

## Partial Breeding – a method of IEC for well-structured large scale target domains

Tatsuo Unemi

Department of Information Systems Science  
Soka University  
1-236 Tangi-machi, Hachioji, Tokyo, 192-8577 Japan  
unemi@iss.soka.ac.jp

*Abstract*— This paper proposes a method of Interactive Evolutionary Computation (IEC) for large scale target domains of which structure is well organized. In general, it is effective to divide a large problem into a number of sub-problems to solve it efficiently. For IEC tools, it becomes possible by adding a GUI-based facility for the user to indicate some part of genotype protected against random initialization and mutation. The user can try alternative candidates only on the non-protected parts as a sub-problem through breeding process: iteration of subjective selection, mutation, and crossover. This method was invented through development of an application for musical composition support system, but it is useful for other domains, such as graphics.

*Keywords*— Interactive evolutionary computation, Problem decomposition, Large scale problem.

### I. INTRODUCTION

Interactive Evolutionary Computing (IEC) [1] is a promising technique to find better solutions in the domains for optimization by user's subjective criteria. Differently from ordinary methods of evolutionary computing, fitness values are not calculated automatically by the predefined evaluation function but are given by the user for each individual in some manner. IEC technique will provide useful help for bridging the digital divide because it makes the user able to obtain his/her favorite design without any domain knowledge on constructive procedure.

A number of researchers have tried to build applications of IEC in various domains, such as CG arts, music, Web page design, image filtering, image retrieval, design support systems, and so on. One of the important issues to make them successful is the design of genotype, that is, how we should encode the solution candidates into the computer memory. It strongly affects the process of evolution since it gives restriction of the search space. Obviously, the more complex the target domain is, the harder to make an appropriate design of genotype. It requires higher computational cost and brings more user's fatigue as search space becomes combinatorially larger.

This paper proposes a framework to solve this problem by dividing a genotype into some parts that can be bred independently. This method was invented through our project to build a musical composition support system, named SBEAT, utilizing IEC technique [2], [3]. To certify the flexibility of the method, another example of

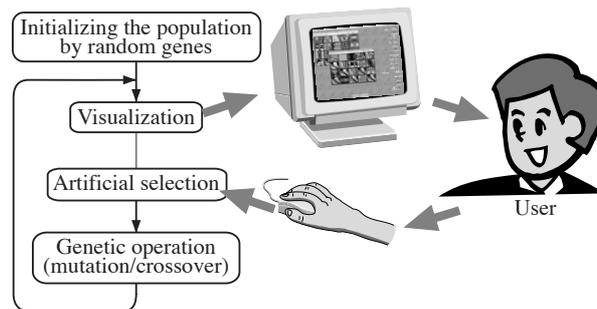


Fig. 1. A schematic framework of Interactive Evolutionary Computing (IEC).

the application, named SBTtitle, for a domain of graphical title image is also shown. The following sections describe an overview of the basic concept and examples of GUI design and breeding processes.

### II. APPLYING IEC TO LARGE SCALE PROBLEMS

Evolutionary computing (EC) is an optimization technique to find better solution by searching with a population of candidates, which was inspired from the idea of Darwinian evolutionary theory. The main procedure of EC is iteration of evaluation, selection, combination and mutation. Ordinary EC employs a computational procedure to automatically calculate the fitness values for each candidate for selection. IEC is a modified version of EC by replacing the procedure by human-machine interface to obtain evaluation from the user as shown in an illustrative flowchart of Fig. 1. IEC is just like breeding process by artificial selection that have been done for thousands years to develop agricultural products and domesticated animals.

As similar to another method of problem solving, it becomes more difficult to find a feasible solution when the problem size gets larger in terms of the number of parameters characterizing solution candidates. The user might have to check too many useless alternatives to continue the interaction with the computer until he/she find a satisfactory solution. It is important to introduce any method to make the system to generate less number of useless candidates not only for efficient search but also reducing the user's fatigue.

This problem will be solved if we can restrict the search space including only the plausible candidates as

few as possible. One of the methods is to estimate the fitness function as a model of user's preference using techniques of artificial neural networks, as proposed in [4], [5]. These mechanisms seemed effective for some problems but we have obtained no sufficient result to show their effects yet. It is also a problem to process a large number of parameters for preference estimation.

Our method proposed here is based on an interactive decomposition of the target problem by dividing the genotype and protecting a part of genotype during breeding process as described below.

### III. DIVIDING GENOTYPE

It is a good method to divide a large problem into independent sub-problems to solve it faster. But it is often difficult because of dependency among parts of the problem features. Musical tune consists also of a complex information including melody, rhythm, tempo, timbre, and so on. Graphical title image is characterized by not so complex but well-structured features such as foreground color, background color, font, ornaments, and so on. We can easily divide the information into these functional parts, but evaluation of a product usually depends on combination of them. Combination of good melody and good timbre is not always good. Similar colors for foreground and background colors are not good, but it quite changes if some ornament, such as outline or shadow, is attached.

One of the advantages of EC comparing with the other optimization technique by search is that we can design different structure of search space than the structure of solution space using mapping between genotype and phenotype. Phenotype corresponds to the solution candidate and genotype corresponds to the search point. Musical tune can be divided into sections, parts, bars and so on. It is also helpful to build each of parts or sections independently and to combine them later. In addition, we can breed each of functional parts of music, rhythm, melody, timbre and so on independently by encoding them in separated parts of genotype.

Because of dependency among parts in terms of the effect for the quality of the solution, it is difficult to obtain good solution by optimizing each part sequentially. We often need to revise a part previously optimized during the search for another part. Even in the ordinary style of EC, it is still a research theme how to apply EC to multi-objective optimization problems. Fortunately in IEC, the user can control the evolutionary process by indicating which parts should be fixed and which parts should be modified.

We designed the structure of genotype for SBEAT as it consists of three types of chromosomes for rhythm, melody and velocity, as shown in Fig. 2. Velocity, that means loudness of the sound of note as a technical term in music, is an important factor for tune to sounds natural for human's ears. The individual in SBEAT population is a bar of sixteen beats and eight parts. Each of chromosomes is an array of sixteen by eight to include information for each possible note. Some loci in the melody and velocity chromosomes are ignored if the corresponding loci of rhythm chromosome indicates rest

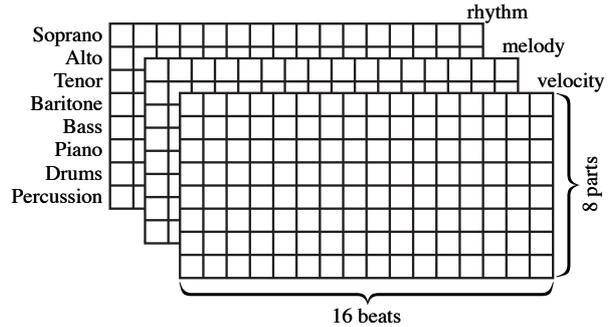


Fig. 2. Structure of genotype in SBEAT.

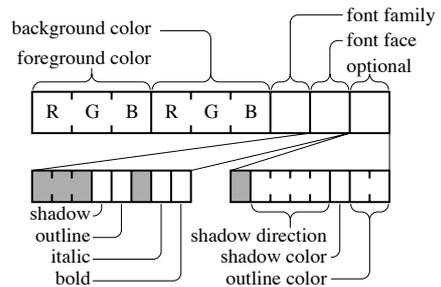


Fig. 3. Structure of genotype in SBTitle. A genome consists of nine bytes.

or continuation.

Fig. 3 shows the structure of genotype in another application named SBTitle for graphics design. It includes foreground color, background color, font family, font face, and optional information for outline and shadow. Gene of color consists of three bytes for red, green and blue factors. Gene of font family indicates the identity number available on the computer on which SBTitle is running. Gene of font face includes four effective bits to represent on or off of four types of features, bold, italic, outline and shadow. The optional part includes three kinds of information. The allele for outline color are black, white, intermediate color of foreground and background color, and its compliment color. The allele for shadow color are only black and white. The allele for shadow direction are center and eight orientations.

Each structure of genotype is divided into an appropriate number of parts by following the data structure. GUI for indicating each part should be designed to show the structure graphically to make it possible for the user to understand it intuitively as described later.

### IV. PROTECTION OF GENE

We designed a graphical user interface to make it possible for the user to indicate which part should be the object of genetic operation. In ordinary cases, all of features are the objects of mutation. So we introduced check buttons to indicate protection against mutation. Fig. 4 shows the dialog window of our system, SBEAT, which includes many radio buttons and check boxes. The radio buttons are for setting the correspondence be-

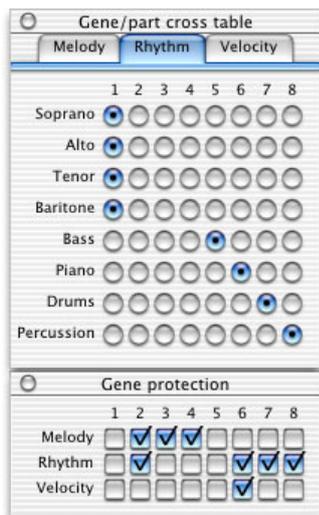


Fig. 4. Gene/part cross table and gene protection in SBEAT. Each row of circle buttons' array corresponds to the part, from soprano to percussion, and each column corresponds to the part of genotype, from #1 to #8. Tabs labeled melody, rhythm and velocity are to indicate which feature would be operated. Check boxes, eight by three round square buttons in the lower window, are to indicate protection of each part of genotype.



Fig. 5. Dialog window of SBTitle for gene protection.

tween musical parts and parts of genotype. The check box allocated for each part and feature, melody and rhythm, are for indicating protection.

It is possible for the user to indicate protection of arbitrary parts of genotype corresponds to functional or physical part of phenotype at any time he/she wants. To reduce the number of operations for indicating arbitrary number of check boxes at once, we implemented a method to choose a number of buttons by *press-drag-release* operation of mouse. The buttons in the rectangular area indicated by mouse operation act as being clicked. It makes easy to revise any parts again and again by breeding independently until any acceptable solution is found.

Fig. 5 shows the dialog window of SBTitle to indicate

which part of genome should be protected. This dialog has two group boxes named Font face and Shadow by which the user can indicate protection of a group of parts. All of sub-check boxes are marked to be checked when the user clicks the check box of the group box to check it. All of sub-check boxes turn to be unchecked when the user clicks the check box of the group box to erase the check mark. If some of sub-check boxes are checked and the others are unchecked, the group box's check box is marked mixed by “-” sign. It is no problem to design the arrangement in dialog window in such a style when the number of check boxes is not so many, but we need hierarchical style of dialog box design such as combination of disclosure triangle and scrolled list, for larger scale problems.

## V. EXAMPLE OF BREEDING WITH PARTIAL PROTECTION

Fig. 6 shows a field window of SBEAT just after producing mutant with protection on rhythm. You can see that all of individuals displayed on sub-windows consist quite same rhythm pattern for each part, but the pitches are slightly different among individuals. By protecting some parts, the user can seek acceptable score of rhythm section, main melody line, backing and so on independently. It is relatively easy to build a score like improvisational play of jazz music, fixing a basic melody line and rhythm section.

The newest version of SBEAT has capability to breed a tune including maximally 23 parts. It is almost impossible to steer the breeding process toward an acceptable solution without the function to protect partial gene combining with another control mechanism of switch on and off the sounds for each part. Partial breeding is a necessary method for this type of large scale problem domains.

Fig. 7 shows a field window of SBTitle filled with mutant of the upper left individual surrounded by red border, under protection of foreground color and shadow direction. Foreground color is blue and shadow direction is lower left for all of individuals, but the other features such as font are different among individuals in the population.

Fig. 8 shows a field window of SBTitle after three steps of operation, protecting gene of font family, filling it with mutant, then resetting all of the other parts of gene. Mutation provides relatively small changes because it would be useful for local search, but we sometimes want more drastic jump in some features. Combination of protection, mutation and random initialization is useful to try wider variety of candidates for a part of features. This figure demonstrates wide variety of alternatives of colors, font face and style of shadow for same font family.

## VI. FUTURE EXTENSION

Through our trial of these two types of application, some orientations of future extension of partial breeding could be considered as follows.

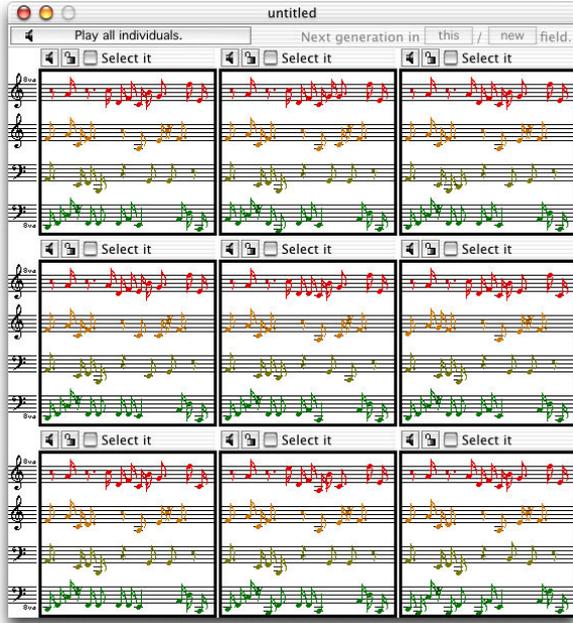


Fig. 6. A field window of SBEAT just after producing mutant under protection of rhythm.



Fig. 7. A field window of SBTittle just after producing mutant under protection of foreground color and shadow direction.

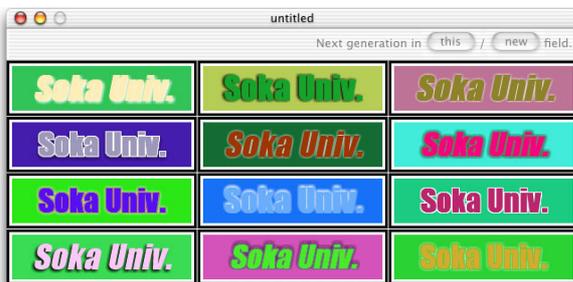


Fig. 8. A field window of SBTittle after protecting gene of font family, filling it with mutant, then resetting all of the other parts of gene. The user can examine a variety of alternatives under the fixed font family.

### A. Probabilistic protection

Partial protection can be seen as a special case of probabilistic protection where the user controls mutation rates independently for each part of genome. To protect a gene means to assign zero as probability of mutation. If the user assigns one to some gene, he/she can examine random alternatives for that feature.

### B. Restricting allele

Through the experience with SBTittle, the author felt it would be better to control the set of candidates of the value of loci, that is, some parts of genotype. Especially for font family, there are too many alternatives but they can be divided into a number of groups according to similarity. Some features should be encoded in well-organized gene so that it represents taxonomy of candidates. But sometimes it would be impossible because the criteria of appropriate taxonomy might be subjective and it will be clarified and changing during the breeding process.

These methods may bring more flexible process of breeding under user's control.

## VII. CONCLUSION

We proposed a method to solve a relatively large problem with IEC by dividing genotype to breed each part independently. To make it easy for the user to control this type of partial breeding, we designed a graphical user interface with check boxes for indicating which part of genotype should be protected from mutation. This method was invented through an application for music, but it will be useful for another domain where the solution candidates are complex but well-structured.

### Acknowledgments

The author thanks Manabu Senda for his co-operation to develop SBEAT version 2 and 3, and Nobuyuki Mizuno for his basic idea of SBTittle.

## REFERENCES

- [1] H. Takagi, "Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation," *Proceedings of the IEEE*, vol. 89, pp. 1275–1296, 2001.
- [2] T. Unemi and E. Nakada, "A Tool for Composing Short Music Pieces by Means of Breeding," *Proceedings of the IEEE Conference on Systems, Man and Cybernetics 2001*, Tucson, AZ, USA, pp. 3458–3463, 2001.
- [3] T. Unemi and M. Senda, "A New Musical Tool for Composition and Play Based on Simulated Breeding," A. Dorin (ed.), *Proceedings of Second Iteration*, Clayton, Victoria, Australia, pp. 100–109, 2001.
- [4] J. A. Biles, J. A., P. G. Anderson and L. W. Loggi, "Neural Network Fitness Functions for a Musical IGA," *IJA'96/SOCO'96. International ICSC Symposia on Intelligent Industrial Automation and Soft Computing*, B pp. 39–44, 1996.
- [5] M. Ohsaki and H. Takagi, "Improvement of Presenting Interface by Predicting the Evaluation Order to Reduce the Burden of Human Interactive EC Operators," *Proceedings of the IEEE Conference on Systems, Man and Cybernetics 1998*, San Diego, CA, USA, pp. 1284–1289, 1998.