# Evolutionary Interactive Music Composition

Tao-yang Fu
Dept. of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
tyfu@cs.nctu.edu.tw

Tsu-yu Wu
Dept. of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
tywu@cs.nctu.edu.tw

Chin-te Chen
Dept. of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
chinte@cs.nctu.edu.tw

Kai-chu Wu
Dept. of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
kcwu@cs.nctu.edu.tw

Ying-ping Chen
Dept. of Computer Science
National Chiao Tung University
HsinChu City 300, Taiwan
ypchen@cs.nctu.edu.tw

## ABSTRACT

This paper proposes the CFE framework—Composition, Feedback, and Evolution—and presents an interactive music composition system. The system composes short, manageable pieces of music by interacting with users. The most important features of the system include creating customized music according to the user preference and providing the facilities specifically designed for producing large amounts of music. We present the structure as well as the implementation of the system and the auxiliary functionalities that enhance the system. We also introduce the auto-feedback test with which we verify and evaluate the interactive music composition system.

## Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Methodologies and techniques*

## General Terms

Design

## Keywords

Evolutionary computation, real-world application, interactive music composition, auto-feedback test

## 1. INTRODUCTION

Music plays an important role in our daily life. The definition of pleasant music is different from people to people. We can easily observe that almost everyone puts different ringtones on their cellular phones. As a result, customization for pleasant music is desirable for our modern life. Although there are a lot of computer software which can help people compose music, it is still hard, if not impossible, to create pleasant music for unskilled people. Hence, we are trying to make the computer automatically create music for us instead of merely letting us put notes into tracks.

In order to reach the goal, making the computer automatically create music, we develop a system which creates music by interacting with users. We design the system according to two backgrounds. One is Evolutionary Computation [1, 2, 3]. The kernel mechanism is built based on the concept of Evolutionary Computation. The other background is the MIDI format. When we create music in the MIDI format, we can guarantee that the created music can be played on almost any computers, cellular phones, or other customizable electronic devices.

## 2. THE CFE FRAMEWORK

The evolutionary algorithms may be an appropriate technique to optimize music if we take the user preference as the fitness. However, what if the user needs more than one song? Similar jobs have to be done over and over to produce more songs which the user favors. To avoid this awkward situation and to achieve a better music composition system, we propose the CFE framework in this study.

The CFE framework contains three major parts, Composition, Feedback, and Evolution. In this framework, the individuals in the evolutionary environment is not complete songs but some musical elements or guidelines. The Composition part uses these musical elements and guidelines to construct new melodies. The composed melodies then wait for the user's response such as making a grade. After the system receives the information, the Feedback part distributes these feedbacks among the musical elements and guidelines to evaluate how fit these composing components are. For discovering better elements, the technique of evolutionary computation is adopted such that new elements and guidelines are born into the population.

The three parts can be done independently. Therefore, once the user is satisfied with the composed music, no more work is necessary when he or she needs more pieces of music because Composition can be conducted alone. Since Composition and Evolution are isolated, for making use of the domain knowledge, such as the constraints, indication, and implications in the music theory, it is easier to embed such knowledge into the Composition part than to interfere with the regular operations of evolutionary algorithms. The separation of Feedback and Evolution provides the feature that

*GECCO'06,* July 8–12, 2006, Seattle, Washington, USA.
ACM 1-59593-186-4/06/0007.

the pace of evolution can be determined by how many feedbacks we get from the user.

We implement a reference system based on the proposed CFE framework to automatically compose music. We design a hierarchical architecture which contains three levels of components: notes, music blocks, and music phrases. Higher level components consist of lower level ones. The note is at the bottom level of the hierarchy and is the fundamental element. It consists of three music factors: the pitch, the tempo, and the intensity. The music block is at the middle level of the hierarchy. It has an array consisting of variable-length small sequences of notes and records the block information, such as the block fitness.

The music phrase is the theme music with a specific length, say, 8 measures to 16 measures at the highest level of the hierarchical architecture. As the relationship between notes and music blocks, a music phrase has an array consisting of music blocks and records the music phrase information, such as phrase grades. The system include some two-dimensional arrays, which records the fitness of each pair of the note relation and of the music block relation. These tables help Composition pick the proper music blocks and fill in the incomplete music phrases.

The Feedback part provide the interface for users to make their responses to the system. By grading the music phrase, users express their satisfactory degrees and train the evolutionary environment. Once the grading is made, the score is distributed to all the music blocks contained in that phrase. Thus, the fitness value of a music block is determined by the average grade of all the music phrases in which the particular music block occurs. The Evolution part, seeking for the fittest music blocks, plays an essential role in the music composition system. We employ an evolutionary algorithm similar to a typical genetic algorithm.

Although the flow path can function properly, we still need to enhance the system for two reasons. First, we should make the grading runs as few as possible. Moreover, we would like to improve the music composition. In order to reduce the grading runs, for each grading event, our system can change the number of evolution rounds according to the diversity of the new scores. In order to improve the music composition, we add basics elements of the music theory into the system. We set the pre-defined fitness into the fitness tables and expect the default fitness table to help compose not-too-bad music at the early stage. And as the real world music, a sequence of notes repeating in the whole song often occurs, such as “Happy Birthday”. We make the Composition part to implement this feature.

## 3. AUTO-FEEDBACK TEST

We need an efficient and objective mechanism that helps us to verify our system for two reasons. Firstly, making a lot of tests by using manpower is not efficient. Secondly, people’s feelings for the same music may change due to some irresistible causes, such as the mood at the time. We design the Auto-Feedback test, which can automatically interact with our system, simulates a user’s preference to grade the music phrases according to some factors of music and provides the testing statistics to verify our system.

The Auto-Feedback test can grade a music phrase on four aspects, rhythm, specific pitch, specific pitch sequence, and specific pitch interval. We can set up multiple rules regarding the four aspects to simulate user’s preference. The emiprical results of the Auto-Feedback test can help us to verify the utility of the proposed functionalities.

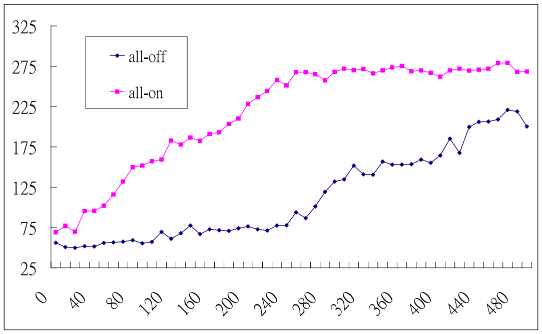

**Figure 1: All of the enhancement functions for the system are switched on vs. all are switched off.**

Figure 1 shows the results of testing the utility of the proposed enhancing mechanisms. In Figure 1, the X-axis is the feedback times and the Y-axis is the grade. The **all-on** means all of the enhancing modules we design to enhance our system are in use. The grades are growing higher than those for **all-off**, which means none of the modules are used. It shows our system can evolve according to the preference set by Auto-Feedback.

## 4. CONCLUSIONS AND FUTURE WORK

We show our system is feasible and promising to automatically compose customized music. Although the system currently acts only on short music, the design should be extensible for longer music.

For the future work, more measurements from the real users are needed because it is difficult to model the user’s preference with certain rules in the Auto-Feedback test. For practical use, we will provide some post-composition functionalities for the user, such as modules for automatically generating harmonies to match the music phrase and for editing music. After all these procedures are done, different types of music file can be generated and used in many applications, such as cell phone rings.

## 5. ACKNOWLEDGMENTS

The work was partially sponsored by the National Science Council of Taiwan under grant NSC-94-2213-E-009-120. The authors are also grateful to the National Center for High-performance Computing for computer time and facilities.

## 6. REFERENCES

[1] H.-C. Chang. Applying genetic algorithms to music composition: Modeling and simulation of art systems, 2002. Master’s Thesis, National Chiao Tung University, Hsinchu, Taiwan.

[2] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. SpringerVerlag, 2003. ISBN: 3540401849.

[3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, New York, 1989.