conventional client-server network model. While a conventional client-server network model explicitly distinguishes nodes providing services (servers) from nodes receiving services (clients), a P2P network model does not assign fixed roles to nodes. One type of P2P networks are unstructured P2P networks that do not include any mechanism to manage data locations and which consist only of nodes that communicate with each other through direct connections. A direct connection between two nodes in unstructured P2P networks is represented by a logical network link, and therefore, a structure formed by logical links and nodes, that is, a P2P network topology, can be formed freely. Thus, unstructured P2P networks are flexible. However, they require some mechanisms to realize quick, accurate, and reliable searches. A free-form P2P network topology can be a control object for realizing such searches. One method that can adaptively modify a free-form topology of a running unstructured P2P network for quick, accurate, and reliable searches is an evolutionary algorithms (EA) inspired by biological genetics and evolution. Although EA is not the only way for adjusting the entire network topology of a running P2P network, EA would be suitable for this modification because EA can hold several solution candidates (i.e. future options for network topologies) simultaneously

Kei Ohnishi
Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan, e-mail: ohnishi@cse.kyutech.ac.jp

Mario Köppen
Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan, e-mail: mario@ndrc.kyutech.ac.jp

Kaori Yoshida
Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan, e-mail: kaori@ai.kyutech.ac.jp

Yuji Oie
Kyushu Institute of Technology, 680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan, e-mail: oie@cse.kyutech.ac.jp

in a running P2P network. An evolutionary P2P networking technique (EP2P) and a parallel evolutionary P2P networking technique (P-EP2P) are both a fusion technique of an EA and an unstructured P2P network, and optimize the manner in which the nodes belong to different P2P network topologies in realtime. We will introduce EP2P and P-EP2P and show simulation results that validate both techniques.

1 Introduction

Recently, peer-to-peer (P2P) network models have attracted a great deal of attention. The concept of the P2P network model is completely different from that of a conventional client-server network model. While a conventional client-server network model explicitly distinguishes hosts providing services (servers) from hosts receiving services (clients), a P2P network model does not assign fixed roles to hosts. Hosts composing P2P networks, referred to as *peers*, can be both servers and clients, so that P2P networks could be used to facilitate autonomic and decentralized service management. In a client-server network, all data required by clients are held only by a server. Meanwhile, in a P2P network, data are held by each peer. So, no matter what type of P2P network is used, a mechanism to locate peers holding data requested by some peer is needed in P2P networks. We use the general term of "*node*" for representing "*peer*" hereinafter.

P2P networks are in fact roughly classified into three types. The first type are hybrid P2P networks. Hybrid P2P networks consist of an index server and nodes. The index server does not hold any data, but manages what data each node holds. Therefore, a node asks the index server when it desires some data, and from this it knows which node holds the desired data, and then it acquires the desired data from that node through direct communication, that is peer-to-peer communication. One of this type of real P2P file sharing networks is Napster [2], which first operated in 1999 and handles music files. Napster is actually the first real-world application of P2P network.

The second type are pure P2P networks. A pure P2P network consists of nodes that communicate with each other through direct connections, and does not include an index server. For example, nodes in P2P file sharing networks [15] provide files to each other through direct connections among the nodes. A direct connection between two nodes is represented by a logical network link, and therefore, a structure formed by logical links and nodes, that is, a P2P network topology, can be formed freely. Actually, according to [15], there are several forms in pure P2P networks. One form are structured P2P networks that have a mechanism to manage data locations in a network, This form forces nodes to form a specific network topology to reliably locate requested resources. The other form are unstructured P2P networks that do not have such a mechanism for reliably locating resources. One of the real unstructured P2P file sharing networks is Gnutella, which first operated in 2000. There are several Gnutella based networks such as LimeWire [1] and Phex [3]. In this book chapter, we will focus on such unstructured P2P networks.

The third type are super-node based P2P networks. This type are sort of hybrid P2P networks. However, super-node based P2P networks are not using a fixed index server but so-called super nodes. The super nodes are basically the same as the fixed index server in terms of functionality, but they are dynamically selected from among general nodes according to the current situation. In fact, hybrid P2P networks have the problem of single point failure because an index server is fixed. However, super-node based P2P networks are more flexible because they include a mechanism to vary super nodes equivalent to index servers in a dynamic fashion. One of the real applications of super-node based P2P networks is Skype [4], which is an application that enables voice and video calls and chats over the Internet and was first developed in 2003.

As mentioned above, here we focus on unstructured P2P networks. Unstructured P2P networks are flexible and robust because they do not rely on servers or super nodes at all. So, they are expected to have good scalability in terms of the number of nodes. However, since searching unstructured P2P networks for desired data is blind, unstructured P2P networks need some mechanisms to enhance search performance. To enhance search performance in unstructured P2P networks, query forwarding methods and data replication methods [7][8][16][14][24] have been investigated. The basic strategy to enhance search performance in unstructured P2P networks is that a data replication method places more data at easy reachable nodes by a given query forwarding method or that a query forwarding method forwards queries to nodes holding more data. However, enhancing search performance means that particular nodes are more frequently accessed, thus causing a load bias to particular nodes. This problem is actually similar to the problem of single point failure in client-server network services. So, it is important to consider a trade-off between search performance and load balancing in unstructured P2P networks.

Furthermore, a free-form P2P network topology can also sometimes be a control object for enhancing the quality or efficiency of P2P services. For example, a P2P network topology can be modified dynamically and adaptively in order to realize quick, accurate, and reliable searches. We here focus on topologies of unstructured P2P networks as a control object to realize quick, accurate, and reliable searches in the networks. To achieve that, we need a method for adaptively adjusting topologies of unstructured P2P networks.

One method that can adaptively optimize system parameters in general are is evolutionary algorithms (EAs), which were inspired by biological genetics and evolution [5]. One characteristic of EAs is that several solution candidates are held at any moment during an optimization process while a better solution is sought using these candidates. If we intend to use an EA as a method that adaptively optimizes a topology of a running P2P network, it is necessary to make several solution candidates coexistent, that is, several P2P network topologies, at any moment. Since, as mentioned above, a P2P network topology is a structure that is formed by logical links, it is possible to make several P2P network topologies coexisting at any given time. Therefore, it is also possible to adaptively change the topologies of the running P2P network by an EA. Although an EA is not the only way for adjusting the entire network topology of a running P2P network, an EA would be suitable for this case because an EA can hold several options (i.e. possible future network topologies) simultaneously in a running P2P network, which is a highly dynamic environment. EA has been used to optimize the parameter values of a P2P network using fitnesses obtained from a simulation model of the P2P network [17][25], but this EA is not an online approach to optimize the parameters of P2P networks. Other computational intelligence techniques such as artificial immune systems, neural networks, and memetic algorithms have also been applied to the resource discovery problem in P2P networks in an offline manner [10][9][19]. Besides P2P networks, EAs have been applied to structure optimization of communication networks in an offline manner [26][6][12] and also to on-line optimization of communication networks, such as on-line optimization of routing tables of routers in the Internet [18] and of protocol stacks [11]. Furthermore, there are a number of methods for local topology reconstruction in a sole P2P network topology based on the observation of local network states [13][23][22]. However, these methods cannot adjust the entire network topology for a given purpose.

In the present chapter, we introduce an evolutionary P2P networking technique [20] (referred to as **EP2P** hereinafter) whereby the topologies of a running P2P network are dynamically and adaptively modified by EA and show results of simulations to examine whether EP2P can provide reliable search capability in dynamic P2P environments. In simulations, we assume dynamic P2P environments in which nodes leave and join the network with their own probability and in which search objects vary with time. This technique is a fusion of an evolutionary algorithm (EA) and a P2P network, which optimizes in real time several P2P network topologies that every node belongs to at a given time under a given fitness function.

EP2P mentioned above, however, does not consider a large size of networks as seen in real P2P networks. For example, LimeWire [1], which is one of the real P2P file sharing networks was reported to hold several millions of users simultaneously. As will be mentioned later, EP2P needs a special node that plays a role of collecting fitnesses from nodes and executing EA to adaptively change the P2P network topologies, which is also called a super node but different from that in the super-node based P2P networks in terms of functionality. 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. Unlike EP2P, a parallel evolutionary P2P networking technique, which is referred to as **P-EP2P** [21], aims at realizing adaptive large-scale networks. P-EP2P first divides an entire 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. In the present chapter, we also introduce P-EP2P and show results of simulations to examine whether P-EP2P can provide reliable search capability for large-scale P2P networks.

The remainder of this chapter is organized as follows. In Section 2, we describe EP2P and Section 3 shows simulation results of EP2P. In Section 4, we describe P-EP2P and Section 5 shows simulation results of P-EP2P. Finally, Section 6 presents conclusions and describes areas for future research.

2 Evolutionary P2P Networking

The evolutionary P2P networking technique (**EP2P**) is a technique that evolutionarily reconstructs a set of P2P network topologies for a given purpose (see Figure 1). Generally, in evolutionary methods, the next generation of individuals is created from individuals in the present generation which are better adapted to the environment. In EP2P, P2P network topologies correspond to the individuals described above, and all of the nodes are included in the P2P network topologies. In addition, in EP2P, the fitness function for the EA is a P2P network that includes several P2P network topologies that are optimized by the EA, and the mechanism of topology reconstruction for the P2P network is the EA. Moreover, in EP2P, the set of P2P network topologies is not determined by the EA through simulations in advance. Rather, the fitness of each P2P network topology is obtained from nodes in the running P2P network, and a set of P2P network topologies is reconstructed by applying evolutionary operators to individuals that encode the topologies based on the fitnesses obtained while running the P2P network.

Next, we explain the details of the EP2P used herein.

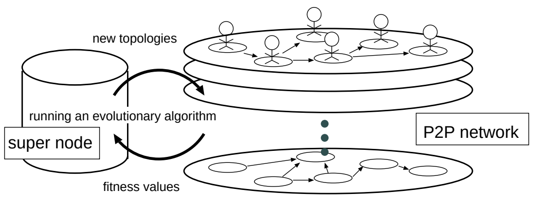


Fig. 1 Overview of the evolutionary P2P networking technique [20].

2.1 Network Composition

As shown in Figure 1, a network including 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 a super node, in which the EA is used to optimize the topologies.

The actual role of the super node is (1) to determine links for a node that joins the network for the first time, (2) to reconstruct the network topologies by executing the EA, and (3) to monitor which nodes join the network at each moment. We will describe how each network topology is assigned a fitness and how the network topologies are reconstructed later herein. Note that here the super node does not manage which services each node can provide to other nodes. For example, in a P2P file-sharing network, the super node does not manage which files each node holds.

2.2 Joining and Leaving Nodes

The P2P nodes communicate their joining and leaving of the network to the super node. Thus, the super node can determine which P2P nodes 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 network 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

A P2P node uses all network topologies including this node for a time period T and then assigns a fitness to each of the topologies. The fitness of each network topology is initially set to zero and again after 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 data or service in some topologies while not being able to find the service in other topologies. If the desired data or service is found in a certain topology, the fitness of the topology is increased by one. Otherwise, the fitness does not change. Although disconnected network topologies are usually not useful for the search, such network topologies are unlikely to be selected by the EA.

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. Then, the topologies with larger fitnesses are regarded as better in the EA used herein.

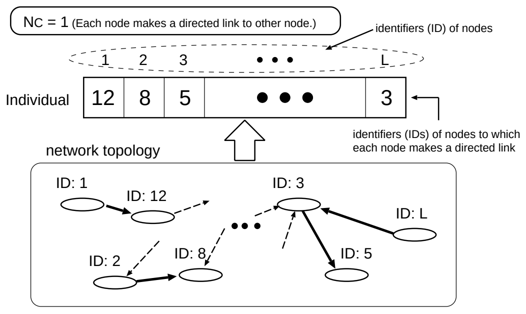
2.4 Representations of Network Topologies

In the EA, a solution candidate for an optimization problem is represented in an alternative form. This alternative form is designed by the 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

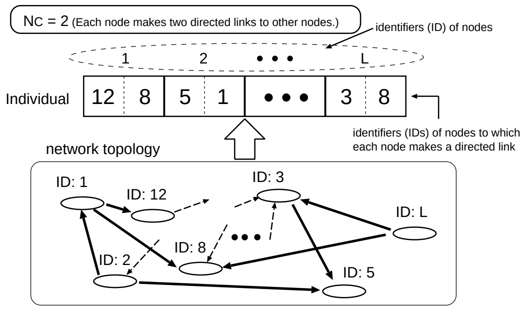
EP2P, the P2P network topology is an object of optimization and an individual is an alternative form of a P2P network topology.

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. The EA individual used here is shown in Figure 2. As shown in Figures 2(a) and 2(b), in which N_C is 1 and 2, respectively, the individual is a one-dimensional vector with $L \times N_C$ elements. An element in an individual is generally referred to as a gene in the EA.

Each node is assigned a serial number as its identifier, and the identifier corresponds to the index of the vector representing the individual. 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 a 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.


NC = 1 (Each node makes a directed link to other node.)
identifiers (ID) of nodes
1 2 3 • • • L
Individual
12 8 5 ● ● ● 3
identifiers (IDs) of nodes to which each node makes a directed link
network topology
ID: 1
ID: 12
ID: 3
ID: L
ID: 2
ID: 8
ID: 5

(a) $N_C = 1$.


NC = 2 (Each node makes two directed links to other nodes.)
identifiers (ID) of nodes
1 2 • • • L
Individual
12 8 5 1 • • • 3 8
identifiers (IDs) of nodes to which each node makes a directed link
network topology
ID: 1
ID: 12
ID: 3
ID: L
ID: 8
ID: 2
ID: 5

(b) $N_C = 2$.

Fig. 2 Representation of a P2P network topology in the EA (EA individual) [20].

2.5 Evolutionary Operators

Evolutionary operators are applied to the set of individuals mentioned above, 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, i.e., 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 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 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.

2.5.2 Crossover

The crossover operator used in the present study is hereinafter referred to as *node linkage crossover* (NLX). This operator is applied to an EA population as follows. Other type of crossover operators can also be used and conventional crossover operators will be used for comparison in Section 3.

1. N individuals selected by the selection operator are divided into $N/2$ pairs of individuals. The selected individuals become parent individuals in this generation.
2. The crossover operator is applied to each pair of parent individuals with probability p_c. 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 L 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 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.
   For example, suppose that $N_C = 1$, and the fifth element has been selected as the initial element of the linkage. Initially, NLX refers to the value of the fifth element of the parent individual as a copy source. If the reference value is 10,

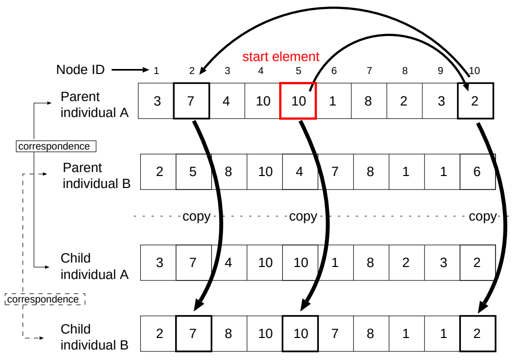
then NLX refers to the value of the tenth element. Furthermore, if the value of the tenth element is 2, then NLX refers to the value of the second element. By referencing the element values N_L times, NLX generates N_L element values and then copies them to the child individual corresponding to the other parent individual. In this example, N_L is 3, and the values of the reference elements are 5, 10, and 2, in that order. Nodes corresponding to the values of the reference element are linked by directed links. An example of this form of recombination is illustrated in Figure 3(a).
Figure 3(b) shows an example of NLX with $N_C = 2$. In Figure 3(b), 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 linkage is formed. Generally, when $N_C \geqq 2$, NLX is performed in the same manner.
6. Repeat Steps (3) through (5) $N_C \times L$ times.

2.5.3 Mutation

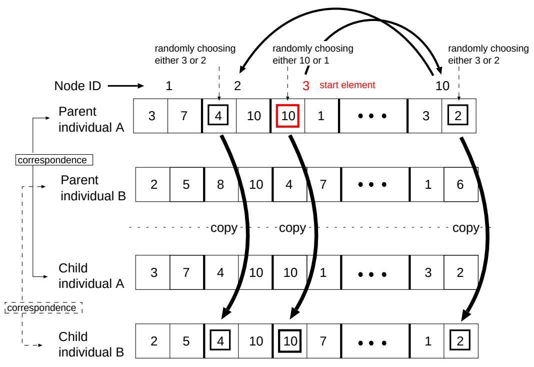
The mutation operator used herein is such that the value at each position (the gene) on the N individuals obtained after the node linkage crossover (NLX) is randomly changed to some other possible value with probability p_m, which is referred to as the mutation rate. This mutation operator is introduced 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 EP2P approaches to a random method.

2.6 Timing for Topology Generation

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.



(a) $N_C = 1$.



(b) $N_C = 2$.

Fig. 3 Example of node linkage crossover (NLX) [20].

2.7 Procedure and Parameters

First, N individuals, which are N network topologies, are randomly generated as the initial EA population. The generated individuals are used in performing the object search for time T and then proceed to the fitness evaluation phase. The fitness of each individual is the number of successful searches using the individual in time T. Next, the evolutionary operators are applied to the present EA population based on their fitnesses in order to generate a new EA population. The segment procedure, starting from the use of the individuals (topologies) and ending with the application of the evolutionary operators, is repeated. In EP2P, the procedure is as follows:

1. The super node randomly initializes the EA population.
2. The super node transforms the EA population into P2P network topologies and then informs each P2P node of the topology information.
3. Each P2P node makes directed links to other nodes.

4. Each P2P node uses all of the network topologies for time T and then assigns a fitness to each individual corresponding to a P2P network topology.
5. The super node applies the evolutionary operators to the EA population.
6. Back to (2).

3 Simulations of EP2P

In this section, we examine whether the EP2P presented in Section 2 can contribute to reliable search in dynamic P2P environments through simulations .

The focus of the simulations here is to examine in what situation an evolutionary way works well for adaptation of network topologies. Actually, it is possible to consider a method for adaptively constructing single P2P network topology using fitness from all of the nodes, which is not an evolutionary approach using multiple network topologies. However, such a proposal is beyond the focus of the present contribution. Instead of proposing such a method, we prepare a random topology construction method that is realized in the framework of the evolutionary approach by making the mutation rate higher.

3.1 Dynamic P2P Environmental Model

In the P2P simulation model used herein, P2P nodes that join the network vary with time. Each of the L nodes decides whether it will join the network at each time unit according to its given probability. This probability is hereinafter referred to as the *participation probability*. The participation probability of each node is determined as a uniform random real number in $(0,1]$. Each node joins the network with its participation probability. If a node does not join the network, then the node is in the state whereby the node leaves the network. After all of the nodes make a decision with regard to participation, each of the nodes conducts one search. A time unit is regarded as the period of time required for all of the nodes to make this decision and complete one search.

The parameter values of the EP2P used in the simulations are listed in Table 1. Some of the parameters take different values in the different evaluation scenarios that are described in detail later. A set of the parameter values shown in Table 1 is one of the possible sets. Other sets of the parameter values may yield different results from those obtained here. However, the focus of the simulations here is to examine in what situation an evolutionary way works well for adaptation of network topologies. For this purpose, most parameter values are fixed but only the mutation rate is changed to realize a random method for comparison of the evolutionary way, which does not rely on the previous good individuals. Thus, the crossover operator is considered to be the main driving force of evolution here. The effect of other types of crossover operators will be examined in Section 3.4.4.

Table 1 Parameter values of the EP2P used in the simulations.

Parameter	Description	Value
L	number of nodes (genes)	$2,000$
N	number of P2P network topologies (individuals)	50, 100 (for two evaluation scenarios)
T	time period for which generated topologies are used	50
H_{max}	maximum allowed number of hops for one search	3, 6 (for two evaluation scenarios)
N_C	number of directed links generated by a node	1
K	tournament size for the tournament selection	2
p_c	crossover rate	1.0
p_e	probability with which recombination is conducted for a selected element in NLX	0.1
N_L	length of node linkage in NLX	5
p_m	mutation rate	$0, 0.05, 0.1, 0.2, 0.5, 0.8, 1.0$

The parameter setting used herein brings the effect that on average, approximately half of the nodes (approximately $1,000$ nodes) join the network at any time.

3.2 Evaluation Scenarios

We prepare two types of evaluation scenarios that include different dynamic P2P environments. For any evaluation scenario, node departure and participation occurs as explained in the previous section. The time period of the simulation, which is 1 to $5,000$ time units, is the same for the two evaluation scenarios.

- Evaluation Scenario 1
  In the first evaluation scenario, replicas of the search objects are created in the network as unstructured file sharing networks. The replication method used here is such that a search object that was retrieved by the node making the query is stored in the storage of the node and will thereafter be shared among all nodes. This method is referred to as owner replication.
  The storage capacity for each node is determined as a uniform random integer in $[50, 300]$. Any one search object consumes one unit of storage capacity.
  There are $2,000$ types of search objects, which is the same as the value of L shown in Table 1. Initially, each node has one type of search object. However, the objects for which nodes actually search vary with time. As mentioned above, the total simulation time period is $5,000$ time units. From $1000(k-1)+1$ to $1000k$ time units, objects possessed by the $(400(k-1))$-th through $400k$-th nodes become search objects, where $k = 1, 2, \cdots, 5$. The search object that each node will search for is determined randomly from among the 400 current search objects that each node does not possess.
  In the first evaluation scenario, the number of coexisting network topologies is 50, and the number of hops allowed for one search is 3. Therefore, it is possible

to forward a search query to nodes a maximum of $150(= 50 \times 3)$ times during one search.

- Evaluation Scenario 2
  In the second evaluation scenario, a replication method is not used. The search objects are P2P nodes, so storage capacity need not be considered. From 1 to 2,500 time units, the search objects are two of the 1st through 2,500th nodes. The two nodes as search objects have a participation probability that is greater than or equal to 0.95. One of the two nodes searches for the other node as a search object. Other nodes search for one of the two nodes selected randomly.
  Although the P2P network applications to which this evaluation scenario corresponds differs from the first evaluation scenario in the following aspects: there is no replication, and there are few search objects at any given moment. In the second evaluation scenario, the number of coexisting network topologies is 100, and the number of hops allowed for one search is six. Therefore, it is possible to forward a search query to nodes a maximum of $600(= 100 \times 6)$ times during one search.

3.3 Observations

As mentioned above, the fitness of each network topology to which P2P nodes are assigned is the number of successful searches using the network topology. Therefore, it is expected that EP2P will evolve network topologies that have a low rate of search failures.

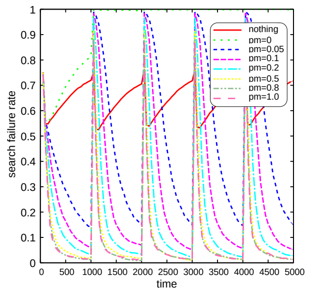
Thus, we consider the search failure rate of the present network topologies during a period of use, which is $T = 50$, as an observation item. Search failure occurs when a requested object cannot be located using all of the coexisting network topologies. We observe the number of search failures in all of the searches conducted during 50 time units and then calculate the search failure rate every 50 time units based on the observed number of search failures.

In addition, we observe the average number of hops for only successful searches in the network topologies during a period of use ($T = 50$). The average number of hops is also obtained every 50 time units.

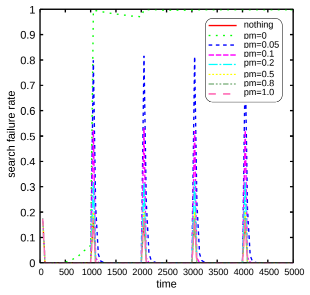
3.4 Results

Simulation results for the first and second evaluation scenarios are shown in Figures 4 and 5, respectively. The results are averages over 10 independent simulation runs. Figures 4(a) and 5(a) show the time-varying search failure rates for the first and second evaluation scenarios, respectively, which assume dynamic P2P environments, and show the results obtained for seven mutation rates (p_m) and the results for the case in which the evolutionary operators are not applied (labeled “nothing”). For comparison, Figures 4(b) and 5(b) show the time-varying search failure rates for the first and second evaluation scenarios, respectively, that exclude node participation and departure. In other words, all of the nodes are in the network at any time. In addition, Figures 4(c) and 5(c) show the time-varying average number of hops only for successful searches for the first and second evaluation scenarios, respectively.

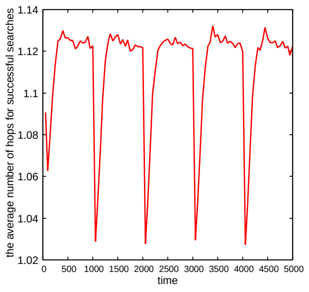
In addition, the average search failure rate over ten independent simulation runs and the standard deviation of the ten search failure rates at the end of simulation run (at 5,000 unit times) are shown in Table 2. The average and the standard deviation of the search failure rates are obtained for both of the first and the second evaluation scenarios including node participation and departure. Since Table 2 shows that the standard deviation is quite small both in the first and the second evaluation scenarios, we will discuss the performance of the EP2P using the average search failure rate below.



(a) Time-varying search failure rate.

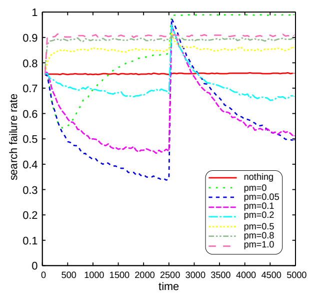


(b) Time-varying search failure rate when no node participation or departure occurs.

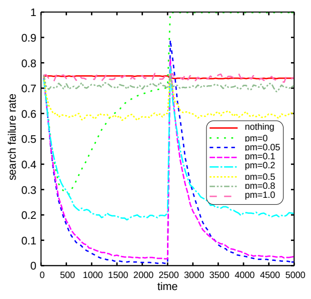


(c) Time-varying average number of hops for successful searches when the mutation rate, p_m, is 1.0.

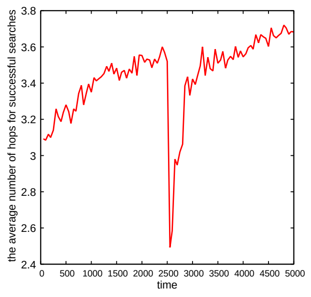
Fig. 4 Results of the first evaluation scenario, in which there are several types of search objects and a replication method [20].



(a) Time-varying search failure rate.



(b) Time-varying search failure rate when no node participation or departure occurs.



(c) Time-varying average number of hops for successful searches when the mutation rate, p_m, is 0.05.

Fig. 5 Results of the second evaluation scenario, in which there are few types of search objects and there is no replication method [20].

Figures 4(a) and 5(a) show that in all of the evaluation scenarios including different dynamic P2P environments, EP2P lowers the search failure rate with time just after the search objects are varied depending on the values of the mutation rate.

Furthermore, Figures 4(c) and 5(c) show that for the situation in which the search failure rate decreases with time, the average number of hops for successful searches increases, because the variance of the number of hops for successful searches becomes larger in processes in which the network topologies become reliable for more nodes. The number of the network topologies would be too small to meet demands of many nodes at once.

We discuss the results for each evaluation scenario below.

Table 2 The average search failure rate over ten independent runs and the standard deviation of the ten search failure rates at the end of the simulation run (at 5,000 time units) in the first and the second evaluation scenarios including node participation and departure. The label of "nothing" in the table represents no evolutionary operators.

The first evaluation scenario								
Value of p_m	nothing	0.0	0.05	0.1	0.2	0.5	0.8	1.0
Average search failure rate	0.717	0.999	0.148	0.052	0.028	0.013	0.013	0.012
Standard deviation	0.006	0.0002	0.011	0.010	0.013	0.011	0.006	0.004

The second evaluation scenario								
Value of p_m	nothing	0.0	0.05	0.1	0.2	0.5	0.8	1.0
Average search failure rate	0.758	0.988	0.496	0.513	0.666	0.857	0.891	0.912
Standard deviation	0.016	0.016	0.044	0.061	0.044	0.013	0.010	0.009

3.4.1 First Evaluation Scenario

Figure 4(a) shows that the higher the mutation rate, the lower the search failure rate. In addition, when the mutation rate, p_m, is 1.0, that is, when the network topologies are fully randomly reconstructed every 50 time units, the search failure rate is quickly minimized. This result means that a continuous large change in network topologies is superior to changing network topologies through an evolutionary method, which yields new individuals from the better individuals of the present generation, in terms of providing reliable search capability.

The reason for this result is as follows. Suppose that a node could find a requested object using the present network topologies at some moment. Then, the node will no longer search for that requested object. Therefore, there is no guarantee that the network topologies that contributed to a successful search for a certain object will be useful for finding other objects later. In addition, the possibility exists that the locations of objects in the network change due to replication of the objects. From the viewpoint of the EA, the fitness landscape, which represents the distribution of the fitnesses on the search space, changes dynamically. In such an unstable situation in terms of the fitnesses of the EA individuals, it is difficult for the EA to evolutionarily identify better individuals and adapt them to the problem.

In the first evaluation scenario, if the evolutionary operators are not used, the topologies of the P2P network are not changed over time. In such a situation, the replication brings improvement of the search failure rate, but once most nodes have had search objects that are located among their reachable nodes, the search failure rate becomes worse with time. Therefore, we can observe in Figure 4(a) that the graph of the search failure rate for "nothing", which represents not using the evolutionary operators at all, is first improved with time and then degraded with the time needed until the search objects are totally changed. Meanwhile, when no node participation and departure occurs, there are more reachable nodes for each node even when not using the evolutionary operators. Therefore, the degradation of the search failure rate is not observed as shown in Figure 4(b).

If the mutation rate is zero in the first evaluation scenario, novel genes that were not supplied in the initial population never appear. Therefore, the selection operator forces the population to converge to one individual. Once such convergence of the network topologies occurs, the search failure rate is hardly improved. Therefore, we can observe in Figures 4(a) and 4(b) that the graph of the search failure rate for $p_m = 0$ is first improved and then degraded over the simulation time.

3.4.2 Second Evaluation Scenario

Figure 5(a) reveals that when the mutation rate is lower, but greater than zero, the search failure rate becomes lower. The results indicate that an evolutionary method, which produces the individuals for the next-generation from the better individuals of the present generation, is effective in this evaluation scenario. In this scenario, the locations of search objects in the network, i.e., the locations of nodes, do not vary because there is no replication, so that the fitness landscape in the EA is static. In addition, there are only two search objects, so that it is possible to somehow evolve network topologies that can lower the search failure rate, although the crossover operator might produce useless network topologies by recombining network topologies that are specialized for different search objects. In order to evolve network topologies more quickly that provide a more reliable search, it is necessary to evolve several coexisting network topologies specific for different search objects. In other words, diverse network topologies should be maintained in a single population.

If the mutation rate is zero in this second evaluation scenario, the selection operator forces the population to converge to one individual as in the first evaluation scenario. Therefore, we can observe in Figures 5(a) and 5(b) that the graph of the search failure rate for $p_m = 0$ is first improved and then degraded over the simulation time.

In this scenario, the quickest and most reliable search can be achieved using network topologies in which each node makes directed links to nodes as search objects. However, it is difficult for the EA considered herein to generate such network topologies from randomly initialized network topologies.

In present state there is no P2P application that fits the second evaluation scenario well. Especially, it is not realistic to assume that only few search objects exist in P2P networks at any moment. Meanwhile, no replication is possible in P2P applications whose search objects are nodes (users). Therefore, in future work, we need to seek a way to maintain diverse network topologies that are specialized for many sorts of search objects in a single population, to apply the EP2P to P2P applications in which search objects are nodes (users), there are many search objects at any moment, and replication of search objects is not conducted.

3.4.3 EP2P in Dynamic and Static P2P Environments

Figures 4(a) and 4(b) show that although the rate at which the search failure rate decreases for the EP2P with the same parameter values differs for the dynamic and static P2P environments in terms of node participation and departure, the tendency of convergence of the search failure rate is roughly the same for the dynamic and static P2P environments for the first evaluation scenario. In addition, Figures 5(a) and 5(b) indicate that for the second evaluation scenario, the tendency of convergence of the search failure rate is roughly the same between the EP2Ps with the same parameter values in the dynamic and static P2P environments.

3.4.4 Other Types of Crossover Operators

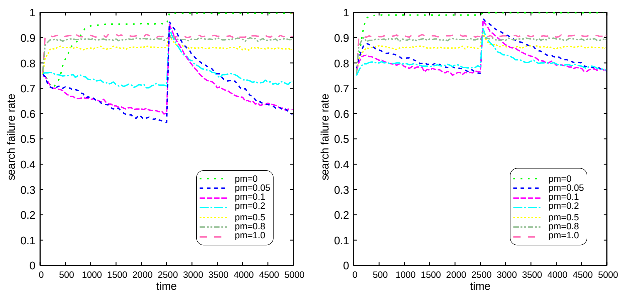
We use other crossover operators in the EP2P and examine if those crossover operators can improve the results of NLX presented in this paper. The crossover operators used herein are one-point crossover and uniform crossover. Both of them are conventional and common in the evolutionary computation research field.

Both of the two crossover operators are applied with probability p_c to each of $N/2$ pairs of individuals that are formed from the selected N individuals. The one-point crossover first randomly selects a position on two parent individuals (vectors) and then exchange the values (genes) beyond that randomly selected position. The uniform crossover exchanges two values at each position on two parent individuals between the two individuals with probability of 50 %.

While NLX presented in this paper is meant to increase good node linkage in the population, the one-point crossover and the uniform crossover are not. Therefore, we can expect that NLX considering good node linkage yields better results than those two crossover operators. We apply the EP2P with the one-point crossover and the uniform crossover to the evaluation scenario 2 mentioned above, in which the EP2P with NLX and low mutation rate worked better than the periodic random topology reconstruction. We examine if the one-point crossover or the uniform crossover improves the result of the EP2P with NLX. The seven different values of mutation rate are also used here.

The simulation results are shown in Figure 6. Figure 6 does not include the result for the case in which the evolutionary operators are not applied. Comparing Figure 6 with Figure 5 that represents the result of using NLX, the tendency of the results is so similar among the EP2P with the one-point crossover, the uniform crossover, and NLX, but the search failure rate for the case of using small mutation rate as $p_m = 0.05, 0.1$ is different. The EP2P is shown to be the best when using $p_m = 0.05, 0.1$. Since a main driving force to obtain better individuals is a crossover operator when the mutation rate is quite low, it is suggested that NLX considering the node linkage can produce better individuals than the two conventional crossover operators that do not consider the node linkage.

Although NLX is shown to be better than the two conventional crossover operators above in EP2P, the performance of NLX may depend on values of N_L. We will



(a) The result for one-point crossover. (b) The result for uniform crossover.

Fig. 6 Time-varying search failure rate when using EP2P using other crossover operators [20].

investigate the effect of N_L value on the performance in future work when considering guidelines on the parameter values settings.

3.5 Using a Different Type of Network Topology

The simulation results presented above indicate that EP2P can provide reliable search capability in the second scenario. In other words, it is shown that topology reconstruction by EP2P is better than random topology reconstruction when only a few types of search objects are present in the network at any moment and these search objects are not replicated. In the EP2P used above, the number of directed links that each node makes to other nodes in a single network topology, N_C, is just one. As mentioned in Section 2, it is possible for every node to hold any number of directed links (N_C) in a network topology. In this section, we examine whether the EP2P that allows nodes to hold several directed links in a network topology can improve the results of the second scenario shown in Section 3, in which the EP2P was better than periodical random topology reconstruction.

Concretely, we compare two different EP2P with different configurations. In one EP2P, the number of directed links of each node, N_C, is set to be two, and the depth of the flooding tree, H_{max}, is three. In the other EP2P, the number of directed links of each node is set to be one, and the depth of the flooding tree is eight. In both of the EP2Ps, the number of times query forwarding is performed in a network topology in one search is eight. The application of the node linkage crossover (NLX) to the individuals was described in Section 2. The EP2P with $N_C = 2$ and the EP2P with $N_C = 1$ use different values for p_e and N_L to obtain the same expected numbers of element values that are copied by NLX between the two EP2Ps. Specifically, the EP2P with $N_C = 2$ takes $p_e = 0.2$ and $N_L = 3$, and the EP2P with $N_C = 1$ takes $p_e = 0.1$ and $N_L = 6$. In addition, the mutation rate for the two EP2Ps is the same as $p_m = 0.05$, which yielded the best result in the second scenario.

Figure 7 shows the time-varying search failure rate for the two different EP2Ps. The obtained results are the averages over 10 independent simulation runs. In addition, we show the results for the two EP2Ps, which exclude node participation and departure, in Figure 7.

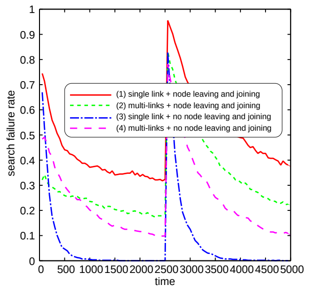


Fig. 7 Time-varying search failure rate when using EP2P with $N_C = 1$ and $N_C = 2$ [20].

Figure 7 shows that, in the second evaluation scenario including node joining and leaving, the EP2P with $N_C = 2$ achieves a lower search failure rate than the EP2P with $N_C = 1$. This suggests that the EP2P that allows nodes to hold several directed links is more suitable for dynamic P2P environments. In contrast, for the case in which node joining and leaving does not occur, the EP2P with $N_C = 1$ achieves a lower search failure rate than the EP2P with $N_C = 2$. These results indicate that the number of directed links that achieves more reliable search depends on how often node joining and leaving occurs.

4 Parallel Evolutionary P2P Networking

In the previous section, EP2P is shown to be able to adaptively reconstruct network topologies of a running P2P network. As mentioned already, EP2P needs a special node called a super node that plays the role of collecting fitness values from nodes and executing EA to adaptively change the P2P network topologies, but due to that, the super node may be overloaded as the number of nodes increases and then the P2P network may stop working. That is, EP2P has a scalability issue in terms of the number of nodes.

So, in this section, we introduce a parallel evolutionary P2P networking technique (P-EP2P) for realizing adaptive large-scale networks. The overview of P-EP2P is shown in Figure 8. P-EP2P first divides an entire network into several smaller networks to avoid the overload of a super node and then applies EP2P to each small network in parallel to make the entire network adaptive. 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 ($k = 1, 2, \ldots, N_G$). In this contribution, we do not, however, discuss a process in which all of nodes are divided into multiple node groups, but assume that they are divided into multiple groups in advance. In addition, we do not discuss how to select a super node in each node group. However, we can easily find ways to obtain multiple node groups, such that the maximum number of nodes that each node group can hold is decided in advance and a node joining the network is assigned to one of available node groups.

P-EP2P, as mentioned before, divides all of nodes into multiple node groups and then applies EP2P to each group. We will explain P-EP2P by describing the difference between EP2P and P-EP2P. As mentioned above, we assume that all of the nodes are in advance divided into multiple groups. Then, the differences are the following two points: (1) how multiple super nodes gather fitnesses and (2) the evolutionary operators used. We here assume that the multiple super nodes are able to know only which nodes are in the network at any moment by exchanging information on the present nodes among them and also that every super node knows only links created by nodes in its node group.

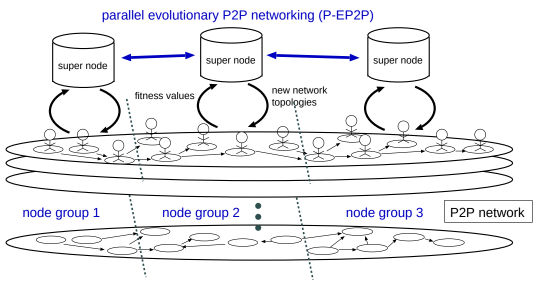


Fig. 8 Overview of the parallel evolutionary P2P networking technique [21].

4.1 Gathering Fitnesses by Super Nodes

In EP2P, one super node gathers fitnesses from all of the nodes. Meanwhile, in P-EP2P, one super node is assigned to each of the N_G node groups and gathers fitnesses only from G_k ($k = 1,2,\ldots,N_G$) nodes that belong to its node group. All L 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 identical network topologies.

4.2 Evolutionary Operators

In a super node of each node group, evolutionary operators are applied only to nodes of the group.

First, 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, as mentioned above, selected individuals can be of different value in the node groups.

Second, NLX is applied to the selected individuals by the tournament selection in each node group. 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. 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 case that those nodes belongs to the same node group, the linkage of $a_1 \rightarrow b_2 \rightarrow c_3$ can be copied. However, in case that those nodes belong to different node groups, 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 Q 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 can not 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. One attempt of NLX is finished when the total number of the vector elements copied becomes Q.

Finally, the mutation operator used herein changes each element value of the individual to one of the possible values at random, but whether the change occurs is determined with a given 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.

5 Simulations of P-EP2P

In this section, we examine the relationship between the number of node groups and adaptability of network topologies.

5.1 Simulation Model and Configurations

To focus only on the examination of the relationship between the number of node groups and the adaptability of network topologies as much as possible, we simplify other things. We consider that all of the nodes are always present in the network without leaving during the simulation period. In addition, it is assumed that all of the nodes excluding a node as a search object search the entire network for only one node to receive some service over the simulation period of time. The node as a search object does not conduct search. The above-mentioned assumptions are not practical, but the network in the simulation model is static and therefore changes in the network topologies affect the adaptability of the network topologies. A time unit is regarded as the period of time required for all of the nodes to complete one search.

The parameter values of the P-EP2P used in the simulations are listed in Table 3. The parameters whose values are changed in the simulations are just the number of nodes L, which takes 10^3, 10^4, 10^5, and the number of node groups N_G, which takes 1, 2, 5, and 10. Also, the number of nodes as a search object is just one and all of the nodes excluding the search object conduct only one search in a time unit, so that the search result for every node is always the same until the network topologies are changed by applying the evolutionary operators to all of the node groups at the same time.

Table 3 Parameters values of P-EP2P used in the simulations.

Parameter	Description	Value
L	number of nodes (genes)	10^3, 10^4, 10^5
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%
Q	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

5.2 Simulation Results

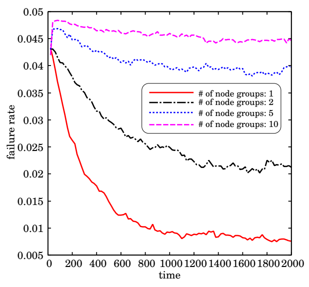
We observed the search failure rate of the present network topologies during a period of use, which is $T = 20$. Figure 9 shows the time-varying search failure rates for a variety of the number of nodes and the number of node groups. The results shown in Figure 9 are averages over 10 independent simulation runs.

We can observe from Figure 9 that the degree of improvement in the search failure rate becomes smaller as the number of the node groups increases, for any number of nodes. One reason for this would be that NLX changes only a link between nodes in a node group of focus or a link from a node in a node group of focus to a node in other node group. In case that a node as a search object belongs to another node group and the evolutionary operators used cannot create a direct link from nodes in the node group of focus to the node as the search object, multiple links from nodes in the node group of focus to the node as the search object via several nodes in other groups need to be generated and maintained. However, such a linkage of nodes via several node groups cannot be created by the evolutionary operators conducted in one node group. Another reason would be that multiple node groups can assign different fitnesses to the identical network topology. If multiple node groups do so, it is hard for the evolutionary operators to create a linkage between a node as a search object and a node searching for the search object via several node groups.

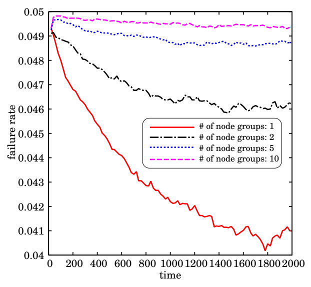
Figure 10(a) shows an example of time-varying ratios of nodes linking to nodes in other node groups when the number of nodes is 10^3. We can observe from Figure 10(a) that the ratio is almost the same over the simulation period of time. This would be because novel links from nodes in a certain node group to other node groups are brought only by the mutation operator, which also changes such a link to a link between nodes in one node group. Figure 10(b) shows an example of time-varying fitnesses that are assigned to the identical network topology by all of node groups when the number of node groups is five and the number of nodes is 10^4. We can observe from Figure 10(b) that the five node groups gave different fitnesses to the same network topology for most of the time and also that the difference between the fitnesses assigned by the five groups became larger with the time. In addition to this fact, probabilistic procedures included in the selection operator would make it hard to evolve the multiple network topologies consistently.

6 Conclusion

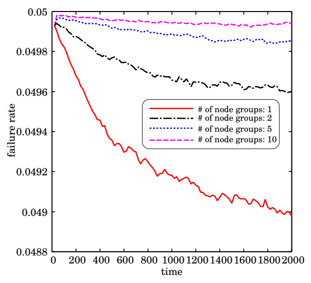
In the present chapter, we introduced an evolutionary P2P networking technique (EP2P) that dynamically and adaptively optimizes several P2P network topologies, in which all of the nodes are included at the same time, in an evolutionary manner. In addition, for two different evaluation scenarios, we examined through simulations whether the EP2P can provide reliable search capability in dynamic P2P environments, including participation and departure of nodes and time-variation of search objects. The simulation results suggested (1) that, with respect to reliable search capability, random topology reconstruction is better than topology reconstruction by EP2P when several types of search objects are present in the network at any moment and replicas of these search objects are created in the nodes, and (2) that topology reconstruction by the EP2P is better than random topology reconstruction



(a) $L = 1000$.

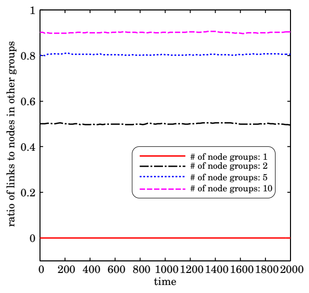


(b) $L = 10000$.

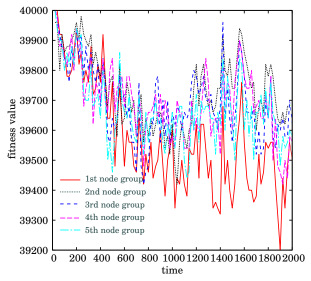


(c) $L = 100000$.

Fig. 9 The time-varying search failure rates for different number of node groups [21].



(a) Example of time-varying ratios of nodes linking to nodes in other node groups when the number of nodes is 1000.



(b) Example of time-varying fitnesses that are assigned to the identical network topology by all of node groups when the number of node groups is five and the number of nodes is 10000.

Fig. 10 Examples for time variation of observed values [21].

when only a few types of search objects are present in the network at any moment and these search objects are not replicated. Moreover, for the scenario in which the EP2P was effective, we showed through simulations that when each node makes not one but several directed links to other nodes in a single network topology, the EP2P improves the reliable search capability. Finally, the number of directed links that yield more reliable search capability was found to depend on how often nodes leave and join the network.

The present chapter also introduced the parallel evolutionary P2P networking technique (P-EP2P) for bringing adaptability to large-scale networks. P-EP2P first divides an entire 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. The results of the simulations for evaluating P-EP2P suggested that there is a trade-off relationship between load balancing among super nodes and search reliability.

In order for EP2P to more quickly evolve network topologies that provide reliable search capability and to extend the circumstances in which the EP2P can induce evolutionary adaptation of network topologies, we will consider a method for holding diverse network topologies simultaneously that are specialized for different search objects. In addition, guidelines on the parameter values settings and an investigation of the relationship between the parameter values and the performance of EP2P are necessary. We also need to more clearly and practically define performance metrics for the load balancing among super nodes as well as for the search reliability for simulation studies of P-EP2P, and then seek methods especially for maintaining useful network topologies under these two performance metrics. Finally, although we showed the simulation results for evaluating the online approaches, EP2P and P-EP2P, in the present chapter, we need to evaluate EP2P and P-EP2P through online real experiments.

Acknowledgements The present study was supported by the Japan Society for the Promotion of Science through a Grant-in-Aid for Young Scientists (B) (22700077)

References

1. LimeWire. http://www.limewire.com/
2. Napster. http://www.napster.com/
3. Phex. http://www.phex.org/
4. Skype. http://www.skype.com/
5. Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press (1996)
6. Cheng, S.T.: Topological optimization of a reliable communication network. IEEE Transactions on Reliability **47**(3), 225–233 (1998)
7. Clarke, I., Snadberg, O., Wiley, B., Hong, T.W.: Freenet: A distributed anonymous information storage and retrieval system. In: Proceedings of Workshop on Design Issue in Anonymity and Unobservability. International Computer Science Institute, Berkeley, CA, USA (2000)

8. Cohen, E., Shenker, S.: Replication strategies in unstructured peer-to-peer networks. In: Proceedings of the ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. Pittsburgh, PA, USA (2002)
9. Das, T., Nandi, S., Deutsch, A., Ganguly, N.: Bio-inspired search and distributed memory formation on power-law networks. In: Proceedings of 10th International Conference on Parallel Problem Solving from Nature (PPSN X), pp. 154–164 (2008)
10. Ganguly, N., Canright, G., Deutsch, A.: Design of an efficient search algorithm for P2P networks using concepts from natural immune systems. In: Proceedings of 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), pp. 491–500 (2004)
11. Imai, P., Tschudin, C.: Practical online network stack evolution. In: SASO 2010 Workshop on Self-Adaptive Networking (2010)
12. Knowles, J., Corne, D.: A new evolutionary approach to the degree-constrained minimum spanning tree problem. IEEE Transactions on Evolutionary Computation **4**, 125–134 (2000)
13. Koo, S.G.M., Lee, C.S.G., Kannan, K.: A genetic-algorithm-based neighbor-selection strategy for hybrid peer-to-peer networks. In: the 13th Intl. Conference on Computer Communications and Networks (ICCCN 2004), pp. 469–474 (2004)
14. Letian Rong: Multimedia resource replication strategy for a pervasive peer-to-peer environment. Journal of Computers **3**(4), 9–15 (2008)
15. Lua, E.K., Crowcroft, J., Pias, M., Sharma, R., Lim, S.: A survey and comparison of peer-to-peer overlay network schemes. IEEE Communications Surveys & Tutorials **7**(2), 72–93 (Second Quarter 2005)
16. Lv, Q., Cao, P., Cohen, E., Li K. Shenker, S.: Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16th international conference on Supercomputing, pp. 84–95. New York, USA (2002)
17. Merz, P., Wolf, S.: 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 IX), pp. 272–281 (2006)
18. Munetomo, M., Takai, Y., Sato, Y.: An adaptive network routing algorithm employing path genetic operators. In: Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 643–649 (1997)
19. Neri, F., Kotilainen, N., Vapa, M.: A memetic-neural approach to discover resources in P2P networks. Studies in Computational Intelligence **153**, 113–129 (2008)
20. Ohnishi, K., Oie, Y.: Evolutionary P2P networking that fuses evolutionary computation and P2P networking together. IEICE Transactions on Communications **E93-B**(2), 317–328 (2010)
21. Ohnishi, K., Oie, Y.: Parallel evolutionary P2P networking for realizing adaptive large-scale networks. In: The Second Workshop on Heuristic Methods for the Design, Deployment, and Reliability of Networks and Network Applications (HEUNET 2011) (SAINT 2011 Workshop), 6 pages (2011)
22. Pournaras, E., Exarchakos, G., Antonopoulos, N.: Load-driven neighbourhood reconfiguration of gnutella overlay. Computer Communications **31**(13), 3030–3039 (2008)
23. Srivatsa, M., Gedik, B., Liu, L.: Large scaling unstructured peer-to-peer networks with heterogeneity-aware topology and routing. IEEE Transactions on Parallel and Distributed Systems **17**(11), 1277–1293 (2006)
24. Thampi, S.M., Chandra, S.K.: Review of replication schemes for unstructured P2P networks. In: Proceedings of IEEE International Advance Computing Conference IEEE (IACC'09), pp. 794–800. Patiala, India (2009)
25. Walkowiak, K., Przewoniczek, M.: Modeling and optimization of survivable P2P multicasting. Computer Communications **34**(12), 1410–1424 (2011)
26. Zhou, G., Gen, M.: A note on genetic algorithm approach to the degree-constrained spanning tree problems. International Journal of Networks **30**(2), 91–95 (1997)