Evolutionary P2P Networking for Enhancing Search Performance

Kei Ohnishi, Kento Ichikawa, and Yuji Oie
Kyushu Institute of Technology
680-4 Kawazu, Iizuka-shi, Fukuoka 820–8502, Japan
email: ohnishi@cse.kyutech.ac.jp, ichikawa@ndrc.kyutech.ac.jp, oie@cse.kyutech.ac.jp

Abstract—The present paper proposes a concept of evolutionary peer-to-peer (P2P) networking in which topologies of a running P2P network are dynamically and adaptively modified by an evolutionary algorithm and an algorithm based on the proposed concept. In addition, we evaluate the proposed concept through simulations. Evolutionary P2P networking allows every node to simultaneously belong to several topologies of a running P2P network. Each node assigns a fitness value to each of the P2P network topologies according to the result of using each network topology. The set of P2P network topologies are reconstructed by applying evolutionary operators to individuals encoding them in a server. In the simulation, nodes are search objects and the number of hops required for finding requested nodes on a certain network topology is the fitness value of the network topology. The simulation results show that the proposed algorithm for evolutionary P2P networking can generate a network topology that provides high search performance for most nodes when object nodes for search are strongly biased.

I. INTRODUCTION

A Peer-to-Peer (P2P) network consists of nodes that communicate with each other using direct connections. In P2P networks, nodes are not assigned a fixed role and so can become both client and server. In addition, the nodes can provide services for other nodes. For example, nodes in P2P file sharing networks [1] provide files to each other using 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 basically be formed freely. A free-formed P2P network topology can sometimes be a control object for enhancing the quality or efficiency of P2P services. For example, a method for searching a P2P network for nodes providing services is required because all services in P2P networks are managed in a decentralized manner, and it is thought that a P2P network topology can be modified dynamically and adaptively in order to realize quick, accurate, and reliable searching.

One method that can adaptively optimize system parameters is an evolutionary algorithm (EA) inspired by biological genetics and evolution [2]. One of the characteristics of EA is to hold several solution candidates at any moment during an optimization process and to search for a better solution utilizing these candidates. If we intend to use EA as a method that adaptively optimizes a topology of P2P network, it is necessary to hold several solution candidates, that is, several P2P network topologies, at any moment. Since, as mentioned above, a P2P network topology is a structure formed by logical links, it is possible to hold several P2P network topologies at any one time. Therefore, it is also possible to adaptively change the running P2P network topologies by EA.

In the present paper, we propose a concept whereby topologies of a running P2P network are dynamically and adaptively modified by EA and realize a concrete algorithm based on the proposed concept. In addition, we conduct a basic evaluation of the realized algorithm through simulations. We refer to the proposed concept as evolutionary P2P networking.

As for the dynamic reconstruction of a single P2P network topology, there exist dynamic topology reconstruction methods for unstructured P2P networks to improve defined performance or to explore trade-off between different performances [3][4]. However, thus far, there has been no report of a topology reconstruction method that dynamically and adaptively modifies several network topologies to improve defined performance. In addition, EA has been used to optimize the parameters of a P2P network based on fitness values obtained from a P2P simulation model [5]. Meanwhile, there has been no report of an approach that makes several sets of parameter values of a P2P network to exist together in the P2P network and evolutionarily optimizes the sets of parameter values of the P2P network based on fitness values obtained from the real P2P network as the evolutionary P2P networking presented in this paper.

The remainder of the present paper is organized as follows. In Section II, we propose the concept of the evolutionary P2P networking. In addition, we present a concrete algorithm realized based on the proposed concept. Section IV shows the evaluation results of the proposed concept. Section V presents conclusions and describes areas for future research.

II. CONCEPT OF EVOLUTIONARY P2P NETWORKING

The concept of evolutionary P2P networking is that EA as an optimization technique dynamically and adaptively reconstructs a set of topologies of a running P2P network for a given purpose (see Figure 1). In this concept, a set of P2P network topologies are not determined in advance by EA through simulations. Rather, a fitness value of each P2P network topology is obtained from nodes in the real used P2P network, and a set of P2P network topologies are reconstructed by applying evolutionary operators to individuals encoding
them based on their obtained fitness values while running the P2P network. EA is implemented in a server.

![Network topologies (population) at time t](image)

Fig. 1. Concept of evolutionary P2P networking.

A set of solution candidates used by EA, that is an EA population consists of a fixed number of P2P network topologies. Each EA individual, that is each P2P network topology forming the population is represented by a vector. The P2P network link assumed herein represents a path that indicates the direction of information flow. This means that a structure formed by P2P network links and a mechanism for propagating information on the structure are not separated, and the information flow depends entirely on the structure.

Each node belongs to all coexisting P2P network topologies. Fitness values are assigned to the P2P network topologies by all of the nodes. For example, the search performance or access load that the nodes experienced can be used as the fitness values. In addition, evolutionary operators such as selection, crossover, and mutation are used as methods for reconstructing P2P network topologies.

III. ALGORITHM

In this section, we realize an algorithm according to the concept presented in Section II.

A. Representation of Individuals

Suppose a P2P network is composed of $L$ nodes. An EA individual encodes a P2P network topology in which each node makes a directed link to another node. The EA individual is, as shown in Figure 2, represented as a vector of $L$ elements. The number of elements of this vector corresponds to a sequence ID number of a node, and the value of an element corresponding to a node is an ID number of another node to which the node makes a directed link. The directed link that each node makes to another node indicates that information can be propagated only in the direction of the link. An EA population consists of $N$ individuals, that is, the population size is $N$.

Although a node makes a link only to a node in the presented algorithm, it is also possible to allow a node to make several links to other nodes and represent such a network topology as an EA individual.

![Individual](image)

Fig. 2. Representation of Individuals.

B. Fitness Values for Individuals

Suppose that nodes search a P2P network for objects such as nodes and files. A fitness value for an individual, which is a P2P network topology, is the average number of hops over all searches that all of the nodes conducted using that individual in time period $T$. In a search, a search query is forwarded to all of the coexisting $N$ individuals, and, as a result, the numbers of hops for all individuals are obtained. A unit of time is set as the time period in which all of the $L$ nodes perform object search only once, so that $L \times T$ searches are performed using each individual in time period $T$.

Also, Time To Live (TTL), which is the number of hops allowed for a search using each individual, is $H_{max}$. If a target object cannot be found within $H_{max}$ hops, the number of hops recorded by a node generating the search query is $H_{max} + B$, where $B$ is a constant.

C. Evolutionary Operators

The selection operator used in the proposed algorithm is a tournament selection with a tournament size of $K$. The tournament selection randomly selects $K$ individuals from the EA population and selects an individual with the best fitness value among the $K$ individuals. This selection procedure is repeated until $N$ individuals have been selected.

The crossover operator used in the proposed algorithm is a uniform crossover. The uniform crossover is applied with probability $p_c$ to each of $N/2$ pairs of individuals that are formed from the selected $N$ individuals, where $p_c$ is referred to as the crossover rate. The uniform crossover exchanges two values at each position on two individuals among the two individuals with probability of 50%.

Finally, the mutation operator used in the proposed algorithm is that values at each position on the $N$ individuals obtained after the uniform crossover are changed to other values with probability $p_m$, which is referred to as the mutation rate.

D. Algorithm Flow

First, $N$ individuals, which are $N$ P2P network topologies, are randomly generated as an EA population. The generated individuals are actually utilized for object search for time $T$.
and then proceed to a fitness evaluation phase. The fitness value for each individual is the average number of hops over all searches using the individual in time $T$. Next, the evolutionary operators are applied to the present EA population based on their fitness values in order to generate a new EA population. The procedure from the real use of the individuals to the application of the evolutionary operators is repeated.

Finally, the algorithm parameters are summarized in Table I.

### TABLE I
PARAMETERS OF THE ALGORITHM FOR THE EVOLUTIONARY P2P NETWORKING.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$</td>
<td>the number of genes (nodes)</td>
</tr>
<tr>
<td>$N$</td>
<td>the number of individuals (P2P network topologies)</td>
</tr>
<tr>
<td>$T$</td>
<td>time period for using generated individuals</td>
</tr>
<tr>
<td>$H_{max}$</td>
<td>(a unit time: time required for all nodes to finish a search)</td>
</tr>
<tr>
<td>$B$</td>
<td>allowed number of hops for a query (TTL)</td>
</tr>
<tr>
<td>$p_c$</td>
<td>penalty for failed search</td>
</tr>
<tr>
<td>$p_m$</td>
<td>mutation rate</td>
</tr>
</tbody>
</table>

### IV. SIMULATIONS

#### A. Algorithm Configurations

The parameter values used here are shown in Table II. The mutation rate, $p_m$, is set to 0, as shown in Table II, so that the mutation operator is not used.

### TABLE II
ALGORITHM PARAMETER VALUES.

<table>
<thead>
<tr>
<th>$L$</th>
<th>$N$</th>
<th>$T$</th>
<th>$H_{max}$</th>
<th>$B$</th>
<th>$p_c$</th>
<th>$p_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>200</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>0.95</td>
<td>0</td>
</tr>
</tbody>
</table>

#### B. Evaluation Scenarios

In the simulations, search objects for nodes are nodes. Three evaluation scenarios are prepared as follows, which are different from each other in terms of how to decide nodes as search objects.

**Scenario 1**: Uniform

The nodes are selected as a search object with equal probability.

**Scenario 2**: Zipf’s law ($\alpha = 1$)

A node with the ID number of $k$ is selected as a search object with probability $P(k)$ in Equation (1), which represents Zipf’s law [6]. In this scenario, $\alpha$ in Equation (1) is 1. The selected nodes become more biased as $\alpha$ increases.

$$P(k) = \frac{k^{-\alpha}}{\sum_{n=1}^{L} n^{-\alpha}}$$

(1)

**Scenario 3**: Zipf’s law ($\alpha = 5$)

A node with the ID number of $k$ is selected as a search object with probability $P(k)$ in Equation (1) with $\alpha = 5$.

### C. Observation Items

1. **Convergence time**

Since the algorithm does not employ the mutation operator and the probability models to decide nodes as search objects are not varied in time, continuous implementation of the algorithm should cause all of the individuals to be identical. We refer to the state in which all of the individuals have become identical as **convergence** and refer to the time required to induce convergence as the **convergence time**.

2. **Final fitness value**

Since all of the individuals are identical at the convergence time, the fitness values for all of the individuals are also identical. This identical fitness value is observed.

3. **Final average number of hops**

This is the average number of hops over all successful searches, which mean that the nodes as search objects were found within $H_{max}$ hops, when using the identical individual at the convergence time.

4. **Final number of search failures**

The final number of search failures is the number of times that search fails, which means that the nodes as search objects were not found within $H_{max}$ hops among all of the searches using the identical individual at the convergence time.

#### D. Simulation Results

The results for the four observation items mentioned in the previous section are shown in Table III, in which the best fitness value at the first evaluation of the EA population (Initial Best) is also shown. These results are the average of 20 independent simulation runs. In addition, an example of the convergence process is shown in Figure 3. Figure 3(a) shows the convergence curves of the best fitness value (Best) and the average fitness value (Average). Figure 3(b) shows change in the average number of hops required for successful searches over time. Figure 3(c) shows changes in the numbers of failed searches over time when using the best individual (Best) and when using all of the individuals (All).

### TABLE III
SIMULATION RESULTS.

<table>
<thead>
<tr>
<th></th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convergence time</td>
<td>28,740.0</td>
<td>14,295.0</td>
<td>11,020.0</td>
</tr>
<tr>
<td>Final fitness value</td>
<td>19.854598</td>
<td>16.758176</td>
<td>4.385680</td>
</tr>
<tr>
<td>Initial Best</td>
<td>19.842754</td>
<td>19.675155</td>
<td>18.560295</td>
</tr>
<tr>
<td>Final average number of hops</td>
<td>5.478815</td>
<td>5.281215</td>
<td>5.478815</td>
</tr>
<tr>
<td>Final number of search failures</td>
<td>98,998.05</td>
<td>77,961.9</td>
<td>3.615.95</td>
</tr>
</tbody>
</table>

In Scenario 1, all of the nodes are selected as a search object with equal probability, and therefore, there exists no P2P network topology that provides high search performance for most nodes. However, since the algorithm used here does not employ the mutation operator that brings novel parameter
values into the EA population, it is thought that some individuals that obtained better fitness values may be stabilized in the EA population, which requires a long time. Table III shows that the algorithm requires a long time to induce the convergence and could not provide high search performance for most nodes.

Scenario 2 causes search objects to be biased more to particular nodes than Scenario 1. However, the probability with which the node with the ID number of 1, which is most frequently selected as a search object, does not occupy the major part of the entirety of the selection probability. Therefore, it should be difficult for the algorithm to evolve P2P network topologies that provide high search performance for most nodes at the convergence time. Table III shows that the algorithm could not provide high search performance for most nodes at the convergence time, although the performance of Scenario 2 is not as bad as that of Scenario 1.

Finally, among all of the scenarios, Scenario 3 causes the search objects the most biased to particular nodes. This bias is so strong that the node with the ID number of 1 occupies the major part of the entirety of the selection probability. In this situation, it should be possible to evolve particular P2P network topologies that yield high search performance for most nodes. Table III shows that the algorithm could eventually evolve a P2P network topology that provides high search performance for most nodes and also that the convergence time is the shortest among all of the scenarios.

V. CONCLUSION AND FUTURE RESEARCH

In the present study, we proposed the concept of the evolutionary P2P networking and realized a concrete algorithm based on the proposed concept. We performed a basic evaluation of the algorithm and showed that, for the case on which the average search performance for all of the nodes is set as a fitness function, the more common search objects are among all of the nodes, an improved P2P network of the topologies can evolve quickly.

The eventual goal of the present research is to implement an algorithm of the evolutionary P2P networking on a real P2P network in order to demonstrate the usefulness of the concept. As such, in the future, we will consider the following: (1) designing a variety of fitness functions, (2) applying evolutionary operators locally, (3) making a more realistic simulation model, and (4) including humans as fitness functions, as in the interactive evolutionary computation [7].

ACKNOWLEDGMENT

This work was funded partly by the National Institute of Information and Communications Technology, and partly by the Japan Society for the Promotion of Science through a Grant-in-Aid for Young Scientists (B)(19700068).

REFERENCES