GENETIC IMAGE NETWORK (GIN): AUTOMATICALLY CONSTRUCTION OF IMAGE PROCESSING ALGORITHM

Shinichi Shirakawa and Tomoharu Nagao

Department of Information Media and Environment
Graduate School of Environment and Information Sciences
Yokohama National University
79-7, Tokiwadai, Hodogaya-ku, Yokohama, Kanagawa, 240-8501, Japan
E-mail: shirakawa@nlab.sogo1.ynu.ac.jp

ABSTRACT

A new method for automatic construction of image transformation, Genetic Image Network (GIN), is proposed in this paper. We previously proposed the system of ACTIT (Automatic Construction of Tree-structural Image Transformation). ACTIT constructs tree structured image processing filters using Genetic Programming (GP). Generally, network structure theoretically includes tree structure (i.e. network structure also represent tree structure.). Thus, the description ability of network representation is higher than that of tree structure. In this way, we construct complex image transformations which cannot be constructed by tree structure. We applied GIN to automatically constructing image transformation and compare GIN with ACTIT and show effectiveness of GIN.

1. INTRODUCTION

This paper introduces a new method for automatic construction of image transformation. This new method, named Genetic Image Network (GIN), uses network representation.

In image processing, it is difficult to select image processing filters to satisfy the transformation from original images to its target images. We previously proposed the system of ACTIT (Automatic Construction of Tree-structural Image Transformation) [1–4]. ACTIT approximates adequate image transformation from original image to their target image by a combination of several known image filters. ACTIT constructs tree structural image processing filters using Genetic Programming (GP) [5, 6]. It is difficult to represent repetition, loop and recursive structure using tree structure.

Whereas, GIN evolves the network structure of image processing filters based on instance based learning in similar ways of the case of ACTIT. The biggest difference between GIN and ACTIT is the structure of connections of image filters. Generally, network structure theoretically includes tree structure (i.e. network structure also represent tree structure.). Thus, the description ability of network representation is higher than that of tree structure. We expect GIN to automatically construct a simple structure for image transformation using its network representation.

In order to verify the effectiveness of GIN, we applied GIN to automatically construction of image transformation. From several experimental results, it is verified that GIN automatically constructs image transformation. Additionally, obtained structure by GIN is compact and unique, and uses loop and feedback structure that cannot be constructed by ACTIT.

This paper consists of five sections. Section 2 is an overview of several related works. Section 3 describes Genetic Image Network (GIN). Several experiments are shown in Section 4. Section 5 is devoted to the conclusions and the discussion of future works.

2. RELATED WORKS

2.1. Genetic Programming and Graph based Genetic Programming

Genetic Programming (GP) [5, 6] is one of Evolutionary Computation techniques, which was introduced by Koza. GP evolves computer programs, which are usually tree structure, and searches a desired program using Genetic Algorithm (GA). Today, a lot of extensions and improvements of GP are proposed.

Parallel Algorithm Discovery and Orchestration (PADO) [7, 8]is one of the graph based GPs instead of the tree structure. PADO uses stack memory and index memory. The execution of PADO is carried out from the start node to the end node in the network. PADO was applied to the object recognition problems.

Another graph based GP is the Parallel Distributed Genetic Programming (PDGP) [9]. In this approach the tree is represented as a graph with functions and terminals nodes located over a grid. In this way it is possible straightforward to execute several nodes concurrently.

Recently, Genetic Network Programming (GNP) [10–12] which has a directed graph structure is proposed. GNP is applied to make the behavior sequences of agents and shows better performances compared to GP.

2.2. Automatic Construction of Tree structural Image Transformation (ACTIT)

ACTIT [1–4] constructs tree structured image processing filters with one-input one-output filters and two-inputs one-output filters by using Genetic Programming (GP) to satisfy the given several image examples. The leaf nodes of a tree are the original images and other nodes are the image processing filters. We give “Training Image Set”, and ACTIT system constructs appropriate image processing automatically. We have solved many kinds of image processing problem by using ACTIT. 3D-ACTIT [3, 4] is extended method which automatically constructs various 3D image processing procedures, and applies to medical image processing.

3. GENETIC IMAGE NETWORK (GIN)

3.1. Structure of GIN

Genetic Image Network (GIN) is a novel method for automatically construction of image transformation. The representation of GIN is network structure, and GIN is composed of several nodes which are well-known image filters whose inputting is one or two.

Figure 1 illustrates an example of Phenotype (network structure) and Genotype (strings representing Phenotype) in GIN. Genotype in GIN consists of strings which indicate image filter type and connections.

We expect GIN to automatically construct a simple structure for image transformation using its network representation.

Initially, we set original images to “in” nodes when GIN executes. All nodes synchronously transform each inputted images, and output the transformed image to destination nodes. After predefined iterations (we call this parameter “the number of steps”), the images of output nodes are evaluated.

The nodes which have no inputted images do not operation. The two-input node which has only one inputted image outputs that same one.

3.2. Genetic Operators

To obtain the best structure of GIN, an evolutionary method is adopted. As usual, GIN uses *crossover* and *mutation* as the genetic operators.

The *crossover* operator affects two individuals, and the uniform crossover is used, as follows;

1. Select several nodes randomly according to the crossover rate P_c for each node.

2. The selected nodes with the same node number are swapped between two parents.

Figure 2 shows an example of crossover. In Figure 2, node 1, node 3 and node 4 are selected. The selected nodes are swapped, and generate offspring.

The *mutation* operator affects one individual.

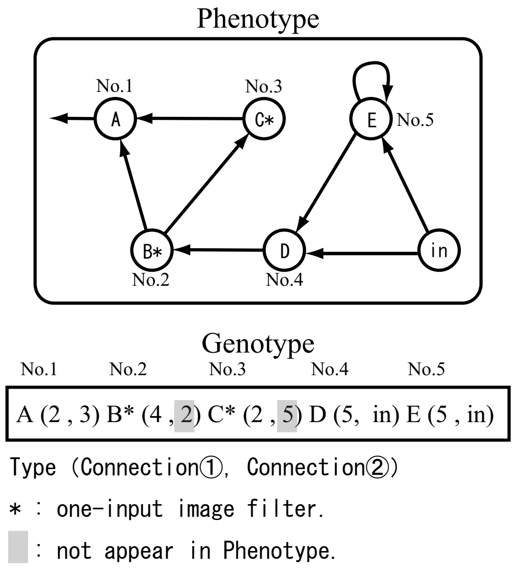


Fig. 1. An example of Genetic Image Network.

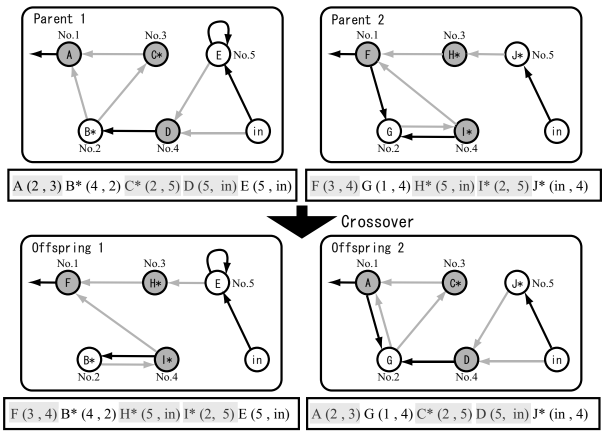


Fig. 2. Crossover operation of two individuals.

1. Select several conections and type of nodes randomly according to the mutation rate P_m for each conection and type.

2. The selected connections and type of nodes are randomly canged.

Figure 3 shows an example of mutation. The selected strings are randomly changed.

4. EXPERIMENTS

In this section several experiments with GIN and ACTIT are performed.

4.1. Setting of Experiments

“Training Image Set” we used in these experiments appears in Figure 4. The goal of these experiments is to obtain image transformation of extraction processing of “key” and “coin”.

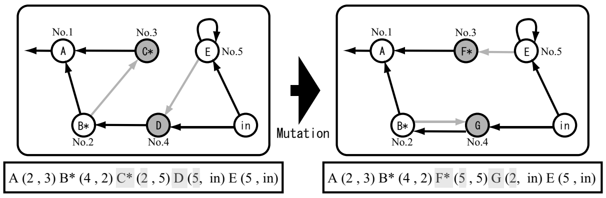
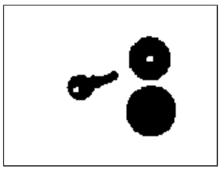


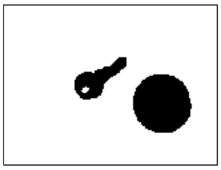
Fig. 3. Mutation operation of one individual.

(a) Original image 1.



(b) Target image 1.

(c) Original image 2.



(d) Target image 2.

Fig. 4. “Training Image Set” used in experiments.

We use the mean error on the “Training Image Set” as a fitness function. The fitness function used in these experiments is :

$$fitness = \frac{1}{N}\sum_{n=1}^{N}\left\{1 - \frac{\sum_{i=1}^{W}\sum_{j=1}^{H}|o_{ij}^{n} - t_{ij}^{n}|}{W \cdot H \cdot V_{max}}\right\} \quad (1)$$

where o_n is the transformed image and t_n is its target one. The numbers of pixel in the direction i and j are W, H respectively, and the number of training image sets is N. The range of this fitness function is [0.0, 1.0]. The higher the numerical value indicates the better performance.

The common parameters between the two methods (GIN and ACTIT) are identical. The parameters of GIN and ACTIT are given in Table 1 and Table 2, respectively.

We prepare “nop” node which means no operation for GIN. Therefore, it is possible GIN to represent less processing using more the number of steps.

4.2. Experimental Results

Results are given for 20 different runs with the same parameter set. In this experiment, the parameters of the number of steps in GIN are 5, 10, 15 and 20. Figure 5 is the transition of fitness of ACTIT and GIN. According to the result, GIN obtains a good solution as well as ACTIT.

Table 1. Parameters of GIN algorithm.

Parameter	Value
The number of generations	500
Population size	101
Crossover rate P_c	0.1
Mutation rate P_m	0.02
Selection	Tournament selection
Tournament size	2
The number of nodes	30
The number of steps	5, 10, 15, 20

Table 2. Parameters of ACTIT algorithm.

Parameter	Value
The number of generations	500
Population size	101
Crossover rate	1.0
Mutation rate	0.9
Selection	Tournament selection
Tournament size	2
Maximum number of functions	30

It is poor performance when we use “5” with the number of steps. Because it is difficult to obtain complex structure using low the number of steps.

The output images of “Training Image Set” using GIN and ACTIT are shown in Figure 6 and 7.

The images of both methods are similar to target images. The both methods automatically construct network structured or tree structured image transformation. We consider that the performance of GIN is identical ACTIT in this experiment.

4.3. Experiment to Test Images

In this section we apply the constructed network or tree structured image filters by GIN and ACTIT to test images which are not used in evolutionary process. The test image used in this experiment are shown in Figure 8.

Figure 9 and 10 are the output images using constructed network or tree structured image filters by GIN and ACTIT, respectively. GIN and ACTIT transform test images to ideal images which extracted “key” and “coin”. It shows that GIN and ACTIT automatically construct general image transformation through learning.

4.4. Obtained Structure

Figure 11 is obtained structure by GIN. The parameter of the number of steps is 15. It includes loop and feedback structure and reuse processed image. These structure cannot be construct by ACTIT.

Each node of GIN synchronously transforms each inputted images, and output the transformed image to destination nodes. In Figure 11, the output image of each node is changing every steps. Finally, input image is transformed

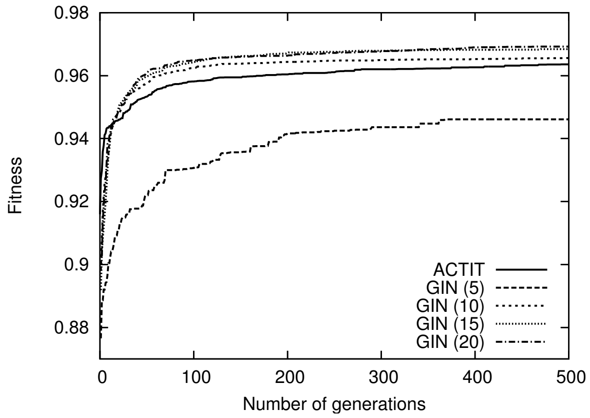
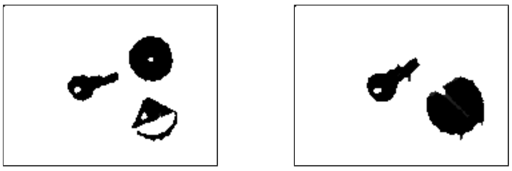
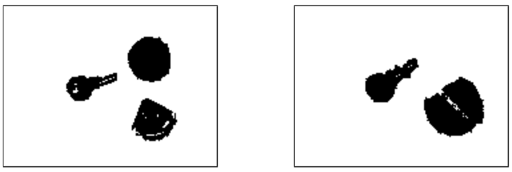


Fig. 5. The transition of fitness of ACTIT and GIN using "Training Image Set" in Figure 4. The number of steps in GIN uses 5, 10, 15 and 20. Each curve is an average over 20 runs.



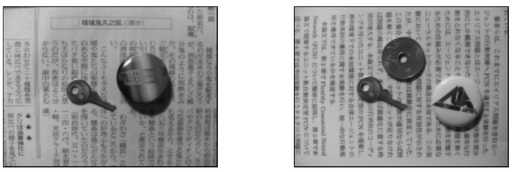
(a) Output image 1. (b) Output image 2.

Fig. 6. Output images of "Training Image Set" using GIN(15).



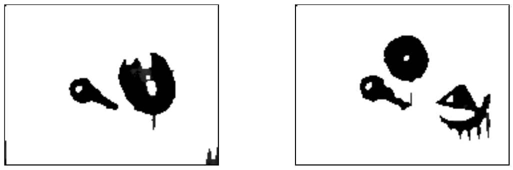
(a) Output image 1. (b) Output image 2.

Fig. 7. Output images of "Training Image Set" using ACTIT.



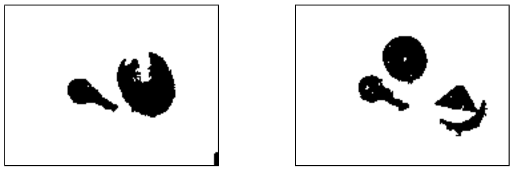
(a) Original image 1. (b) Original image 2.

Fig. 8. Test images which are not used in evolutionary process.



(a) Output image 1. (b) Output image 2.

Fig. 9. Output images of GIN using test images in Fig.8.



(a) Output image 1. (b) Output image 2.

Fig. 10. Output image of ACTIT using test images in Fig.8.

to ideal output image. Although the obtained network structure by GIN is compact, actual image processing is very complex.

4.5. Extension to Plural Outputs using GIN

GIN constructs network structured image filters, thus it is possible to represent plural outputs. GIN enables to simultaneously construct plural image transformation using only a single network structure. It is also possible GIN to reuse the processed image in its network structure. Whereas, ACTIT must be one output, because it uses tree structure for representation of image transformation.

In this section we experiment to construct plural image transformation using GIN. We use the same "Original Images" in Figure 4. There are two "Target Images" in each "Training Image Set" which appear in Figure 12. The goals of this experiment are to extract only "key" and only "coin" at the same time simultaneously.

In this experiment, we use the same parameter in Table 1, and the parameter of the number of steps is 15. GIN has to simultaneously construct different image transformation in its network structured filters.

The results of "Training Image Set" are shown in Figure 13. GIN obtained the two image transformations using its network representation.

Next, we apply the constructed network structured image filters to "Test Images" in Figure 8. The results are in Figure 14. The processing of extracting only "coin" was success, extracting "key", however, failed.

If we use more "Training Image Set" or more variety of image processing filters, GIN enables to represent more complex image processing. Therefore, it will be possible to construct successful and general network structured image transformation which has plural outputs using GIN.

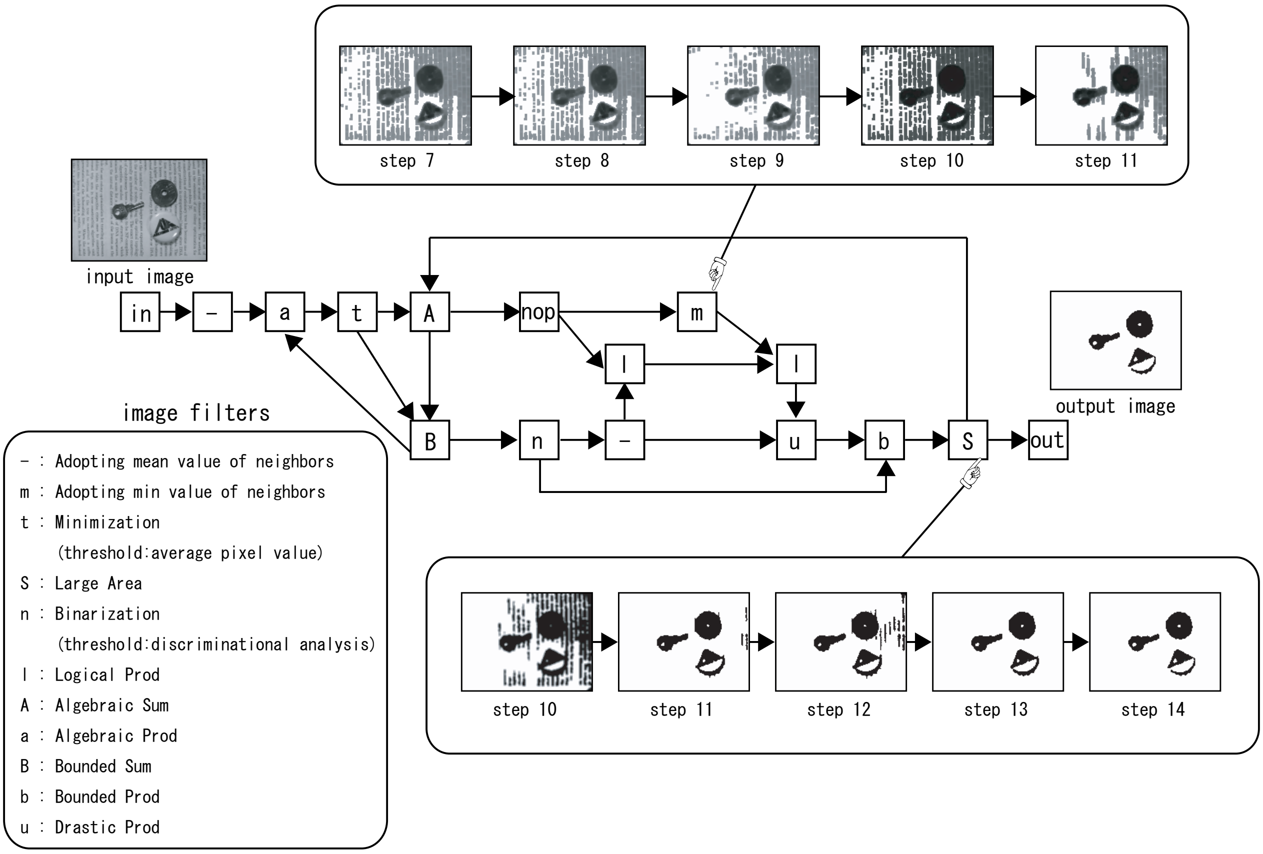
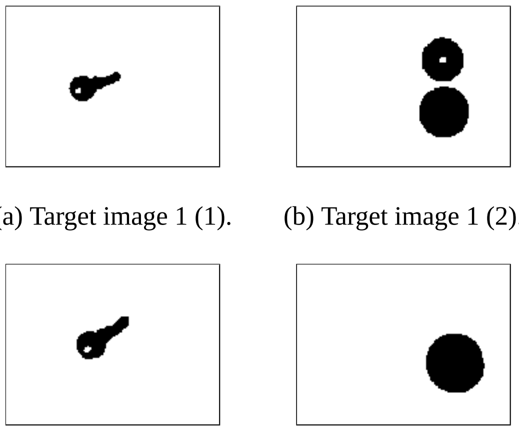


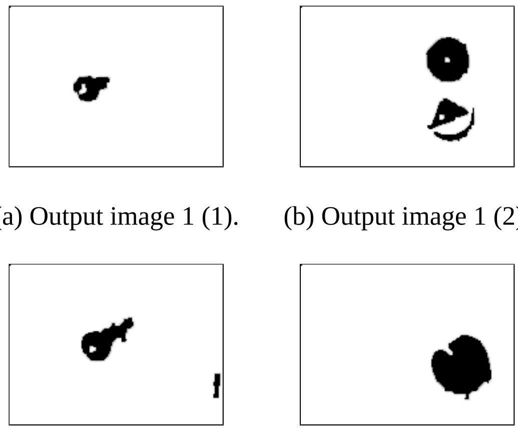
Fig. 11. An example of constructed structure by GIN.



(a) Target image 1 (1). (b) Target image 1 (2).

(c) Target image 2 (1). (d) Target image 2 (2).

Fig. 12. "Target Images" which are extracted only "key" and only "coin". (1) is extracted only "key". (2) is extracted only "coin".



(a) Output image 1 (1). (b) Output image 1 (2).

(c) Output image 2 (1). (d) Output image 2 (2).

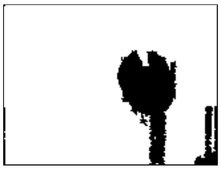
Fig. 13. Output images of "Training Image Set" using GIN(15). (1) is extracted only "key". (2) is extracted only "coin".

5. CONCLUSIONS

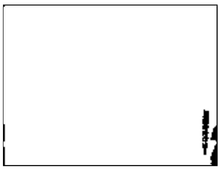
A novel automatic constructing of image transformation which is Genetic Image Network (GIN) has introduced in this paper. We tested the performance of GIN against ACTIT. In all test cases GIN shows good performance as well as ACTIT. Obtained structure of GIN is unique and compact which include loop and feedback.

We extend GIN to plural outputs which cannot construct by ACTIT. GIN construct ideal network structured image

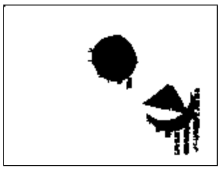
(a) Output image 1 (1).



(b) Output image 1 (2).



(c) Output image 2 (1).



(d) Output image 2 (2).

Fig. 14. Output images of “Test Image” using GIN(15). (1) is extracted only “key”. (2) is extracted only “coin”.

filters and shows effectiveness.

In order to clarify the effectiveness of GIN, more experiments have to be run on a variety of test problems. In future works we plan to extend tests to other problems.

The parameters of executing iteration number and the number of nodes have to be predefined in this paper. We also plan to develop automatic determination of these parameters.

6. REFERENCES

[1] Shinya Aoki and Tomoharu Nagao. ACTIT; automatic construction of tree-structural image transformation. *The Institute of Image Information and Television Engineers*, 53(6):888–894, 1999.

[2] Shinya Aoki and Tomoharu Nagao. Automatic construction of tree-structural image transformation using genetic programming. In *Proceedings of the 1999 International Conference on Image Processing (ICIP-99)*, volume 1, pages 529–533, Kobe, October 24–28 1999. IEEE.

[3] Yuta Nakano and Tomoharu Nagao. 3D medical image processing using 3D-ACTIT; automatic construction of tree-structural image transformation. In *Pcoc of the International Workshop on Advanced Image Technology IWAIT-04*, pages 529–533, Singapore, January12–13 2004.

[4] Yuta Nakano and Tomoharu Nagao. Automatic extraction of internal organs region from 3D PET image data using 3D-ACTIT. In *Pcoc of the International Workshop on Advanced Image Technology IWAIT-2006*, Okinawa, Japan, January10 2006.

[5] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[6] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge, MA, USA, 1994.

[7] Astro Teller and Manuela Veloso. PADO: A new learning architecture for object recognition. In Katsushi Ikeuchi and Manuela Veloso, editors, *Symbolic Visual Learning*, pages 81–116. Oxford University Press, 1996.

[8] Astro Teller and Manuela Veloso. Program evolution for data mining. *The International Journal of Expert Systems*, 8(3):216–236, 1995.

[9] Riccardo Poli. Evolution of graph-like programs with parallel distributed genetic programming. In Thomas Back, editor, *Genetic Algorithms: Proceedings of the Seventh International Conference*, pages 346–353, Michigan State University, East Lansing, MI, USA, 19-23 July 1997. Morgan Kaufmann.

[10] Kotaro Hirasawa, M. Okubo, J. Hu, and J. Murata. Comparison between genetic network programming (GNP) and genetic programming (GP). In *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pages 1276–1282, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001. IEEE Press.

[11] Hironobu Katagiri, Kotaro Hirasawa, Jinglu Hu, and Junichi Murata. Network structure oriented evolutionary model-genetic network programming-and its comparison with genetic programming. In Erik D. Goodman, editor, *2001 Genetic and Evolutionary Computation Conference Late Breaking Papers*, pages 219–226, San Francisco, California, USA, 9-11 July 2001.

[12] Toru Eguchi, Kotaro Hirasawa, Jinglu Hu, and Noriko Ota. A study of evolutionary multiagent models based on symbiosis. *IEEE Transactions on Systems, Man and Cybernetics Part B*, 36(1):179–193, 2006.