Proposal of a System of Function-Discovery Using a Bug Type of Artificial Life

Seiichi Serikawa, Member and Teruo Shimomura, Member (Kyushu Institute of Technology)

The effectiveness of artificial life is investigated from an engineering point of view. A system (named S-system) of function-discovery using a bug type of artificial life is proposed in this study. Some functions are extracted by the system. The chromosome of a bug consists of functions, constants and variables. A tree structure is used for the expression of the chromosome. Some observation data are provided for the bugs. After obtaining the data, they reproduce. The concept of sexual / asexual reproduction is introduced in this study. The number of homogeneous bugs is limited for a variety of species. These ideas are very effective for a function-search. A part of the chromosome changes by mutation. As the generation proceeds, the bugs with the function in agreement with the observation data survive selectively, and finally determine the true function. For the validity of this system, some data which obey the known laws have been given for the bugs. The bugs have evolved and discovered some functions in agreement with the laws. As for an unknown function, observation data on glossiness have been provided. They have also discovered the function. In addition, they have determined the multiple curves included in the image data. The S-system has the characteristics that the solution tends to converge and stabilizes in comparison with Genetic Programming. Moreover, the form of the function is relatively simple.

Key words: Artificial life, genetic programming, genetic algorithm, evolution

1. Introduction

Although the study of artificial life (A-Life) has recently begun, its purpose and method have not yet been established. There are some studies concerning A-Life: the establishment of the evolution model, growth of morphology, evolutionary learning, parallel processing and self-restoration. These are classified into two types. One is the science oriented study, and the other is the engineering study. The science oriented study is used to establish a computational theorem of life, while the engineering study corresponds with the establishment of the system with the ability of evolution and adaptability. From the engineering point of view, A-life is regarded as the extension of the Genetic algorithm (GA). GA is an effective method for the search of optimum values, and its fundamental principle consists of the selection and the combination based on fitness. Recently, Genetic programming (GP) has been proposed for applications except for optimum search. Some unknown functions are discovered by GP. The discovered functions, however, are often highly complicated and are quite different from those discovered by human beings. In addition, the search does not stabilize. This is because a crossover is blindly applied. Accordingly, the form of function changes dramatically.

In this study, a bug type of artificial life with the ability of function-discovery is proposed from an engineering point of view. The chromosome of a bug consists of functions, constants and variables. The bugs move and catch the observation data. As the generation proceeds, the bugs with the functions in agreement with the observation data survive selectively, and determine the function corresponding with the data. We introduce the concept of sexual / asexual reproduction and limit the number of homogeneous bugs. These are important for function-search. For the sake of convention, we call this system S-system: (i.e. system using sexual / asexual reproduction). By the use of this approach, the solution tends to converge and stabilizes. In addition, the discovered function is relatively simple compared to GP. In reality, various functions were discovered from the observation data. The multiple different curves were also extracted from the image data.
2. Model of a bug type of A-life

2.1 Structural expression of the bug's chromosome

In this system, there are many individuals who have the chromosome shown in Fig. 1 in their body. For the sake of convention, we name each individual “bug”. The chromosome of a bug consists of a set of function symbols; functions, constants and variables. For the expression of functions as chromosomes, a tree structure is adopted in the same way as GP[5]. The tree structure is identified by means of the S expression of the LISP language. For example, the function $K_1 \cdot \sin(x) + y/(K_2 \cdot x)$ is expressed in Figs. 2 (a) and (b) by the tree structure and S expression, respectively. Figure 3 represents its genotype, which agrees with the chromosome. Thus, various functions can be represented as chromosomes.

2.2 Meaning of function-discovery

All experimental data $D$ are expressed as follows.

$$D = \{d_1, d_2, \ldots, d_N\} = \begin{bmatrix}
In_{i_1}
& In_{i_2}
& \cdots
& In_{i_N}
In_{i_2}
& In_{i_2}
& \cdots
& In_{i_N}
\vdots
& \vdots
& \ddots
& \vdots
In_{i_1}
& In_{i_2}
& \cdots
& In_{i_N}
Out_{i_1}
& Out_{i_2}
& \cdots
& Out_{i_N}
\end{bmatrix}, \quad (1)$$

where $N$ is the number of experiments. $d_i$ denotes the $i$-th observation data and it is given by Eq. (2).

$$d_i = \{In_{i_1}, In_{i_2}, \ldots, In_{i_N}, Out_i\}, \quad (2)$$

where $In_{i_1}, In_{i_2}, \ldots, In_{i_N}$ indicate the input data and $Out_i$ output data. In the case of a high correlation between the input data and output data, the following equation will hold.

$$Out_i = f(In_{i_1}, In_{i_2}, \ldots, In_{i_N}), \quad (3)$$

where symbol $f$ represents a certain function. It is our aim to discover an unknown function $f$ using a bug type of A-life. When the chromosome of a bug $p$ agrees with the function $f$, we assume that the bug $p$ has discovered the function $f$.

2.3 Algorithm of function-discovery

Before mentioning the algorithm of function-discovery, we define some equations. The fitness $fit_p$ of bug $p$ is defined as

$$fit_p = \frac{1}{1 + \sigma_p}, \quad (4)$$

where

$$\sigma_p = \frac{1}{N} \sum_{i=1}^{N} (Out_i - Bug_p(i))^2, \quad (5)$$

where $Bug_p(i)$ denotes the value which is obtained by the substitution of the $i$-th observation data $In_{i_1}, In_{i_2}, \ldots, In_{i_N}$ into the chromosome of bug $p$. The square of Eq. (5) represents the mean square error between $Out_i$ and $Bug_p(i)$. When $Bug_p(i)$ agrees with $Out_i$, then $\sigma_p = 0$ and $fit_p = 1$. In the case where the values of $Bug_p(i)$ and $Out_i$ differ substantially, $\sigma_p = \infty$, so $fit_p = 0$. This means the range of $fit_p$ is $0 \sim 1$. As $fit_p$ approaches 1, the function $Bug_p(i)$ approaches the observation data $Out_i$. Regarding fitness $fit_p$ as the internal energy of bug $p$, we can say that the calculation of the fitness corresponds to the acquisition of the internal energy of a bug. That is to say, the observation data is equivalent to “food” for the acquisition of internal energy.

Figure 4 is the flowchart of the algorithm of function-discovery by the use of a bug type of A-life in this study. The flow is summarized as follows.

(1) The evolution sometimes depends on the initially created chromosome of bug[5]. Accordingly, numerous bugs with the arbitrary function are generated at random as the initial condition. The number $Pop$ is selected from the numerous bugs in order of high fitness.

(2) The generation Gene of the bug is set to 0.

(3) The value of the internal energy, $energy_p$, of all the bugs is initialized to 0.

(4) The procedures from (5) to (7) are repeated for all the bugs; the bug number ranges from 1 to $Pop$. 

電学論 C, 118 巻 2 号, 平成 10 年 171
Fig. 4 The flowchart of the algorithm of function-discovery using the bug type of artificial life.

(5) The bug $p$ moves. That is to say, the values of constants $K$ in the chromosome change slightly. Here, let the small change of constants be $dK$, then the values of constants $K$ are replaced by $K + dK$, where $dK = (dK_1, dK_2, \ldots, dK_n)$ and $n$ is the number of constants in the chromosome. This concept is based on Ref. (7). The details are given in Sect. 2.6.

(6) The bug $p$ catches the observation data (i.e., fitness $fit_p$ of bug $p$ is calculated from the observation data).

(7) In the case that fitness $fit_p$ reaches the threshold fitness $Fit_{th}$, this algorithm ends. This means a bug determines the function $f$.

(8) The algorithm ends when the current generation $Gene$ reaches the maximum Generation $Gene_{max}$.

(9) After the descendant-generation-routine is called, $Gene$ is added to 1 and the algorithm returns to procedure (3).

2.4 Descendant-generation-routine

The descendant-generation-routine is performed for the production of excellent descendants. The flowchart is displayed in Fig. 5, and is summarized as follows.

(a) The bugs are selected in order of high fitness, and they are passed down to the next generation. The selected number is $Par$. This is based on the generation-gap. The elite strategy is adopted for the generation-gap.

(b) The bug number $p$ is set to be 2. By the repetition of the following procedures from (c) to (g), $Pop - Par$ bugs are generated.

(c) A bug is selected by the tournament strategy.

(d) The selected bug is inspected with respect to whether it has the ability to sexually reproduce. In the case that the selected bug has the ability of sexual reproduction, procedure (e) is performed. In the other case, procedure (f) is carried out.

(e) The bug finds its partner, and they produce two children by crossover. Jump to procedure (g).

(f) Two children are produced by asexual reproduction.

(g) A part of the chromosome is changed by mutation at the rate of $R_{mut}$.

Thus, the descendants of the number of $Pop$ are generated according as the flowchart in Fig. 5. The details
about selection, sexual/asexual reproduction and mutation are explained in Sects. 2.5, 2.6 and 2.7, respectively.

2.5 Selection

The fundamental rule of the selection is that the bug with high fitness bears more descendants than the bug with low fitness. The methods for selection include the roulette, rank and tournament. The tournament method is selected in this study. This has some excellent characteristics as follows:

1. This method can adjust the selection-pressure by parameters.
2. The selection-pressure does not change regardless of the direction of the search (i.e., it does not change even if the values of fitness are biased toward a side).
3. Parallel processing is easily realized.

2.6 Sexual reproduction and asexual reproduction

Sexual and asexual reproductions are introduced in this study. These are important for an effective function-search. First, we explain the concepts of homogeneous and heterogeneous species. The difference between them is shown in Fig. 6. We define the species with the same chromosome-structure as homogeneous species, while the ones with different chromosome as heterogeneous species. In Fig. 6, bug 1 has the same structure as bug 2, so bug 1 and bug 2 are homogeneous species. The chromosome structure of bug 3 is different from that of the others, so bug 3 is a heterogeneous species against the other bugs. Sexual reproduction is practiced between homogeneous species. Heterogeneous species cannot reproduce sexually.

The procedure of sexual reproduction is as follows. A bug is selected by the tournament strategy according to the procedure (c) in Fig. 5. The bug looks for the homogeneous species. In the case that multiple homogeneous species exist, the bug with the highest fitness is selected as the partner. The bug and its partner produce two children by crossover. Only the constants change by crossover. The example is shown in Fig. 7. In Fig. 7(a), only the constants change because they maintain the same chromosome-structure. The targets of crossover are $K_1, K_2, \ldots, K_6$. As far as sexual reproduction is concerned, the crossover operator is equivalent to that of real GA, as shown in Fig. 7(b). We adopt one-point crossover as the operator. Thus, sexual reproduction optimizes a specific function and corresponds to the strategy of real GA. In addition, we improve this strategy. It is reported that the local search ability increases considerably when the bug type GA is introduced instead of real GA. This strategy uses the small changes $dK$ of constants $K$ for a crossover. The example is displayed in Fig. 8. For the detailed procedure, see Ref. (7). When a bug moves, the values of constants
Table 1 The major axis-length of a plane orbit $D$ (normalized by the earth's length) vs a period of revolution $P$.

<table>
<thead>
<tr>
<th>$D$</th>
<th>$P$ [10$^4$ s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3871</td>
<td>0.760</td>
</tr>
<tr>
<td>0.7233</td>
<td>1.94</td>
</tr>
<tr>
<td>1.000</td>
<td>3.16</td>
</tr>
<tr>
<td>1.524</td>
<td>5.94</td>
</tr>
<tr>
<td>5.203</td>
<td>37.4</td>
</tr>
<tr>
<td>9.541</td>
<td>93.0</td>
</tr>
<tr>
<td>19.41</td>
<td>270</td>
</tr>
<tr>
<td>30.05</td>
<td>520</td>
</tr>
<tr>
<td>39.29</td>
<td>777</td>
</tr>
</tbody>
</table>

Fig. 9 Some examples of the replacement of a part of function $K_1 \cdot \sin(x) + y/(K_2 \cdot x)$ with $\tan(x \cdot y)$ by mutation.

Fig. 9(a) shows the replacement of the node $\times$ as $\sin$ by $\tan$ and Fig. 9(b) shows the replacement of the node $\div$ as $\times$ to $\tan$.

One part of the bug's chromosome is replaced by the other functions at a mutation-rate $R_{mu}$. The replacement, however, is sometimes unsuitable. For instance, when the node $\times$ is replaced by $\div$ of $K_1 \cdot \sin(x) + y/(K_2 \cdot x)$ in Fig. 3, it is not a valid function. To solve this problem, we replace the target node and all the lower nodes by the other functions as shown in Fig. 9. This is the same as GP$^3$. As understood from the dotted square in Fig. 9(a), the node $\div$ and the its leaf nodes are replaced by the other functions which are generated at random. In Fig. 9(b), the node $\times$ is the target. Since $\times$ is a terminal node, the only $\times$ is replaced by $\tan(x \cdot y)$.

In this study, only constants change when sexual reproduction is practiced. For increasing the search ability, we introduce the mutation-rate of constant $R_k$ in addition to the mutation-rate $R_{mu}$. At a rate of $R_k$, a constant included in a chromosome is replaced by an arbitrary value in the case of sexual reproduction.

2.8 Limit of the number of a homogeneous species

As generation progresses, the number of homogeneous bugs may be almost entirely dominated by one species. This is because a superior species survives selectively. In that case, it is difficult for the other heterogeneous bugs to appear even if the bug is highly superior. That is, the search is trapped in a local optimum. For the search to be as successful as possible, we limit the number of single type homogeneous species. By this strategy, many different species are in existence at the same time.

The search system proposed in this study has recently introduced the concept of sexual / asexual reproduction. For distinguishing this from GP, we call this S-system (System using Sexual / asexual reproduction).

3. Application to function-discovery

3.1 Kepler's law

For confirming the validity of the S-system, we attempted to search for functions using the observation data which obey Kepler's law, Ohm's law and Snell's law. The function fitting each law has been successfully discovered. In these results, the search for Kepler's law is mentioned here.

Let the major axis-length of a plane orbit be $D$ (normalized by the earth's length) and a period of revolution be $P$, then Kepler's law is expressed

$$\frac{D^3}{P^2} = \text{Const.} \quad (12)$$
Fig. 10 The relationship between generation and fitness $fit_p$ in the case of searching Kepler’s law.

To rearrange the expression, $P = K_1 \cdot D \cdot K_2^2$, where $K_1$ and $K_2$ are constants. Table 1 shows the relationship between $D$ and $P$ of some planets.

The observation data $(D, P)$ corresponds to the data $(In_{in}, Out)$ in Eq. (2). The conditions for the search are as follows.

2. Initially created population size: 6000.
3. Mutation rate of chromosome $R_{mu}$: 0.4.
4. Mutation rate of constant: 0.4.
5. Crossover rate $R_{cros}$: 0.2.
7. Threshold fitness $Fit_{th}$ for algorithm ends: 0.92.
8. Selection method of chromosome: tournament (size: 3).
9. Generation gap: 0.6.
10. Selection on generation gap: elite strategy.
13. Function set: $+$, $-$, $\times$, power.
15. Range of small change $\Delta K$ of constant: $-0.0025$ to $+0.0025$.

The system was practiced ten times, and satisfied Kepler’s law eight times. Example of the results are shown in Fig. 10. This figure indicates the relationship between the generation and the fitness of the bug with the highest fitness $fit_p$. The function is discovered perfectly in the case that $fit_p = 1$. Since errors are included in the observation data in reality, fitness cannot reach 1.0. Threshold fitness $Fit_{th}$ is, therefore, set at 0.92 in this system. As a result, the function of $P = K_1 \cdot D^{K_2}$ was discovered, where $K_1 = 316$ and $K_2 = 150$. The real form of the discovered function is different from generation to generation. The functions determined by the three examples in Fig. 10 are as follows.

$$P = (D \cdot K_1 + D)^{K_2},$$
$$P = D^{K_1} \cdot D \cdot K_2^{K_2},$$
$$P = (D \cdot K_1 + D)^{K_2}.$$  \hspace{1cm} (13)

To simplify these equations, they are represented as $P = K_1 \cdot D^{K_2}$.

### 3.2 Gloss measurement

The search of the function representing the psychological glossiness was carried out as the example of unknown function-discovery. Psychological glossiness depends on many psycho-physical factors. Table 2 displays the relationship among psychological glossiness $G_{ph}$, specular light-intensity $L$, diffused light-reflectance $Y$ and chroma $C$ of stainless steels with various degrees of roughness. We substitute these data into this system. The data $(L, Y, C, G_{ph})$ corresponds to the data $(In_{in}, In_{out})$ in Eq. (2). The conditions for search are as follows.

2. Initially created population size: 6000.
3. Mutation rate of chromosome $R_{mu}$: 0.3.
4. Mutation rate of constant: 0.4.
5. Crossover rate $R_{cros}$: 0.2.
7. Threshold fitness $Fit_{th}$ for algorithm ends: 0.92.
8. Selection method of chromosome: tournament (size: 3).
9. Generation gap: 0.6.
10. Selection on generation gap: elite strategy.

#### Table 2 Psychological glossiness $G_{ph}$ vs specular light-intensity $L$, diffused light-reflectance $Y$ and chroma $C$ of stainless steels.

<table>
<thead>
<tr>
<th>No.</th>
<th>$L$ (cd/m$^2$)</th>
<th>$Y$</th>
<th>$C$</th>
<th>$G_{ph}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1320</td>
<td>0.410</td>
<td>0.412</td>
<td>3.12</td>
</tr>
<tr>
<td>B</td>
<td>645</td>
<td>0.350</td>
<td>0.348</td>
<td>1.56</td>
</tr>
<tr>
<td>C</td>
<td>1015</td>
<td>0.165</td>
<td>0.386</td>
<td>2.71</td>
</tr>
<tr>
<td>D</td>
<td>685</td>
<td>0.490</td>
<td>0.383</td>
<td>1.29</td>
</tr>
<tr>
<td>E</td>
<td>1105</td>
<td>0.537</td>
<td>0.309</td>
<td>2.50</td>
</tr>
<tr>
<td>F</td>
<td>645</td>
<td>0.234</td>
<td>0.397</td>
<td>1.45</td>
</tr>
<tr>
<td>G</td>
<td>568</td>
<td>0.462</td>
<td>0.418</td>
<td>1.49</td>
</tr>
<tr>
<td>H</td>
<td>380</td>
<td>0.147</td>
<td>0.438</td>
<td>1.47</td>
</tr>
<tr>
<td>I</td>
<td>2030</td>
<td>0.169</td>
<td>0.402</td>
<td>3.41</td>
</tr>
<tr>
<td>J</td>
<td>233</td>
<td>0.523</td>
<td>0.432</td>
<td>1.13</td>
</tr>
</tbody>
</table>
Fig. 11 Relationship between psychological glossiness $G_{ph}$ and glossiness $G_{ph}'$ discovered by A-life.

1. Maximum bug number of a homogeneous species: 10.
3. Function set: +, −, ×, power.
4. Range of constant $K$ generated by mutation: $-10 \rightarrow +10$.
5. Range of small change $dK$ of constant: $-0.0025 \rightarrow +0.0025$.

As the result, the following function $G_{ph}'$ is obtained at the generation of 573.

$$G_{ph}' = K_1 \cdot (L - K_2 \cdot Y + K_3 \cdot C - K_4)^{K_5}.$$  (14)

where, $K_1=0.018$, $K_2=8.258$, $K_3=244.011$, $K_4=-64.207$ and $K_5=0.69$. Equation (14) is simplified. Figure 11 expresses the relationship between psychological glossiness $G_{ph}$ and discovered function $G_{ph}'$. The correlation coefficient between them is shown in the lower right-hand side of Fig. 11. It is understood that the psychological glossiness $G_{ph}$ agrees with $G_{ph}'$ because of a high correlation coefficient.

This equation implies that $L$ and $C$ have a positive effect but $Y$ has a negative one on $G_{ph}'$. These results are in agreement with the effects mentioned in Ref. (10). Although the discovered function may have nothing to do with the physical meaning, it is important to define the glossiness in agreement with psychological glossiness for the classification in the industrial field.

As far as the search for glossiness is concerned, some functions are discovered at every evolution. This is because the observation data include numerous errors. Human beings also estimate numerous functions when many errors are included in the observation data.

4. Investigations

4.1 Comparison with GP

GP is the extension of GA. It can handle structural expressions such as the graph-structure and the tree-structure. The main procedure of GP consists of the following:

- **G mutation**: label change of node.
- **G inversion**: rearrangement of brothers (rearrangement of the nodes at the same stage).
- **G crossover**: exchange of partial trees.

These are natural extensions of the GA operator. The optimization of values is the purpose for GA, while the realization of symbolic processing is that for GP. GP is presently utilized for various fields and the validity is confirmed. A function-search is also tried using GP because a function is expressed by a tree structure. There are, however, three disadvantages for function-search.

These disadvantages and the methods for improvement using the S-system are as follows.

1. GP blindly applies a mutation and a crossover. Figure 12 shows the example of a crossover in a GP operator. Since the crossover is carried out between different functions, parents and their descendants differ considerably in chromosome structure. As a result, the schema (profitable partial tree) tends to be broken. The gene with this schema does not always survive.

**S-system**: the tree structure does not change due to a crossover because of the introduction of sexual/asexual reproduction. The bugs with this schema evolve without breakup. However, the crossover of the S-system lacks the variety of functions. For the variety, we introduce a mutation.
Fig. 13 The relationship between generation and fitness $fit_p$ in case of searching Kepler's law using the function-set of $\oplus$, $\ominus$, $\times$, power on GP.

and a limitation of the number of homogeneous bugs.

2. The design of the node is very important for GP. This is because the meaning of the tree dramatically changes by the crossover and mutation. As the result, the search does not stabilize. That is, the solution tends to diverge.

**S-system**: The meaning of the tree remains completely unchanged by the crossover. Because of the introduction of sexual/asexual reproduction and bug type GA(6), the solution stabilizes.

3. There is no method for the control of tree growth, so the function length often becomes very long or extremely short as the search progresses.

**S-system**: In the early generation, the bugs with long function lengths sometimes appear. They, however, do not hold a majority because of the limitation of the number of homogeneous bugs. A bug with true function, whose function length is suitable, appears in due time. It evolves without the breakup of the schema, and achieves a high fitness. Thus, the function length is not too long.

To illustrate an actual example, we practiced the GP operator for the search of Kepler's law. We included the term for power in addition to $\oplus$, $\ominus$, $\times$, variables and constants, because almost all the laws which were discovered in the earlier part of this century included the terms for power and constants(6). The result using GP is shown in Fig. 13. Although the GP procedure was attempted three times, the function in agreement with Kepler's law was not obtained. As understood from this figure, the fitness $fit_p$ remains low even if the generation proceeds. When the generation reached the maximum value, $Gene_{max}$ (=400), the algorithm was aborted. At $Gene_{max}$, the function with the highest fitness $fit_p$ at each trial is as follows.

$$P = K_1 \cdot K_2 \cdot D^2 - K_3 \cdot D + K_4,$$

$$P = K_2 \cdot K_3 \cdot K_4 \cdot K_5 \cdot K_6 \cdot K_7 \cdot K_8 \cdot K_9 \cdot K_{10} \cdot K_{11} \cdot K_{12} \cdot D \cdot K_{13},$$

$$P = (D - K_9)^{K_2} \cdot \{K_3 \cdot (D - K_4)\}^{K_5} \cdot K_6 \cdot K_7 \cdot K_8 \cdot K_9 \cdot K_{10} \cdot K_{11} \cdot K_{12} \cdot K_{13} \cdot K_8.$$ (15)

From Eq. (15), it is obvious that the functions discovered by GP are quite different from that of Kepler's law. In addition, the function is long.

For the quantitative comparison of GP and the S-system, we calculate the value of MDL(11); MDL is the value which takes into account a tradeoff of tree-complexity for approximate error. The value of MDL for tree structure is given as follows(8).

$$MDL = 0.5N \cdot \log(S_e^2) + 0.5k \cdot \log(N),$$ (16)

where $N$ is the number of data, $S_e^2$ is the least square error and $k$ is the parameter number of the tree. A low value of $MDL$ means that the function is excellent, while a high value indicates that the function is unsuitable. For Kepler's law, we calculate the values of $MDL$ ten times. The mean value $m_{MDL}$ and standard deviation $\sigma_{MDL}$ are shown in Table 3. Obviously, the value of $m_{MDL}$ using the S-system is lower than that using GP although the values of $\sigma_{MDL}$ between them are similar.

### 4.2 Problems of the S-system and their prospective improvements

The present limitations of the S-system and their prospective improvements are mentioned here.

(1) The S-system is not suitable for a field which requires fast operation. Similarly to GP, it takes an excessive amount of calculation. This is an inherit problem regarding Genetic approach. For the solution, the same approach as that used Distributed GP should be incorporated(13).
It is impossible to search for the functions which are not represented by a tree structure (for example, square wave, step function and sawtooth waveform). A method for representing these as tree-structures should be considered. For example, the normalized square wave is determined by period $x$ and duty ratio $y$. Therefore, by defining a square wave as the function $Square(x, y)$, we can express a square wave as a tree structure.

The adaptation of a bug in some cases cannot be calculated. For example, when a bug has the function $x/y$ as chromosome, it is impossible to substitute the observation data ($x=1, y=0$) for the function. In this case, we forcibly set the bug's adaptation as 0. This is valid as the function which cannot express the observation data is never a true function and the bug with this function does not need to evolve.

The solution differs with every trial in the case of the data including a large error. As for the S-system, the form of a function does not dramatically change compared to GP. Therefore, if the form of a function is roughly estimated by a human being, it is beneficial to adopt the function as that of the initially generated bug.

A problem requiring immediate attention is the optimization of parameters. The S-system has numerous parameters for search, as mentioned in Kepler's law. These are not optimized, so it is unclear which parameter is the most effective.

Although the S-system is a new system which improves the breakout of a schema, the stability of the solution and function length, this is basically the extension of GP. Therefore, the S-system is available for fields utilizing GP such as image measurement, image recognition and system identification. As an example, we show an application of the S-system to image recognition.

### 5. Possibility of the Application to Image Recognition

The lines or curves in image data are extracted by the use of the Hough transformation. The form of the function must, however, be determined in advance. It is impossible to extract the variously different functions simultaneously. For examining whether this system is adaptable to the extraction of different functions, we try to extract the unknown functions from a sampling image data. The example is displayed in Fig. 14. The black dots are the sampling points of an ideal yacht-image, which includes neither distortion of the image nor any sampling error. The initial conditions for function-search are as follows.

- **Population size:** 600.

![Fig. 14 Sampling data of ideal yacht image and extracted curves $l_1, \ldots, l_6$ from them.](image)

- **Initially created population size:** 10000.
- **Mutation rate of chromosome $R_{mu}$:** 0.4.
- **Mutation rate of constant:** 0.4.
- **Crossover rate $R_{mx}$:** 0.2.
- **Crossover method:** one-point crossover.
- **Threshold fitness $Fit_{th}$ for algorithm ends:** 0.95.
- **Selection method of chromosome:** tournament (size: 3).
- **Generation gap:** 0.6.
- **Selection on generation gap:** elite strategy.
- **Maximum bug number of a homogeneous species:** 10.
- **Maximum chromosome size:** 20.
- **Function set:** $\{+,-,\times\}$.
- **Range of constant $K$ generated by mutation:** $-10 \sim +10$.
- **Range of small change $\delta K$ of constant:** $-0.0025 \sim +0.0025$.

Six different curves are included in Fig. 14. Since multiple curves cannot be represented by one function, the fitness cannot reach 1.0. In order to avoid this problem, we added the following four strategies.

1. Only the data within the local search area $A_{loc}$ are used for the calculation of fitness instead of all the data. The local search area $A_{loc}$ is defined by Eq. (17).
   \[
   A_{loc} = 2^{\alpha u}, \quad (u=0, 1, 2, \ldots).
   \]
   The data satisfying the following equation are regarded to be the data within $A_{loc}$.
   \[
   A_{loc} \times \{|Out_{ix} - Bug_{ix}(t)|\}.
   \]
   The value of $u$ is set to be 0 at first. Until the number of data within $A_{loc}$ exceeds five, $u$ is added to 1 ($u \leftarrow u + 1$). When six data or more are obtained within $A_{loc}$, the fitness of the bug is calculated.
   By this procedure, the fitness of curves $l_1, \ldots, l_6$ in Fig. 14 reached to more than 0.95.

2. The sampling data fitting the curve are rejected from
the entire data set as soon as the function is discovered. For example, when the curve \( l_1 \) is discovered in Fig. 14, the data drawn on \( l_1 \) are rejected. The searching time is shortened greatly by this procedure.

(3) All the initially generated bugs have the chromosome of the function \( K_1 \cdot x^2 + K_2 \cdot x + K_3 \), where the values of \( K_1, K_2, K_3 \) are generated at random.

(4) When the fitness does not change during the successive 200 generations, the algorithm shown in Fig.4 restarts. As a result, the curves \( l_1 \ldots l_6 \) are obtained by the use of these strategies. From Fig. 14, we see that the image consists of six different curves.

As mentioned above, it is possible to apply A-life to the field of image recognition. However, since the errors are included and there are many sampling data in the real image, this system should be improved in the future.

6. Conclusions

The effectiveness of artificial life is investigated from an engineering point of view. The function-search system (named S-system) using a bug type of artificial life is proposed in this study. The chromosome of a bug consists of functions, constants and variables. For the expression of a function as a chromosome, a tree structure is utilized in the same way as GP. Observation data are provided for bugs. They move and catch the data. They produce offspring. The concept of a sexual / asexual reproduction and the limitation of the number of homogeneous bugs are newly introduced. As the generation proceeds, the bugs with the functions in agreement with the observation data survive selectively, and determine the function corresponding with the given data.

For the validity of this system, some data which obey the known laws were given for them. They evolved and finally discovered the functions in agreement with the laws. As for an unknown function, the observation data on glossiness were provided. After the evolution, they found the function in agreement with glossiness. Furthermore, the sampling data for an image were supplied. After the evolution proceeded, they extracted some different curves included in the image. In this way, artificial life has a potential for searching the unknown function.

This work was supported in part by the Grant-in-Aid for Exploratory Research from the Ministry of Education, Science, Sports and Culture.

(Manuscript received Nov. 25, ’96, revised Sept. 16, ’97)

References


Seiichi Serikawa (Member) He received his B.S. and M.S. degrees from Kumamoto University in 1984, 1986 and his Dr. Eng. from Kyushu Institute of Technology in 1994. Currently, he is a research associate of the Department of Electronics Engineering, Kyushu Institute of Technology. His research interests include human visual information and artificial life.

Teruo Shimomura (Member) He received his B.S. and M.S. degrees from Kyushu Institute of Technology in 1969, 1971 and his Dr. Eng. from Tokyo Institute of Technology in 1979. Currently, he is a Professor in the Department of Electronics Engineering, Kyushu Institute of Technology. His research interests include color of LCD and ECD and visual information processing.