EMACI: A Visual Interface of Multi-Parameters for Interactive Evolutionary Algorithm

Kisa Ohyama[*], *Takayuki Itoh*[*], *Fumiyoshi Yamashita*[**], *Koji Koyamada*[***]
*Graduate School of Humanitics and Sciences, Ochanomizu University
**Graduate School of Pharmaceutical Sciences, Kyoto University
***Center for the Promotion of Excellence in Higher Education, Kyoto University

ABSTRACT

This paper proposes a technique for visualizing the progression of evolutionary algorithms such as genetic algorithms (GAs). Our technique supposes that a GA solves optimization problems that maximize target functions with multiple parameters. Then, our technique presents the time-varying parameters during the progression of the GA. The technique applies two-tone pseudo-coloring for the precise representation of changes in parameter values, so that users can understand the behavior of the GA. We also provide a user interface to control the configuration of the GA so that users can obtain the optimal solution earlier. The paper shows experimental results to demonstrate the efficacy of the technique.

1. INTRODUCTION

An Evolutionary Algorithm (EA) is a very powerful optimization scheme that has been applied to various technical fields. A Genetic Algorithm (GA), one of the most famous EA schemes, is especially useful for problems that lack clear solutions and that cannot be solved using full search schemes because the GA's parameter space is huge and unstructured. A GA evaluates the fitness of an organism to an environment [1], based on principles of Darwin's evolutionary theory. Suppose that the genes of animals correspond to the input parameters of certain functions, and the fitness values correspond to the functions' return values. Here, the GA gradually discovers the optimal solution of the problem according to the evolution of the input parameters.

The following points are important to quickly discover optimal solutions using a GA:

- Coding input parameters as genes, and
- Controlling configurations for evolution of genes.

We especially need to effectively and carefully configure GAs while observing the exploration of parameter spaces according to the evolution of genes. However, it is generally difficult to fully automate controling the configuration of GAs. Therefore, several studies focus on interactive implementations of GAs to effectively control the exploration of parameter spaces.

This paper proposes EMACI (Evolutional Multi-parameter Analysis and Control Interface), a visual interface of the process of GAs. Regarding the parameters and target function as a multi-dimensional data, EMACI represents the repetitive process of a GA as time-sequence multi-dimensional data. EMACI also provides a user interface to effectively control the configuration while users observe the visualization results. This user interface helps users to control the configuration intuitively, and consequently to discover optimal solutions in shorter times.

The naming of EMACI comes from "Emakimono", hand-scrolled, horizontally illustrated narratives created during Heian, Kamakura, and Muromachi periods in Japan; we feel the technique looks like Emakimono since it displays the time sequence of multi-parameters as a horizontal spread.

This paper also introduces an application of the EMACI to the parameter optimization of heart cell simulations. This optimization problem defines the target function as the error of simulation result of time-varying electric potential against the real measurement, with parameters as density or other measurements of ions such as K^+ and Na^+. We need to repeat the simulation inputting a variety of parameters, in order to discover the optimal solution [2]. We are applying EMACI to this optimization problem, by coding the measurements of ions as genes, and by repeating the simulation with the evolution of the parameters by the GA.

2. RELATED WORK

2.1 Genetic Algorithm

A Generic Algorithm (GA) is an optimization scheme inspired by Darwin's evolutionary theory, which consists of several processes mimicking evolutionary processes of animals, including selection, crossover, and mutation [1].

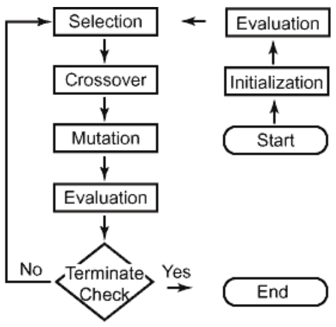


Figure 1: Processing flow of a GA.

Figure 1 shows a typical processing flow of a GA. It consists of the following key processes:

Initialization: randomly generates initial genes.
Evaluation: calculates the fitness values for each gene, and sorts the genes according to the evaluation result.
Selection: extracts two parents, where their probabilities are weighted according to the evaluation results.
Crossover: generates children genes from the selected two genes after the Selection process. The GA repeats the Selection and Crossover processes until it generates the specified number of children. Here we may apply "elite"

reproduction, which just copies the genes of highly fit parents to their children as clones.

Mutation: randomly modifies the genes on a given probability. This operation corresponds to a failure to expresscertain DNA. Mutation is a useful technique to break out local optimal solutions and to get closer to the global optimal solution.

Terminate Check: stops the repetition of the GA according to predefined conditions.

This implementation mimics the gametogony of animals, during which the exchange of genes of selected individuals results in more conformable children. The repetition of GA processes leads to the optimal solution of target functions.

2.2. Visualization for Genetic Algorithm

Many studies involving GAs have attempted to analyze the trajectory of parameter spaces and the evolution of target functions, to understand the effort of methodology and parameters for the quick and certain exploration. It is effective to display the evolution of target functions as time-sequence data. However, many GA-applied optimization problems have multiple parameters. Therefore, it is generally difficult to understand the relations between the parameters and the target function, if one simply draws them as polygonal charts. Visualization techniques are desirable so that we can visually understand the exploration of parameter spaces and the evolution of GA processes.

Moreover, it often happens that the GA falls into local optimal solutions from which breaking out is difficult. Visualization of parameter space exploration is useful to understand the causes of the falls and to discuss the strategy to break out of the local optimal solutions.

Interactive, visual implementations of GAs are useful since users can control the exploration of parameter spaces while they observe the repetition of the GAs.

The performance of the parameter space exploration of a GA depends on not only on methodology but also on many configurations, such as the number of individuals, the probability of mutation, and the probability of crossover. It is generally difficult to develop conclusive techniques to completely, automatically control the configuration. The interactive implementation of a GA is useful because one can control the configuration via a user interface, so that one can thereby improve the parameter space exploration. Visualization of GA processes helps to support configuration control.

There are several recent works on the visualization of GA processes from the above viewpoint. The following are examples of related works on the visualization of GA processes:

- Visualization of multi-parameter spaces using Self Organizing Map (SOM) [3]. The technique represents clusters of solutions to easily understand the exploration of the space. The technique has been applied to some multi-criterion optimization problems.
- Visualization of solutions led by Interactive Evolutionary Algorithm (IEA) and Interactive Genetic Algorithm (IGA) [4]. The technique enables subjective control of GAs, since users interactively evaluate the solutions.
- Visualization of the landscape of target function values [5]. The technique codes multi-dimensional parameters as input vectors, and summarizes them into a 2-dimensional display space. It can represent the distribution of parameters during the repetition of a GA process.

Against the above existing work, our technique focuses on the representation of time sequence changes in parameter values for real-time adjustment of configurations of GA.

2.3. Multi-Dimensional Data Visualization

As mentioned above, many GA-applied problems optimize target functions of multi-dimensional parameters. Therefore, the visualization of parameter spaces corresponds to the visualization of multi-dimensional time-sequence data.

The following are typical multi-dimensional data visualization techniques. Many of such famous multi-dimensional data visualization techniques originally deal with static data, and visualization of time-varying multi-dimensional data is still a hot and changing topic.

- Parallel Coordinates [6]. The technique draws vertical lines in a 2-dimensional orthogonal coordinate system, where each line corresponds to each dimension, and generates polygonal charts by plotting values of the input information onto the vertical lines.
- Worlds within Worlds [7]. The technique assigns 3 variables of the input data as each axis of 3-dimensional orthogonal coordinate system, and plots each data item. It then assigns other 3 variables as each axis of a 3-dimensional orthogonal coordinate system located inside a part of the first world.
- Several well-known techniques project the input multi-dimensional data onto 2- or 3-dimensional spaces using dimension reduction techniques. Design Galleries [8] is a well-known technique applying such dimension reduction technique.
- Several techniques apply "Glyphs" to represent multi-dimensional values by varying their shapes, sizes, and colors. Ebert's technique [9] is an example of multi-dimensional data visualization using Glyphs.

2.4. Two-Tone Pseudo Coloring

The technique proposed in this paper applies two-tone pseudo coloring [10]. It assigns two colors to a one-dimensional range, and represents the value by controlling the widths of the two colors. It is very suitable to precisely represent long-scale time-sequence data in a small display space. Also, it represents both the overview and details of the dataset in one image.

Figure 2 illustrates the visualization of time-sequence data by two-tone pseudo coloring and traditional polygonal charts. In the final phase when the data slightly changes, two-tone pseudo coloring captures the fine details. In contrast, it would be difficult to recognize the changes in the traditional polygonal chart.

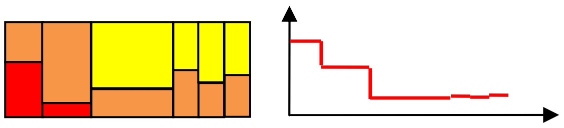


Figure 2. (Left) Visualization by two-tone pseudo color. (Right) Visualization by traditional polygonal chart.

3. EMACI: THE PROPOSED TECHNIQUE

Let us define a target function of an optimization problem as:

$$s = f(x_1, x_2, \ldots, x_n) \quad \ldots(1)$$

where we would like to discover the maximum s value and the set of parameters x_1 to x_n. Here EMACI visualizes the progression of a GA and provides a user interface to interactively control the configuration of a GA.

Processing flow of EMACI is as follows:

1. Initialize the GA.
2. Perform selection, crossover, and mutation, as shown in Figure 1.
3. Evaluate the genes and specify the maximum s value, and x_1 to x_n which produce the maximum s.
4. Visualize x_1 to x_n using two-tone pseudo coloring.
5. Perform terminate check, as shown in Figure 1, and terminate if the progression satisfies the conditions.
6. Occasionally pop-up a dialog window to control the configuration of the GA, and close the dialog after a while (*i.e.*, 10 seconds later).
7. Return to 2.

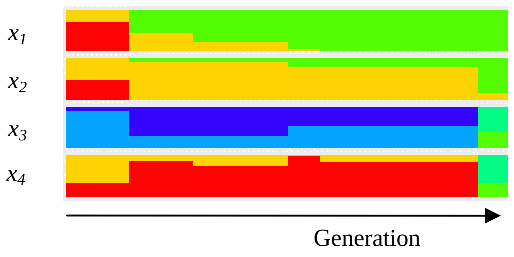


Figure 3. Visualization of parameters.

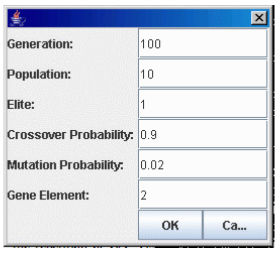
Figure 3 shows an example of visualization of 4 parameters during the progression of a GA. The horizontal axis denotes the generation of the GA, and colored belts represent the changes in parameter values. Here we represent the parameter values as follows, similar to that which is described in [10]:

- Specify the range of a parameter x_i as $[\min_i, \max_i]$.
- Define intervals $B_i = (\max_i - \min_i)/m$.
- Let $A_{ij} = \min_i + jB_i$.
- Define $(m+1)$ colors C_0 to C_m.
- For each generation,
  - Specify j that satisfies $A_{ij} < x_i < A_{i(j+1)}$.
  - Paint the vertical line segment at x_i with two colors C_j and $C_{(j+1)}$. Divide the line segment into two parts so that the ratio of the length of the lower part is $(x_i - A_{ij})/B_i$, and use C_j for the upper part and $C_{(j+1)}$ for the lower part.

This visualization represents changes in parameter values as the changes of combinations or widths of two colors. It can inform users of slight changes in parameter values, as discussed in Section 2.4. Users can acquire a great deal of information from the visualization, for example:

- Even if every parameter is stable in long generations, one may not be sure whether the progression has already arrived at the global optimal solution. One may want to encourage mutations to drastically move the parameter space.
- A specific parameter is always stable in whole generations. One may want to narrow the range of the parameter for a more stable progression and accurate solution.

EMACI provides a dialog window to change the configuration of the GA so that users can interactively employ their acquired knowledge from the visualization. Figure 4 is an example of the dialog window. Our implementation can control the frequency of the pop-up of the dialog window: pop-up for every generation, once in several generations, only when every parameter is stable for long generations, and so on.



Generation:	100
Population:	10
Elite:	1
Crossover Probability:	0.9
Mutation Probability:	0.02
Gene Element:	2

Figure 4. Example of a dialog window to control the configuration of GA.

4. EXAMPLE

We implemented EMACI in Java 1.5, and executed it on an IBM ThinkCentre with Windows XP. Our current implementation defines 6 colors C_0 to C_5, which are blue, sky blue, moss green, yellowish green, bright yellow, and red. Figure 5 denotes the colors and ratio of their lengths in our implementation.

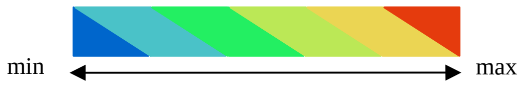


Figure 5. Colors used in our implementation.

As a simple example problem, we also developed a target function which requires 10 parameters and contains 6 local optimal solutions. Ranges of all parameters are 0.0 and 1.0. Return values of the target function are normalized so that the return value of the global optimal solution is 1.0.

Figure 8 shows an example of the visualization of the time sequence of 10 parameters during the GA process. The upper part of the figure shows the best parameters during 1 to 75 generations, and the lower part shows them during 76 to 150 generations. The figure denotes that the values of the best parameters drastically changed during generations1 to 40, then stayed during generations 40 to 90 , and slightly changed several times during generations 90 to 150.

When the best parameters stay during long generations, it is often better to change the configuration of the GA. Our current implementation pops-up the dialog window shown in Figure 4, when the best parameters do not change during 10 generations. In this experiment we changed the following configurations: the probability of crossover, the probability of mutation, and the percentage of elites.

Table 1 shows the statistics of the GA results based on changing configurations during the repetition of the GA process. In this table "Global optimal" denotes the ratio of results that GA arrived around the global optimal solution, not local optimal solutions, where larger values are better. "Average" denotes the average of the best values of the target function, where larger values are better. The results that the interactive changes in configuration values brought better results: a change in the probability of crossover and the percentage of elites brought better values of target functions, and the probability of mutation brought a better ratio approximating the global optimal solution. We would like to more carefully analyze the effect of interactive change in future work. We also would like to have more experiments with more complicated target functions, additional configuration changes, and non-expert examinees.

Table 1 statistics of GA results changing the configurations.

Interactive change	Global optimal	Average
None	0.48	0.9781
Probability of crossover	0.52	0.9911
Probability of mutation	0.88	0.9729
Percentage of elites	0.60	0.9817

5. APPLICATION TO CELL SIMULATION

This section introduces an application of EMACI to the parameter optimization of heart cell simulation [2].

The human body is a very complicated mechanism, and analyzing it is therefore a difficult problem. Recently there have been many studies to compute the biological functions, including organs, blood circulation, and energy vicissitude, to clarify this complicated mechanism. Cell simulation is an active research topic, because the cell is the smallest and most fundamental unit of human body.

We are applying EMACI to a heart cell simulation model [2]. The model is designed so that it realistically reacts according to an environmental change or an external impulse. We expect that such computer simulations can contribute not only to clarify the complicated mechanism of the human body, but also to improve advances in medical treatments and drug design.

The heart supplies blood to the entire body during the life of a human. This organ is comprised of muscular pumps, and it continues to work thanks to action potential and to ions around heart cells. Pulses of the action potential are generated approximately once per second, and then densities of the ions drastically change according to the changes of the action potentials. These pulses cause the heart muscles to contract, pushing blood outside the heart. All the heart cells work together for this mechanism to make a heartbeat.

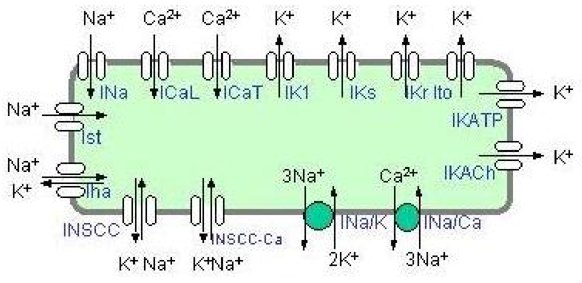


Figure 6. Mechanism of ion channels. (from [11]).

Changes in densities of ions result in the pulses of action potential. Here, ion channels of cell membranes work as doors to control the densities of ions. Cell membranes themselves do not allow for the transport of ions; however, the ions can transport when so-called ion channels open. Each ion channel transports only its specific ion: therefore, the ion channels are categorized according to the specific ion, such as the natrium channel, the kalium channel, the calcium channel, and so on. Figure 6 briefly denotes the mechanism.

The cell simulation model [2] calculates the temporal value of the action potential of the cell, and its error against the real measurement. Our approach treats the error as the target function, and the related values of ions—such as electric current—as parameters. EMACI visualizes the exploration of the values of ions to discover the optimal solution of the cell simulation.

Figure 7 shows an example of a visualization of a GA process for heart cell simulation. Here our implementation applies 4 parameters shown in Table 2 during the GA process. This figure shows that parameters were drastically changing during the early stage of GA process, but that the parameter space stabilized during the later stage.

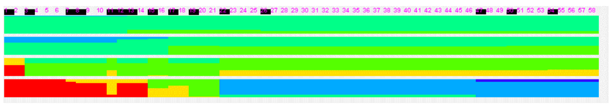


Figure 7. Example of a visualization of a GA process for heart cell simulation.

Table 2. Parameters applied in our implementation.

Name	Description	Range (pA)
ICaL	Voltage-dependent L-type Ca+ channel.	[4500.0, 5500.0]
IK1	Inwardly-rectifying K+ channel.	[1.0, 3.0]
IKr	Quickly-delayed-rectifier K+ channel.	[0.025, 0.045]
IKs	Slowly-delayed-rectifier K+ channel.	[0.01, 0.05]

6. CONCLUSTION

This paper presented EMACI, a technique for the visualization of GA processes. It represents the time sequence of multi-dimensional parameters by applying two-tone pseudo coloring, so that users can easily look over the change in parameter values during the GA processes. It also provides a user interface to flexibly control the configuration of the GA process. The paper provided the results demonstrating the effect of EMACI, as well as an application to cell simulation.
Our future work will include the following issues:

- Feasibility tests using target functions which require large numbers of parameters.
- Experiments with additional configuration changes, such as changes in ranges of parameters.
- Experiments with non-expert users.
- Proof of the contribution of EMACI to the cell simulation.
- Combination with other visualization techniques [3,4].

ACKNOWLEDGEMENTS

We appreciate Prof. Nobuaki Sarai, Dr. Lu Jianyin, and Ms. Yukiko Yamashita, for their advice on heart cell simulation.

REFEERNCES

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning , Addison-Wasley Publishing Company (1989).

[2] N. Sarai, S. Matsuoka, A. Noma, A. Amano, T. Matsuda, Object-Oriented Biodynamic Simulation Platform Based on the Biological View, Journal of Japan Society for Simulation Technology, Vol. 23, No. 1, pp.4-13 (2004).

[3] D. Yamashiro., K. Yamamoto, T. Yoshikawa, T. Furuhashi, Understanding Effects on Genetic Operators with Visualization of Search Process, Applied FCS/Techo-Sympo/MPS Symposium, pp. 33-38 (2005).

[4] H. Takagi, Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation, Proceedings of IEEE, Vol. 89, No. 9, pp. 1275-1296 (2001).

[5] N. Hayashida, H. Takagi, “Acceleration of EC Convergence with Landscape Visualization and Human Intervention,” applied Soft Computing, Vol. 1, No. 4F, pp. 245-256 (2002).

[6] A. Inselberg, B. Dimsdale, Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry, IEEE Visualization '90, pp. 35-38 (1990).

[7] S. Feiner, C. Beshers, Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds, ACM Symposium on User Interface Software and Technology (UIST'90), pp. 76-83 (1990).

[8] J. Marks, et al., Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation, ACM SIGGRAPH '97, pp. 389-400 (1997).

[9] D. S. Ebert, et al., Automatic Shape Interpolation for Glyph-based Information Visualization, IEEE Visualization '97 (1997).

[10] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, T. Kaseda, Two-Tone Pseudo Coloring: Compact Visualization for One-Dimensional Data, IEEE Information Visualization ’05, pp. 173-180 (2005).

[11] http://www.med.akita-u.ac.jp/~yakuri/km_home/index_jp.html

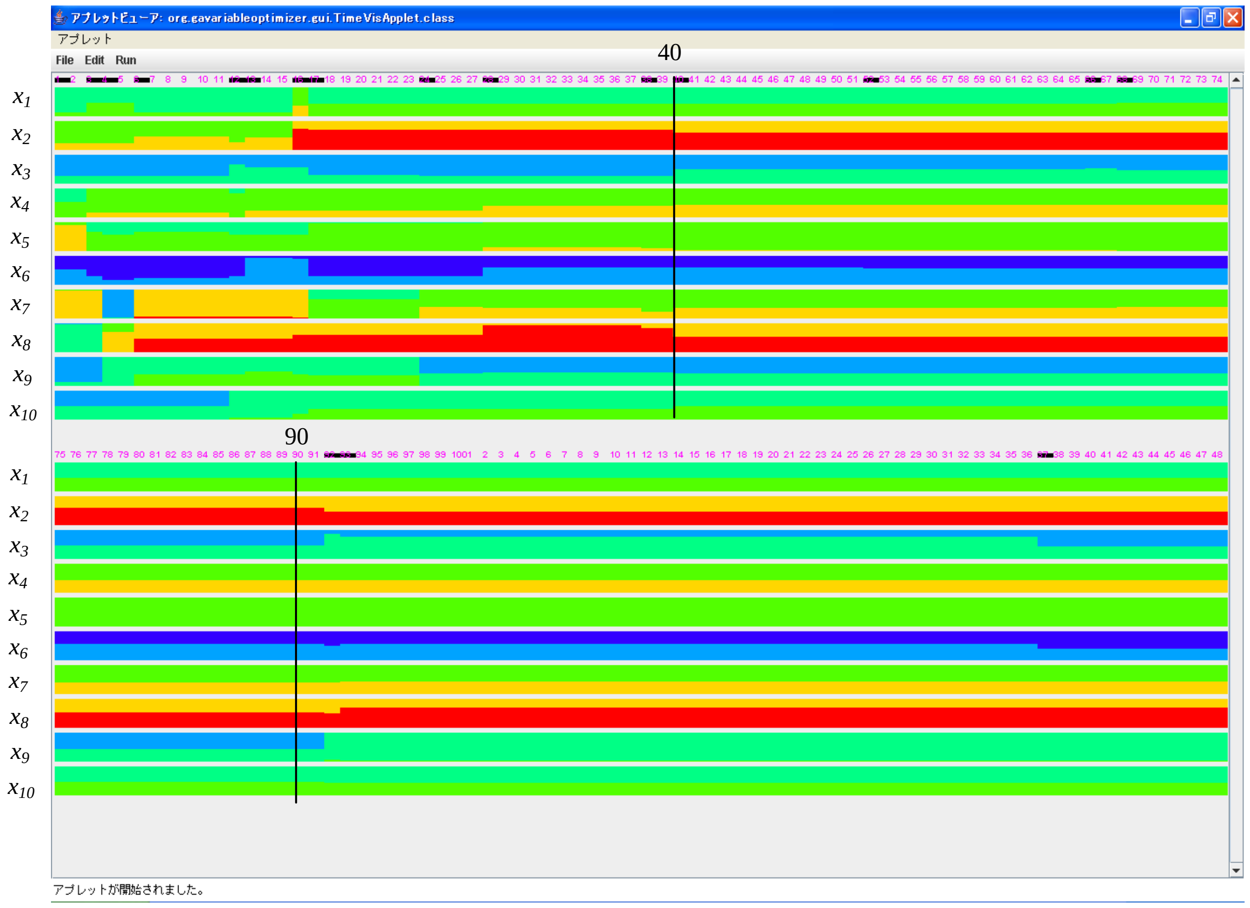

アプレットビューア: org.gavariableoptimizer.gui.TimeVisApplet.class
アプレット
File Edit Run
40
90
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
アプレットが開始されました。

Figure 8. Example of visualization of time sequence of 10 parameters during the GA process.