Real-time Breeding Composition System by means of Genetic Programming and Breeding Procedure

Daichi Ando
Faculty of System Design, Tokyo Metropolitan University
dandou@sd.tmu.ac.jp

ABSTRACT

The use of laptop computers to produce real-time music and multimedia performances has increased significantly in recent years. In this paper, I propose a new method of generating club-style loop music in real time by means of interactive evolutionary computation (IEC). The method includes two features. The first is the concept of "breeding" without any consciousness of generation. The second is a multiple-ontogeny mechanism that generates several phenotypes from one genotype, incorporating ideas of co-evolution and multi-objective optimization. The proposed method overcomes certain limitations of IEC, namely the burden of interactive evaluation and the narrow search domain resulting from handling few individuals.

A performance system that generates club-style loop music from the photo album in mobile devices is implemented by means of the proposed method. This system is then tested, and the success of performances with the implemented system indicates that the proposed methods work effectively.

1. INTRODUCTION

The application of interactive evolutionary computation (IEC) as an aid to composition has been studied actively since the second half of the 1990s. However, the target of most of this research has been "off-line" composition. "On-line" composition, whereby the optimization process is performed in real time as music is generated, has received relatively little attention. The author believes that the application of IEC into on-line composition has not flourished because the target of most past research is classical music. Optimization procedures for these previous studies represent the simulation of classical composers' ideas, and it is natural that only the final output is performed.

On the other hand, streams of loops, similar to those programmed into synthesizers and samplers, have become increasingly popular since the 1990s. This loop-music category includes techniques common in minimal classical music. Representative music in these categories consists of the repetition of very short musical phrases. Today, loop music is synonymous with dance music for many listeners, and is commonly referred to as "club music."

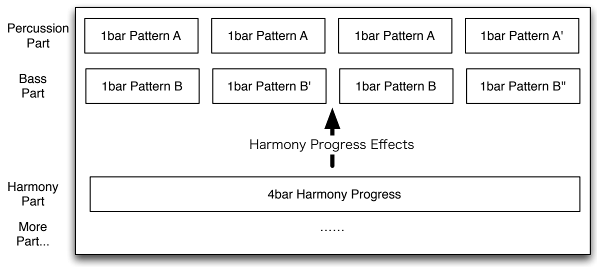

Figure 1. Typical Structure of Loop Club Music

A feature of club music is that short musical phrases become more and more transformed, and this stream of transformed musical phrases realizes the performance. The author expects that the IEC composition process itself is similar to the club music performance. Thus, a system that realizes a real-time club music performance from the IEC procedure has been developed.

1.1 Club-style Loop Music and Possibilities of Applying IEC

There are many types of club-style loop music. However, the typical style of this club music consists of a four-bar loop that gradually changes with repetition. There are no definite phrases or melodies.

Figure 1 shows an overview of the structure of four-bar loop music. The loop may consist of percussion, bass, melody, harmony, arpeggio, texture, and other sounds. The one part excludes harmony progress part and texture sound part, consists of 4 times repetition of 1 bar same phrase pattern and the variations (also effected all phrases change by harmony progress).

As mentioned before, we recognize that a loop-music performance consists of gradually changing and developing this group of four-bar loops without definite melodies.

Most real-time generated performances using "Track-maker," a composer and performer of club-style loop music, consist of many prepared phrases that are switched through the composition. However, Track-maker cannot prepare an infinite number of phrases for one performance. Hence, the performance and real-time generated music are almost fixed.

Let us now consider applying IEC to music phrase generation. In evolutionary computation, the first generation is generated randomly, and the reproduction operation is stochastic. Therefore, the author expects that applying IEC to club-style four-bar loop music will assist Track-maker in generating an infinite number of phrases. In addition, note that multi-point optimization with population convergence, such as in evolutionary computation, is suitable for generating musical variations [1]. The converged population includes similar individuals; thus, combining a music phrase generator with evolutionary computation provides many similar musical phrases. These are suitable for effecting gradually changing loop music in Track-maker.

1.2 Overview of Proposed System

The purpose of the proposed system is to generate real-time club-style four-bar loop music and effect parameters by means of the optimization process of IEC. The ideas of a "breeding" procedure and an IEC interface are developed, enabling performers to enjoy and create a good performance using IEC.

1.2.1 Co-Evolution and Multi-Objective Optimization to apply Multi-Parameter Control

In IEC, users should listen and add their score separately. Accordingly, the number of individuals making up the population should be small. Consequently, convergence is fast, and the optimization falls into a local minimum or maximum on the first generation. In addition, many parameters are required for music generation; nevertheless, IEC forces users to optimize with a small number of individuals. This indicates that IEC music generation imposes a large burden on users.

In the proposed procedure and interface, the author adopted the Genome Storage procedure for IEC [2]. The developed procedure includes co-evolution and multi-objective optimization. To be specific, the proposed procedure does not deal with concepts of "generation" used in normal evolutionary computation, but the procedure includes ideas of Simulated Breeding and Genome Storage. Furthermore, multiple ontogeny that generate many phenotypes from one chromosome have been adopted. This technique generates a huge number of parameters for music generation from a small number of individuals. Figure 2 shows an overview of the proposed procedure.

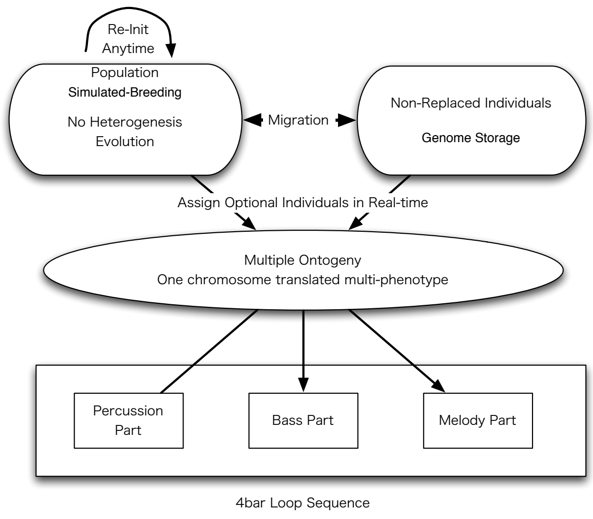

Figure 2. Proposed Method including Co-Evolution and Multi-Objective Optimization

2. OPTIMIZATION PROCEDURE AND GENE REPRESENTATION

2.1 No Heterogenesis by Simulated Breeding

The proposed breeding procedure reduces the user burden, allowing simple optimization. This procedure is based on Mutasynth [3], which includes the Genome Storage procedure. There is no complete heterogenesis. The user can always re-initialize the population and crossover with past populations.

Details of the proposed procedure are as follows:

1. Initialize population. Initial population is generated randomly.
2. Store favorite individuals in Genome Storage.
3. Select two optional favorite individuals as parents, then instigate reproduction. New children are generated, provided that the generated population does not replace individuals in Genome Storage.
4. Listen to the child population, then add favorite individuals to Genome Storage.
5. Select two individuals from the child population and Genome Storage, then instigate reproduction.
6. Re-initialize population. Select two parents from re-initialized population and existing individuals. A new mixed population will be generated. Store favorite individuals in Genome Storage.

2.2 Problem of Gene Representation in Past Research

The proposed system adopts genetic programming (GP) [4]. GP enables the representation of chromosomes as symbolic expressions (S-expressions).

An advantage of adopting GP for this compositional aid is that an S-expression tree state program can represent gradually changing musical repetition very simply. The tree state program representation has been adopted for many composition-aid systems, such as IRCAM Open Music, Common Lisp Music, AC Toolbox, David Cope's systems, and more. Many composers can easily understand the tree state program as music; hence, we can say that the tree state program is suitable for a composition system using IEC.

The GP representation has been used for MIDI information in past research. For example, Laine generated musical phrases by means of numerical expressions [5], and Johanson adopted function nodes to represent musical repetitions and chord terminal nodes [6]. Dahlstedt used a recursive developing tree to represent musical repetitions, and Putnam adopted GP representations to deal with synthesizer parameters [7].

Table 1. Function Node Set for Each Ontogeny

Numerical Calc Function	+, -, *, /, pow and more Each has two arguments
Sequential Exec Function	prog2(2), prog3(3),setParamA(1) and numerical calc function set

However, these past techniques are limited, as one chromosome represents only one phenotype. Their optimization efficiency is bad, as the application of IEC deals with only small-size populations.

2.3 Multiple Ontogeny

To solve the problem of the small population size in IEC, the proposed system adopts "multiple ontogeny" to allow each chromosome to make many phenotypes. This incorporates the ideas of co-evolution and multi-objective optimization.

Figure 3 gives an overview of the multiple ontogeny. In the first step, the chromosome consists of only numerical calculation function nodes. The ontogeny process replaces the numerical function nodes by another set of function nodes based on the node labels. On the left side of the figure, the chromosome tree consists of only numerical function nodes. These are not replaced, directory ontogeny is applied. The center of the figure shows an example of replacing sequential executable function nodes. Table 1 lists some examples of function node sets.

The chromosome program outputs one scalar value. This output is adopted as a synthesizer or musical phrase generation parameter. In Section 2.3.1, this scalar value is adopted as a synthesizer parameter, and then for musical phrase generation.

In contrast, the sequence of executable function nodes in the center of the figure are used for general evolutionary computation. The function node "progn" is placed as the top node, and the chromosome program executes sequentially. In this case, function nodes "setParamA," "setParamB" are included in the function node set. These function nodes set the values of *parameter A* and *parameter B*. This sequence of executable function nodes is able to set multiple parameters for the synthesizer, and a variable number of parameters can be controlled by the chromosome. Details of this function node set are discussed in Section 2.3.1.

The right side of the figure shows that the ontogeny generates phenotypes from the topology of the tree chromosome, rather than by evaluating the chromosome program. This ontogeny is described in Section 2.3.2.

2.3.1 Ontogeny for Synthesizer Parameter

The proposed method provides three ways of obtaining the synthesizer parameters from the chromosome program output. As mentioned in Section 2.3, all chromosomes are evaluated in these three ways.

The first method obtains one synthesizer parameter from the numerical calculation function set. In this case, one

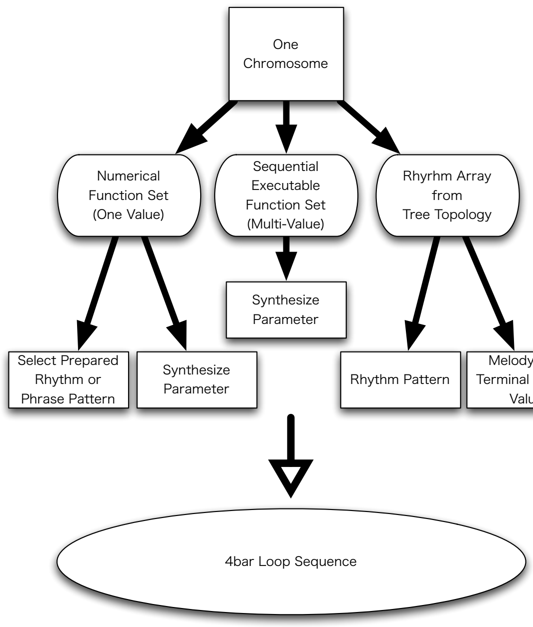
One Chromosome
Numerical Function Set (One Value)
Sequential Executable Function Set (Multi-Value)
Rhyrhm Array from Tree Topology
Synthesize Parameter
Select Prepared Rhythm or Phrase Pattern
Synthesize Parameter
Rhythm Pattern
Melody with Terminal Node's Value
4bar Loop Sequence

Figure 3. Multiple Ontogeny: Several Phenotypes Generated from a Chromosome

chromosome outputs one value. Thus, it is useful if the outputted value is an important synthesizer parameter.

The second method uses the sequence of executable function nodes to obtain many synthesizer parameters. A weak point of this method is that many parameters are generated randomly, and there are inevitably certain ones that cannot generate any sounds in the synthesizer. Consequently, generating all of the parameters is not suitable, and it is preferable to generate only important parameters.

The third method is applicable when the number of oscillators and parameters are fixed, generating an envelope time-line from the chromosome. In GP research, function node sets are frequently used to generate time-series values for problems of function regression. In particular, in the club-style loop-music scene, track-makers regard the envelope of the synthesized music to be important. Hence, this method works very effectively.

2.3.2 Ontogeny for Musical Phrases and Sequences

To generate musical phrases and sequences, there are currently two ontogeny-based approaches. The first generates one value by means of the numerical calculation function set, then selects a phrase from a previously prepared phrase set. The second approach generates musical rhythm patterns and phrases from the topology of the chromosome tree.

The first method is applied when specific phrases are predominant in a piece and genre, as fewer phrase pattern variations are needed.

The second method is a generative approach, and is applied to the main and sub-phrases if the time and scale are suitable and give no sense of incongruity.

Figure 4 illustrates the second ontogeny-based method. (1) is a chromosome consisting of only numerical calculation function sets. In (2), we remove the function labels

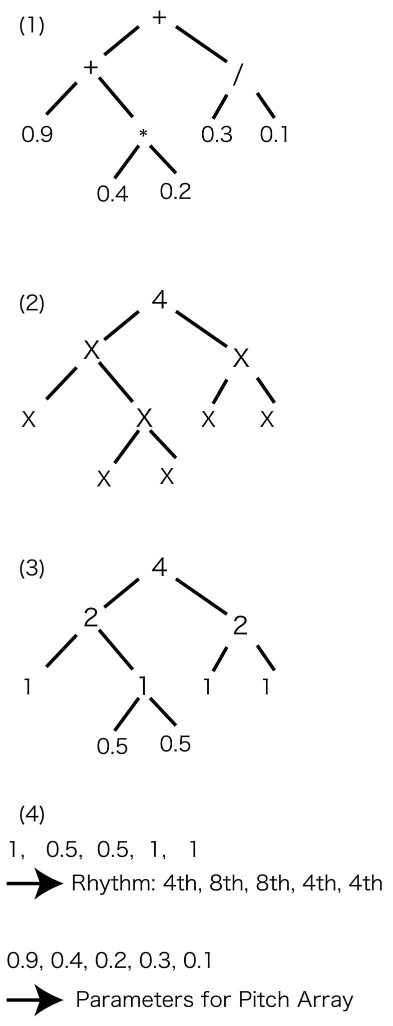

Figure 4. Ontogeny to Generate Musical Phrases and Rhythm Patterns from Chromosome Tree Topology

and terminal nodes. The time signature of the piece is set into the top node. In (3), if the parent node is a binary tree, half of the value of the parent node is given to the two child nodes. If the parent node is a triplet tree, we assign a third of the value of the parent node to each child node. Lastly, in (4), the generated rhythm pattern is fixed for the time signature of the piece. In the example in the figure, rhythm patterns of "Quarter," "Eighth," "Eighth," "Quarter," and "Quarter" are generated. The values of the terminal nodes are used for the pitch array.

3. IMPLEMENTATION: GENERATE CLUB-STYLE MUSIC FROM PHOTO ALBUM

3.1 Motivation

In recent years, mobile devices such as the iPhone and iPad have been utilized for photo storage. In addition, many applications that enable the performance of generative music have been developed for such devices.

The author believes that combining the demand for generative music and photo album storage in mobile devices will allow the generation of club-style loop music from within the stored photo album. Prototype software is developed and described.

In the past, many software applications and techniques have been proposed for the generation of music from photos. However, most past software enables only static and deterministic music to be generated by mapping from photos.

The proposed system incorporates a dynamic and stochastic music generator controlled by the performer. The proposed system provides multiple possibilities for real-time music generation.

To implement the proposed system, we use SuperCollider [1] to process the evolutionary computation, phrase generator, and real-time synthesis, and Processing [2] as the GUI. The sound and evolutionary computation engine and GUI communicate using OpenSound Control [3]. All implementation languages are free from executable platform. The sound synthesis and ontogeny parts running on SuperCollider are implemented independently from the system, so the performer can easily program such modules for their pieces and performances. In addition, perfect networking via distributed computation between each module ensures there is enough processing power for the performance.

In recent years, SuperCollider has become popular in the club-style music community. A number of club-style trackmakers can program their performance ideas, and easily include the proposed system.

3.2 Gene Representation and Applied Ontogeny

Figure 5 illustrates the process of generating a four-bar loop sequence from a photo.

A pair of photos is used for processing, with their pixel information used to generate a one-bar phrase. The pair of photos moves to the next bar. When four pairs of photos

[1] http://www.audiosynth.com/
[2] http://processing.org/
[3] http://opensoundcontrol.org/

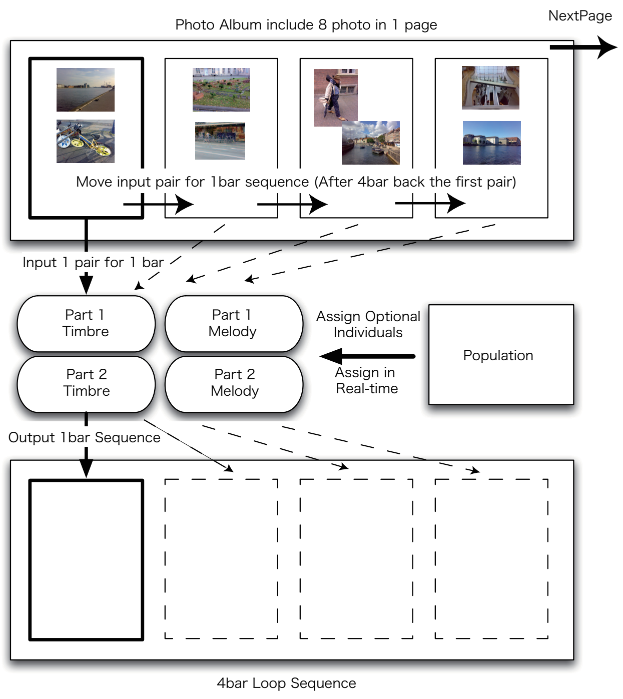

Figure 5. Generating a 4 Bar Loop Sequence from a Photo Album

are in the window, four-bar club-style phrases are generated.

Inputting pixel information and generating phrases is done in real time. The performer also assigns individuals in real-time. The assigned individuals are valid in the next bar.

The mapping for generating musical phrases from photos is as follows: (1) select prepared phrases, (2) generate rhythm patterns, (3) generate musical phrase.

As mentioned in Section 2.3.2, the first method is applied in (1), and the second and third methods are applied in (2) and (3). In (3), the generation of phrases is strongly involved in harmony progress, with notes selected from those in the harmony.

In Section 2.3.2, we saw that, if the basic rhythm pattern is not that of club-style music, the generated sound will not be recognized as club-style music. Hence, method (1) is used to generate bass drum and snare drum patterns. As decoration, rhythms such as tom drums and cymbals are generated by method (2).

Table 2 lists the adopted mappings.

3.3 Parameters of Genetic Programming

As mentioned before, pixel information such as color elements, frequency, and so on are adopted as the terminal nodes of GP. Table 3 lists the terminal node set.

3.4 User Interface

Figure 6 shows the user interface. In area (1), individuals are assigned to generate each musical part. Area (2) assigns the selected parents. After assigning two individuals in this area, the reproduction button becomes active, and the generated children appear in area (3). Area (4) denotes

Table 2. Mappings for Music Generation from Photo Album

Bass Drum	Rhythm-Pattern Timbre	Output value from numerical function set. Fixed.
Snare Drum	Rhythm-Pattern Timbre	Output value from numerical function set to select prepared rhythm pattern. Output value from numerical function set to synthesize.
Tom Drum	Rhythm-Pattern Timbre	Generate from Tree. Output value from numerical function set to synthesize.
Cymbal	Rhythm-Pattern Timbre	Generate from Tree. Output value from numerical function set to synthesize.
Harmony	Harmony Timbre	Four outputs of numerical function set are applied to each note of a four-note chord. Four values from numerical function set to four parameters of synthesizer.
Main Melody 1 and 2	Phrase-Generation Timbre	Generate from Tree. Four values from numerical function set to four parameters of synthesizer.
Arpeggio	Pitch-Select Pattern Timbre	Value from numerical function to select from harmony notes. Output value from numerical function set to select prepared pattern. Value from numerical function set to synthesize.
Bass	Phrase-Generation Timbre	Generate from Tree. Value from numerical function set to use timbre.
External Effects	Synthesize	Values from numerical function set to Mic. Input for duo performance.
Internal Effects	Synthesize	Values from numerical function set to SuperCollider sounds.
Texture Sound	Synthesize	Values from numerical function set to generate texture sounds.

Table 3. GP Terminal Nodes used for Music Generation from Photo Album. Subscript "a" denotes applied to both photos in a pair.

$R\%_a$	Percentage of R elements of Photo (G and B are also used)
$Xsize_a$	X size of Photo
$Ysize_a$	Y size of Photo
SimilCol	Similarity of two photos
$Frequency_a$	Frequency of photo

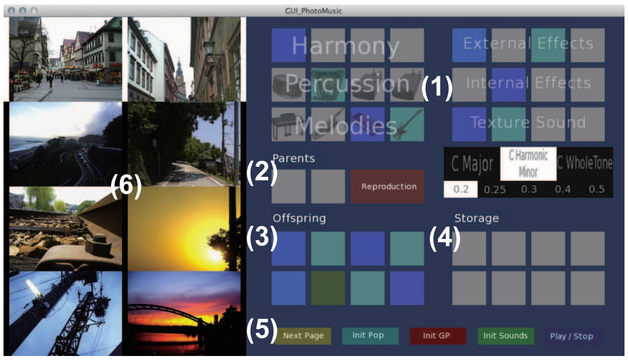

Figure 6. User Interface for Music Generation from Photo Album

the Genome Storage, used for the "breeding" operation. The control buttons are in area (5).

Assignment area (1) consists of 17 parts. Details of these parts are listed in Table 2. Instrument icons are shown in their respective phrase generation areas.

Area (6) displays four pairs of photos. The top pair is surrounded by a white border, denoting that this pair are generating the current bar. Pressing the "Next Page" button displayed in the left of (5) turns the photo album pages, thus displaying the next eight photos.

Areas displayed in gray do not have individuals assigned, such as (1), (2), and (4). Individuals are colored by chromosome features.

Colored individual icons can be moved using the drag-and-drop operation.

3.5 Jam Session: Performers All Bring Photos

This implemented prototype allows multi-user operation when multiple performers bring photos. Using OpenSound Control, the performers share the sound engine, and the GUI and photo data are displayed and executed in each performer's computer. This mechanism enables jam-session-style performances.

Furthermore, in recent years, generative music has become more popular in the club-style music scene. Visual elements are increasingly important in the club scene. Using the proposed system, performers can realize generative music with visual elements.

4. CONCLUSION

This paper presented a new club-style music generator. The proposed approach generates music from photo albums by means of Genetic Programming and a new "breeding" procedure that does not involve heterogenesis. The proposed software and its GUI have been implemented and described.

Acknowledgments

At the end of the Conclusions, acknowledgements to people, projects, funding agencies, etc. can be included after the second-level heading "Acknowledgments" (with no numbering).

5. REFERENCES

[1] A. R. Burton and T. Vladimirova, "Generation of musical sequences with genetic techniques," *Computer Music Journal*, vol. 24, no. 4, pp. 59–73, 1999.

[2] D. Ando, P. Dahlsted, G. Nordahl, and H. Iba, "Computer aided composition by means of interactive gp," in *Proceedings of International Computer Music Conference 2006, New Orleans, USA*. ICMA, October 2006, pp. 254–257.

[3] P. Dahlstedt, "A mutasynth in parameter space: interactive composition through evolution," *Organized Sound*, vol. 6, pp. 121–124, 2004.

[4] D. E. Goldberg, *Genetic Programming: On the Programming of Computer by Means of Natural Selection*. MIT Press, 1992.

[5] P. Laine and M. Kuuskankare, "Genetic algorithms in musical style oriented generation," in *Proceedings of First IEEE Conference on Evolutionary Computation*. Washington D.C.: IEEE, 1994, pp. 858–861.

[6] B. E. Johanson and R. Poli, "Gp-music: An interactive genetic programming system for music generation with automated fitness raters," School of Computer Science, The University of Birmingham, Tech. Rep. CSRP-98-13, 98.

[7] J. B. Putnam, "Genetic programming of music," 1994, unpublished manuscript. Socorro, NM: New Mexico Institute of Mining and Technology.