CONSTRUCTION OF COMPLEX NETWORKS USING MEGA PROCESS GA AND GRID MP

YOSHIKO HANADA

Graduate School, Department of Knowledge Engineering
and Computer Sciences,Doshisha University, 610-0321 Kyoto, Japan
Email: hanada@mikilab.doshisha.ac.jp

TOMOYUKI HIROYASU AND MITSUNORI MIKI

Department of Knowledge Engineering
and Computer Sciences,Doshisha University,
610-0321 Kyoto, Japan
Email: {tomo@is, mmiki@mail}.doshisha.ac.jp

In this study, a new Genetic Algorithm for large-scale computer systems comprised of massive processors, we call Mega Process GA, is introduced. Our method has a GA-specific database possesses information of searched space. In addition, the local search for non-searched spaces is applied using individuals stored in the database. Applying this local search, the searched space can be expanded linearly in accordance with the increase in computing resources and the exhaustive search is guaranteed under infinite computations. This method was applied to the problem regarding the construction of complex networks for the base study of interactions among proteins. Through the experiments, we indicated the prospect of construction of complex networks by our method. We examine the performance of the proposed method on a distributed computing environment, which is built up by the commercially available middleware produced by United Devices Inc., named Grid MP.

1. INTRODUCTION

Genetic Algorithms (GAs) are among the most effective approximation algorithms for optimization problems. GAs are well suited to parallel processing environments due to their abilities to search with multiple points and their tolerances for extinction of search points. In consequence GAs have found applications in large-scale computing [1-4]. Due to the recent emergence of super PC clusters and Grid computation environments, such as PC Grid comprised of desktop machines for home use or offices, the number of available computational calculation resources is increasing; therefore, GA that uses large-scale computer systems comprised of massive processors, i.e., Mega Processors, has become feasible. However, the

application of GAs on large-scale computing environment, such as grid computation environments, has the drawback that these algorithms lack scalability in their performance improvement in accordance with increases in available computing resources. Therefore, huge computing resources cannot be used effectively. This is caused by overlapping searches.

In this study, a Genetic Algorithm for large-scale computing system that has mechanisms to use massive computation resources laconically and to search effectively, we call Mega Process GA, is introduced. Our method has a GA-specific database that possesses information regarding the space that has been searched already to avoid overlapping searches and make effective use of computing resources in consideration of scalability of search performance when GAs use large computer resources. In our previous work for Mega Process GA, we proposed the expression of the searched region using schemata and a local search mechanism as a compression method to store large regions searched by several individuals. We then showed that the searched space could be expanded as the number of computing resources increased, enhancing accuracy and reliability by applying the database with the local search mechanism to a GA, and it was clear that the proposed method also showed superior performance with limited computing costs [5]. Nevertheless, in this work, there was the drawback that computing costs increased exponentially in accordance with generations. We proposed a new database and local search for Mega Process GA based on our previous work and it was shown in preparative experiments that our method ensured an effective exhaustive search and had almost the same performance as a conventional GA in primitive functions and test functions of continuous optimization problems [6].

In this paper, the proposed GA was applied to the problem regarding the construction of complex network for the base study of interactions among proteins. We examined the performance of the proposed method on a distributed computing environment composed of machines belonging to Doshisha University and RIKEN Genomic Sciences Center, which was built up by the commercially available middleware produced by United Devices Inc., named Grid MP.[a]

2. CONCEPT OF MEGA PROCESS GA

In this study, we introduce the Mega Process GA. Our method has GA-specific database that carries information regarding the regions that have already been searched to avoid search overlapping. At the same time, the proposed GA performs the local search for the space that is not searched to expand the searched space. To obtain optima earlier, our method of searches uses mainly schemes of GA; any methods of operators such as a crossover and a mutation or a generation alternation

[a] United Devices : http://www.ud.com

model can be applied. To use idle computing resources of enormous computing environments effectively, a local search is applied. The outline of our method is shown in Fig. 1.

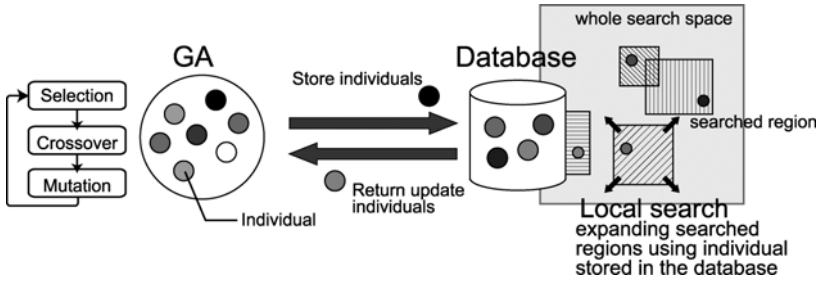

Fig. 1. The outline of our method

2.1. Database

When a database stores all the individual information, it takes a large time to check an individual that has already been searched due to the vast amounts of data. Therefore, all search regions should be stored as highly compressed expressions and checking of individuals stored in the database should not be time-consuming. We used binary-coded individuals, and an individual or set of individuals is represented by 2-dimensional coordinates using the mapping method proposed by Collins, which converts the multidimensional search space to a 2-dimensional plane[7]. In this mapping method, an individual is treated as the coordinates (x, y), where the integer x is coded by Gray-coding from the string composed of genes of 2k-th loci extracted from the chromosome, and integer y is coded from the string of extracted (2k+1)-th loci. Each individual has a one to one correspondence with a set of coordinates. Hence, the 2-dimensional plane can express the whole space. In our proposed database, the set of individuals that have contiguous coordinates on the 2-dimensional plane are represented by a rectangle and are stored by two diagonal vertexes (x_{min}, y_{min}), (x_{max}, y_{max}) of the rectangle. In addition, the best individual in the searched region is stored.

Fig.2 illustrates the whole search space of a problem consisting of 6 bits represented as a 2-dimensional plane. Examples for a set of individuals represented as rectangles are shown in this figure. Fig. 3 shows the database that possesses searched regions shown in Fig. 2. In this figure, a rectangle is represented with (x_{min}, y_{min})-(x_{max}, y_{max}), which are diagonal vertexes. This notation enables us to comprehend the quantitative rate of the searched region by calculating the square measure of the rectangle.

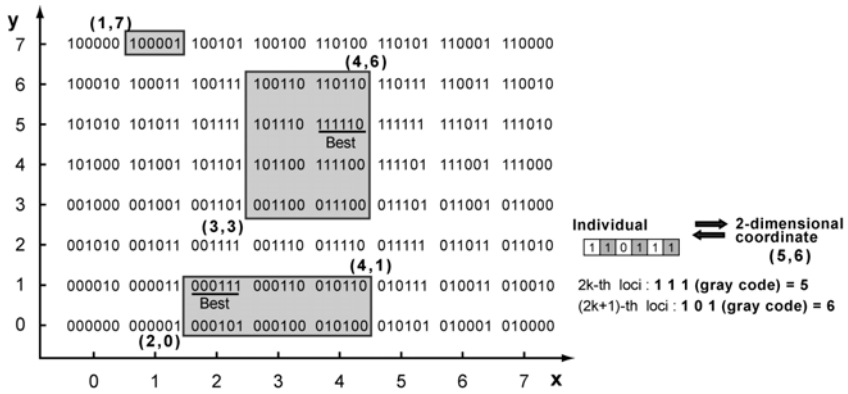

Fig. 2. An Example of a 6-bit Problem Space created by the Mapping

	Searched region	Best individual	Size of searched region
data1	(3,3)-(4,6)	111110	8 (4x2)
data2	(2,0)-(4,1)	000111	6 (2x3)
data3	(1,7)-(1,7)	100001	1 (1x1)

Fig. 3. Database representing searched regions illustrated in Fig.2

2.2. Local search

Our proposed database possesses searched individuals in the GA search using the expression described in the previous section. In addition, using individuals stored in the database, a local search for the space that has not been searched is applied to use idle computing resources of enormous computing environments effectively.

In our proposed local search, each rectangle stored in the database as searched regions is expanded vertically and horizontally on a 2-dimensional plane. In one step in our proposed local search, each rectangle is expanded vertically and horizontally by 1. Fig.4 illustrates our proposed local search. In this figure, the rectangular regions, or individuals, painted in gray are searched region. The regions surrounded with bold lines are searched additionally in one step of local search. As a result, searched regions are expanded. The directions of longitudinal expansion and lateral expansion are determined by fitness of the best individuals on the edges of the rectangle.

In this study, our method is applied to the problem regarding the construction of complex networks for the base study of interactions among proteins. We describe about complex networks and how to construct them using GAs in the next session.

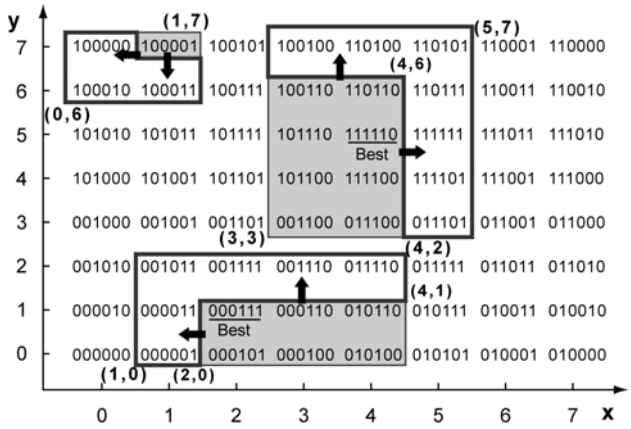

Fig. 4. An Example of our proposed local search

3. COMPLEX NETWORKS

In this paper, we applied our method to the problem regarding the construction of complex networks. Complex network is the network that consists of multiple elements in population, or system, and indicates an unique behavior that surpasses total of partial behavior as the whole due to influences and interaction among its elements [8]. Complex networks have been often found in interaction among proteins as shown in Fig.5; however, their structures and characteristics have not been clear [9]. Various studies to analyze existing networks have been discussed.

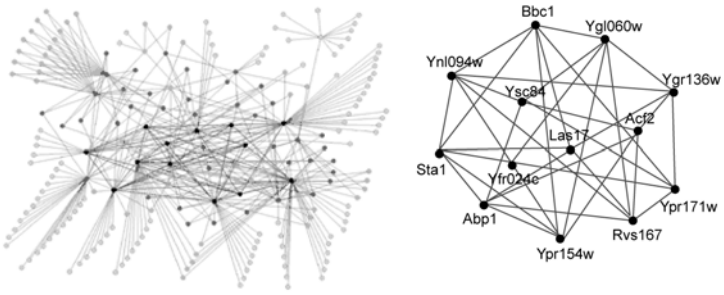

Fig. 5. An example for protein-protein interactions

We apply another approach that focuses on the characteristic in complex network and constructs the network using optimization algorithm. Characteristics of the constructed network are then examined in this approach.

3.1. Model of Problem

Our approach is to construct complex networks by GA that optimizes a representative value of the network. We used the average of curtate distance between nodes that is an objective function that should be minimized and we fixed the number of links and nodes as constrained conditions.

The problem used in here is defined as (1), where D is the average of curtate distance between nodes and D_{ij} is curtate distance between nodes i and j.

$$D = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} (D_{ij}) \quad (i \neq j) \tag{1}$$

In this paper, we examine whether networks obtained by GAs minimizing D can satisfy conditions of complex networks. Most existing networks are known in generally as scale-free networks [10]. Scale-free networks are composed of the node named Hub that has huge number of links and the node that are linked with few nodes[9]. These networks have larger cluster coefficients and smaller average distances between nodes than random networks [11]. Cluster coefficient is one of characteristics in network and defined as (2) where C is the cluster coefficient and C_i is the cluster coefficient at node i [12].

$$C \equiv \frac{1}{N} \sum_{i=1}^{N} C_i, \; C_i = \frac{2Ei}{k_i(k_i - 1)} \tag{2}$$

By comparing obtained networks by GA with random networks, we acknowledge that obtained networks by GAs have the characteristic of scale-free networks, i.e., complex networks, if they have larger cluster coefficient and smaller average distances than those of random networks.

3.2. Construction of Complex Networks by GA

3.2.1. *Coding Method*

Each topology of nodes has to be coded into chromosomes to optimize complex networks by GA. First, topologies are represented by matrix, each element a_{ij} of which indicates connection between node n_i and node n_j ($0 <= i, j < N$: N is the number of nodes). a_{ij} stands at '1' where node n_i and node n_j are connected by the link and stands at '1'. '0' indicates that two nodes are not connected. This matrix is symmetric; therefore, we use elements belonging in the upper triangular matrix.

Fig.6 shows the genotype of a network composed of 6 nodes and 5 links. In this coding method, the length of chromosome is $N(N\text{-}1)/2$ and the scale of the whole search space is $2^{N(N\text{-}1)/2}$.

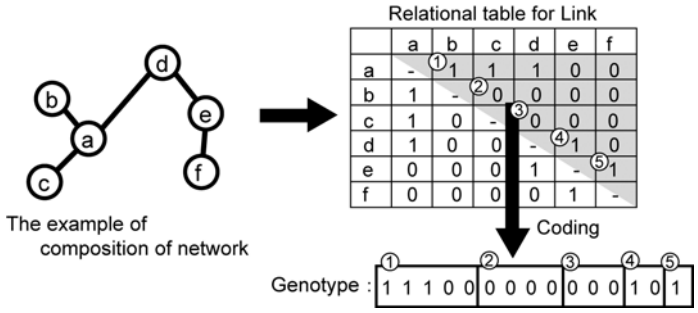

Fig. 6. An example for the coding for a network

3.2.2. *Crossover*

In this problem there are two constraints: The total number of links stays constant and each node has to be connected directly or remotely to other nodes. Networks shown in Fig. 7 are not feasible.

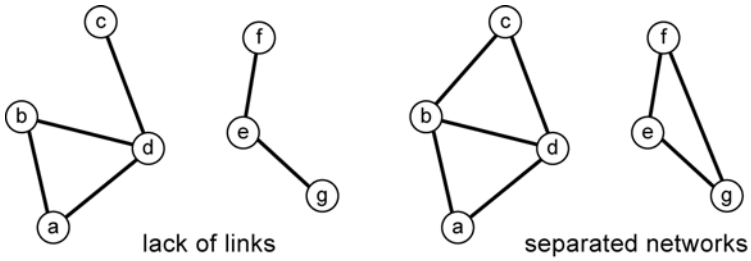

Fig. 7. Examples of non-feasible networks in 7nodes-8links problem

To avoid generating of non-feasible networks, we introduce the crossover that keeps the number of links. In our crossover, different links between two networks are exchanged in random. Fig. 8 shows an instance for this crossover.

Modification is then applied if the generated network does not satisfy the second constrained condition as shown in Fig.9. In this figure, one of loose-fitting links is cut and two separate networks are then bridged by the arbitrary link.

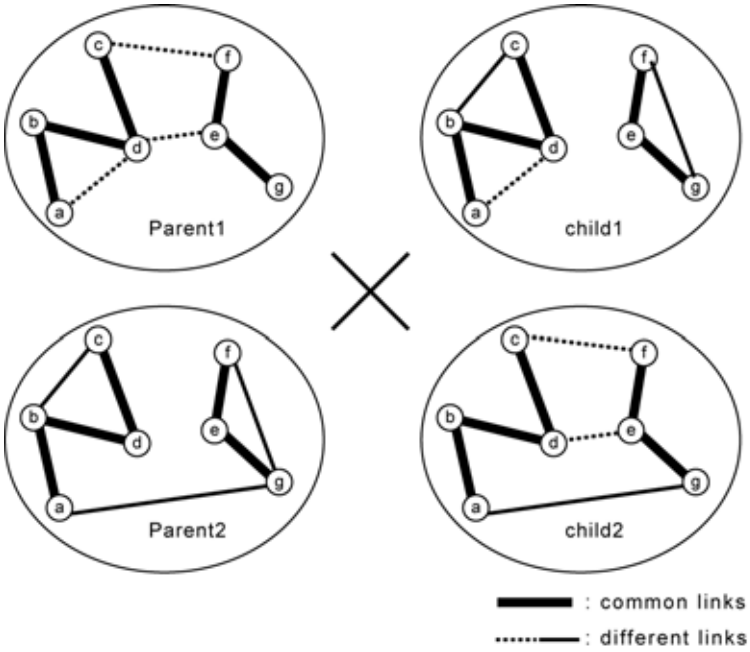

Fig. 8. Examples of Crossover

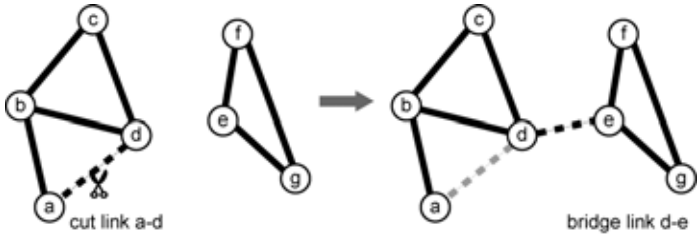

Fig. 9. Modification of a non-feasible network

These crossover methods were applied to construct networks. We then examined whether networks obtained by GA and our proposed GA minimizing the average of curtate distance between nodes can satisfy conditions of complex networks.

4. NUMERICAL EXPERIMENT

It was shown in preparative experiments that our method ensured an effective exhaustive search and had almost the same performance as a conventional GA in primitive functions and test functions of continuous optimization problems [6].

Here, its effectiveness in limited computation costs was examined in the network construction problems by comparing it to a conventional GA. We then examined whether networks obtained by GA and our proposed method can satisfy conditions of complex networks. In addition, we performed proposed method on a distributed computing environment built up by Grid MP and showed that our method could expand the searched space in accordance with the increase in computing resources.

4.1. Performance with Limited Computation Cost

In our method, there are various usages for the proposed database with GA. In this experiment that used one computation node, a local search was conducted alternately with a GA using individuals stored in the database. To obtain a good solution at as early a stage as possible, the search was advanced based on the GA. The proposed method is outlined in Fig.10.

First, the population in the GA is initialized. The number of individuals in the database at initial generation is 0. Next, genetic operations, such as crossover, mutation, and selection, are conducted in the population of GA, and the best individual of the population is preserved in the database. However, if this individual is already preserved in the searched region of the database, it will not be preserved again. Next, a local search operation is applied to all individuals in the database. When an individual with better fitness than the individual of interest is found, the copy of the individual will replace the worst individual in the GA population.

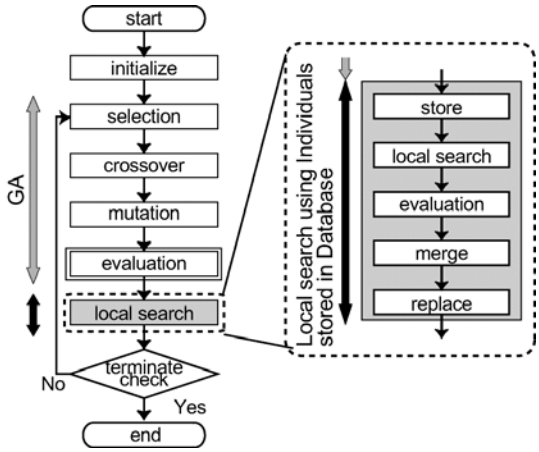

Fig. 10. The flow of the proposed GA performed in one computation node

The proposed GA and a conventional GA are compared in the limited number of evaluations. In this experiment, we took cases of the alignment of node of

benchmarks problem used in Traveling Salesman Problem; three instances, (29 nodes, 45 links), (51 nodes, 76 links) and (101 nodes, 150 links), were used. (*N* node, *M* links) denotes that the number of nodes is *N* and number of links is *M*.

The ER model [13] was used as the alternation in each generation. The population size was 100 and each couple, or parents, generated 20 children by crossover. The maximum numbers of evaluation calculations was limited to $4x10^4$ in (29 nodes, 45 links) and (51 nodes, 76 links). In (101 nodes, 150 links), $8x10^4$ evaluations were achieved.

Table 1 shows the averages of curtate distance between nodes of obtained network by GA and our method in three instances. Cluster coefficients are shown in Table 2 for comparing them with those of random networks.

Table 1. Averages of curtate distance between nodes

	Conventional GA		Proposed Method		Random Network
	best	average	best	average	best
29nodes 45links	1016.80	1019.81	1015.18	1019.27	1796.93
51 node 76link	38.96	39.02	38.97	39.02	81.30
101 nodes 150 links	38.65	38.75	38.61	38.78	98.58

Table 2. Averages of Cluster coefficients

	Conventional GA	Proposed Method	Random Network
29 nodes 45 links	0.235	0.229	0.100
51 node 76link	0.125	0.128	0.053
101 nodes 150 links	0.092	0.097	0.035

These results are the best and average fitness obtained by two methods are from 20 trials and the result of random network is from 100 trials.

Result shows that our method retains the performance of a conventional GA in limited computational cost though a large amount of evaluation calculation is necessary in the local search. In addition, by comparing an obtained network with random networks, it was clear that obtained network had larger cluster coefficient and smaller average distances than those of random network. Scale-free network is the network that has larger cluster coefficients and smaller average distances between nodes than random networks; therefore we conclude that both methods are effective for constructing scale-free networks, i.e., complex networks.

4.2. Implementation on Grid MP

To discuss effectiveness of our approach on a large-scale computing system, we constructed a distributed computing environment using Grid MP produced by United Devices Inc. as a grid middleware. Parallelization is applicable to this local search by allotting each rectangle, i.e., a searched region stored in the database, to computation nodes. There are no strong dependencies and little communication among searches; therefore, to execute the local search on distributed computing systems is expected to yield high throughput computation.

In the previous section, we have described an implementation of our approach that conducts the local search alternately with a GA in limited computation costs. This implementation is not appropriate on distributed computing environment because of large overhead in synchronization between GAs and local searches. In distributed environment built up with Grid MP, a GA and local search are conducted in asynchronous way shown in Fig. 11. The best individual found in GA is stored into the database every generation. The database is kept by the server named MP Server, and the local search that expands the searched region using information stored in the database is executed respectively in worker nodes named Devices, of the system. Each node returns periodically its current searched region to server to update the database. GA refers to the database for information about searched regions during evaluations.

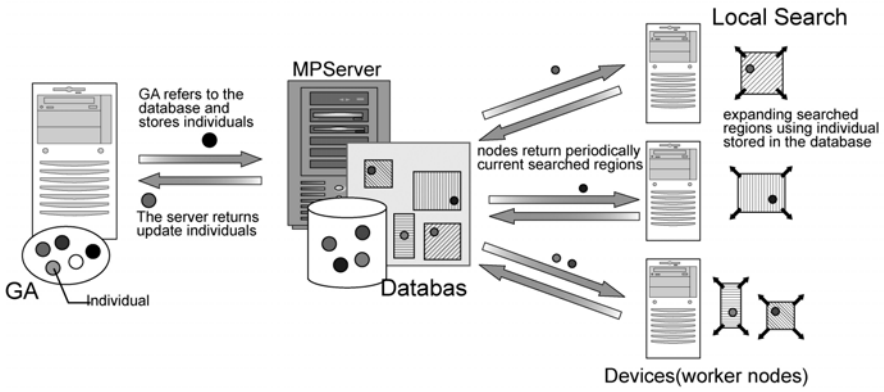

Fig. 11. Implementation on Grid MP

We examined whether it assured the scalability of search performances, i.e., the scale of searched region, against the number of computing resources. We used Grid MP platform version 4.1. The specification of machines used for the experiments is shown in Table 3. The term “Local Machine” in Table 3 indicates the machine that performs GA and submits jobs regarding local search to the MP Server.

Table 3. Specification of machines used for the experiment

	Affiliation	#Nodes	Processors	Memory
Local Machine	- -	1	Pentium M 1.8GHz	1GB
MP Server	Doshisha	1	Xeon 2.8GHz x 2	2GB
Devices	Doshisha	9	Xeon 2.4GHz x 2	1GB
	RIKEN GSC	21	Celeron 1.3GHz	896MB

We used 10, 20 and 30 Devices shown in Table 3. Sets of Devices used in this experiment were (3,7), (6,14), and (9,21) where the notation (n_1, n_2) indicates that n_1 nodes of Doshisha and n_2 nodes of RIKEN GSC are used. The parameters of GA were the same as in the previous section except that population size was 150, and the 51 nodes 76 edges problem were examined.

In the local search run on Grid MP, each Device returned the current searched region every 3 minutes. Fig. 12 shows the scale of searched region obtained by 2 hours of execution using 10, 20, and 30 Devices shown in Table 3. From this result, it is obvious that the searched space can be expanded linearly in accordance with the increase in computing resources as we proposed. In the optimization process, the scalability of the searched region against the number of calculation resources is very important. The environment used in this study is a small environment and we should examine the performance of the proposed method on large scale distributed computing environments. This examination will be conducted in our future studies.

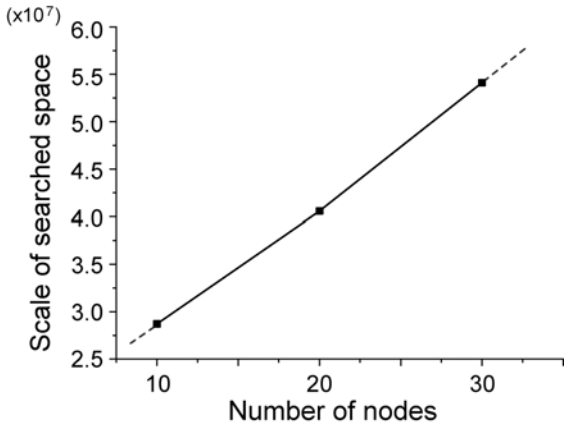

Fig. 12. Increase in scale of searched region against the number of nodes

5. CONCLUSION

GAs are well suited to parallel processing environments in a certain amount of calculation nodes. Due to the recent emergence of super PC clusters and Grid computation environments the number of available computational calculation resources is increasing. Therefore, GA that uses large-scale computer systems comprised of massive processors, i.e., Mega Processors, has become feasible. Mechanisms to use massive computation resources laconically and to assure the scalability of search performances against the number of computing resources are necessary if large-scale computer systems are available.

In this paper, a Genetic Algorithm for large-scale computing system that had mechanisms to use massive computation resources and search effectively, we called Mega Process GA, was introduced. Our method has a GA-specific database that possesses information regarding the space that has been searched already to avoid overlapping searches and make effective use of computing resources in consideration of scalability of search performance when GAs use large computer resources. In addition, the local search for non-searched spaces is applied using individuals stored in the database. Applying this local search, the searched space can be expanded linearly in accordance with the increase in computing resources.

Our proposed method was applied to the problem regarding the construction of complex network for the base study of interactions among proteins. In limited computational cost, it was clear that our method retained superior performance of a conventional GA though a large amount of evaluation calculation was necessary in the local search. We then performed proposed method on a distributed computing environment built up by Grid MP and showed that our method could expand the searched space in accordance with the increase in computing resources. The environment used in this study is a small environment. In network construction problem composed of huge nodes and links, enormous amount of computation are required due to its wide spread search space. In consequence, we should examine the performance of the proposed method on large scale distributed computing environments. This examination will be conducted in our future studies.

Acknowledgments

We are grateful to Prof. Dr. Akihiko Konagaya and Fumikazu Konishi of RIKEN Genomic Sciences Center and Hiroyuki Kobayashi of Sumisho Electronics Co., Ltd. for valuable discussion and contributions to the development of the distributed computing environment built using Grid MP.

References

1. Y.Tanimura et al.: Development of Master-Worker System for The Computational Grid. Information Processing Society of Japan: Computing System. vol45, No.SIG6(ACS6), pp.197-207, May 2004. in Japanese.
2. H. Imade et al.: A Grid-Oriented Genetic Algorithm for Estimating Genetic Networks by S-Systems, Proc. SICE Annual Conf. Pp3317-3322, 2003.
3. H. Imade et al.: A framework of grid-oriented genetic algorithms for large-scale optimization in bioinformatics.
4. H. Nakata et al.: Protain structure optimizaion using Genetic Algorithm on Jojo Journal of Information Processing Society of Japan. 2002-HPC-93, pp. 155-160, 2003. in Japanese.
5. Y. Hanada et al.: Mega Process Genetic Algorithm Using Grid MP, Life Science Grid 2004, LNAI, vol.3370, pp. 152-170, 2004
6. Y. Hanada et al.: An Improvement of Database with Local Search Mechanisms for Genetic Algorithms in Large-Scale Computing Environments, 2005 IEEE Congress on Evolutionary Computation.(to appear)
7. T. Collins:Understanding Evolutionary Computing: A Hands on Approach, KMI-TR-48. September 1997.
8. G. Nicolis et al.: Exploring complexity. R. Piper GmbH and Co. KG Verlag, 1989.
9. A.-L. Barabasi. Linked: the new science of network perseus books. 2002.
10. A.-L. Barabasi, R. Albert. Emergence of scaling in random networks. Science 286, pp. 509-512, 1999.
11. P. Erdoas et al.:. Publ.math.inst.hungary academy sci 5,17.1960.
12. W. Souma et al.: The role of small world network. Technical report of IEICE, NGN2001-12. in Japanese.
13. D. Thierens et al.: Elitist Recombination: an integrated selection recombination GA, Proceedings of the 1st IEEE Conference on Evolutionary Computation, pp.508-512. 1994.