# Music Pattern Mining for Chromosome Representation in Evolutionary Composition

Chien-Hung Liu and Chuan-Kang Ting
Department of Computer Science and Information Engineering and
Advanced Institute of Manufacturing with High-tech Innovations
National Chung Cheng University
Chia-Yi 621, Taiwan
Email: {lch101p, ckting}@cs.ccu.edu.tw

***Abstract*—Artificial intelligence (AI) has bloomed in many novel fields such as computational creativity. Recently, research on automatic composition using AI technology, especially evolutionary algorithms, has received considerable promising results. Traditionally, chromosomes are represented as a series of numbers to indicate the notes for evolutionary composition. This study attempts to explore the composition styles by mining music patterns of a specific composer. The patterns are used as genes for chromosome representation. Accordingly, the composition styles are considered in generating music by evolutionary algorithms. The fitness function is based on music theory to smooth the progression between phrases. Experimental results show that the patterns mined from compositions can reflect the composer's style and benefit generating satisfactory songs by evolutionary algorithms.**

***Index Terms*—evolutionary computation; genetic algorithm; pattern mining; automatic composition; creative intelligence**

## I. Introduction

Computational intelligence and artificial intelligence technologies have been widely applied to many areas in creativity, such as music, art, literature, and engineering design. Music composition using artificial intelligence attracts much attention since music plays an important role in human life. However, composing music is particularly difficult because many elements need to be considered, to wit, rhythm, melody, texture, musical form, tone color and tonality. In addition, compositions contain explicit or implicit characters and styles of composers. To mimic a composer's work is another thorny issue to be considered.

Evolutionary algorithms have shown their effectiveness in various search and optimization problems. Recently, evolutionary algorithms are utilized to compose music and have received several promising results. For example, genetic algorithm (GA) is commonly used in the music composition and accompaniment systems. The evolutionary composition systems ordinarily represent a chromosome by a series of numbers to indicate the notes in a composition. The musical characters and styles, nonetheless, are not considered in the representation and thus require subsequent fitness evaluation or genetic operators to accomplish it. Furthermore, most of the evolutionary composition systems evaluate the chromosomes (compositions) based on interaction with humans. Although human feedback is useful for evaluation, it suffers from fatigue and decreased sensitivity after long-time listening, which makes fitness evaluation very exhaustive.

This study aims to address the above issues. First, we propose using pattern mining to extract the musical style of a composer. The patterns reflect the favored melodic combinations and can reduce the search space for evolutionary composition. In this study, we mine the patterns from the compositions of Jay Chou, a very popular male singer and composer in Asia. His prolificacy in composition also helps in pattern mining. The resultant patterns are used to represent chromosomes for evolutionary composition. Second, this study adopts music theory as the basis of fitness evaluation to address the fatigue issue at interactive evolutionary composition systems. The new compositions are evolved from existing compositions. The generated melody is further applied with musical form and accompaniment for increasing euphony.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III introduces the sequential pattern mining for musical patterns. Section IV describes the proposed evolutionary composition system. The experimental results are presented in Section V. Finally, Section VI gives the conclusions of this study.

## II. Related Work

### A. Evolutionary Composition

Evolutionary computation is widely used in computational composition in the light of its recognized capability in global search and optimization. McIntyre [1] first applied GA to generate four-part Baroque harmony. Laine and Kuuskankare [2] adopted GA to find the music functions and estimate their parameters for generation of music. The music functions serve as a logical and general music expression. Pazos et al. [3] used GA with the interaction of several musicians to build a model for creating rhythmic patterns. Further, Marques et al. [4] generated and weighted the composition rules as a basis which can distinguish good and bad music. Towsey et al. [5] analyze the features of good songs. Through the analysis, they divided the songs into five categories and utilized these features to evolve music by GA. In addition, Schoenberger [6] analyzed works of many famous composers like Bach based on Western tonal theory. Khalifa et al. [7] proposed adopting

978-1-4799-7492-4/15/$31.00 

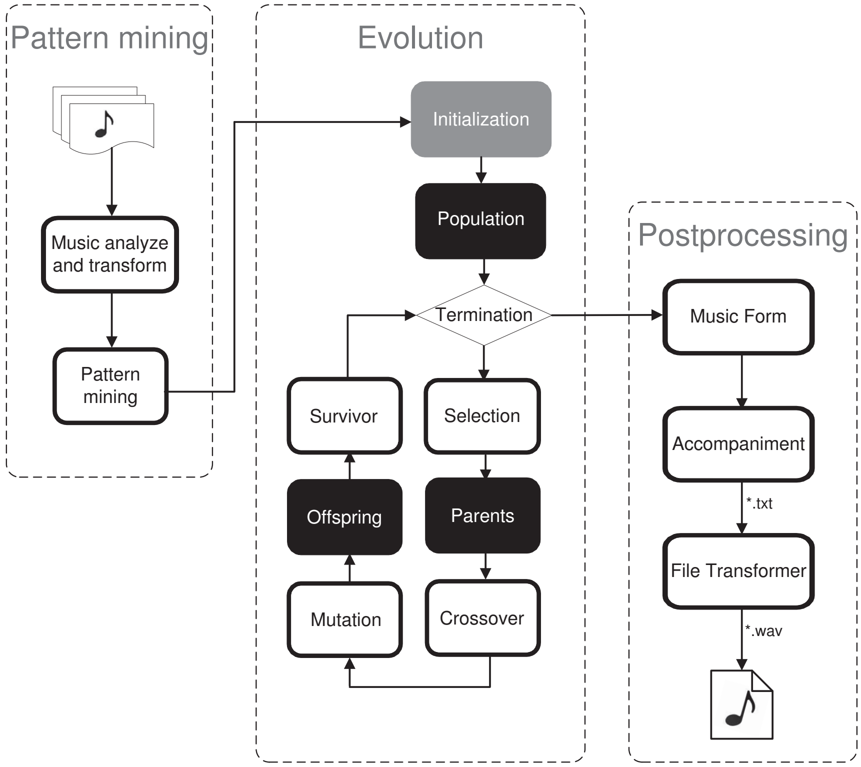


Figure 1. Flowchart of the evolutionary composition system

four motifs in composing music. These motifs are evaluated according to some grammar rules. Chen et al. [8], [9] presented the CFE framework considering feedback as a key element in music composition. Liu and Ting [10] proposed using music theory and charts to design an objective evaluation criterion. The fitness rules are based on music theory and the weights of rules are determined through music charts. Wu et al. [11] proposed a novel GA which keeps the rhythm, chords and structure of phrases from existing music. The method imitates the progression of melody at composing music.

Some studies focus on automatic accompaniment. In addition to melody, accompaniment plays an important role in music. Good accompaniment can effectively strengthen harmony, tighten the structure of tunes, and reinforce the expression of music. Luo et al. [12] developed a real-time accompaniment system for the sung voice. The system detects the pitches of the sung voice and accordingly creates the score from mixture of sung voice and accompaniment. Experimental results show that this system is robust against noise and can provide good accompaniment for the sung voice. Chen et al. [13] designed a tempo-based accompaniment through analysis of the tempo and an interactive system. Additionally, Jo et al. [14] established a chord-based music composition system using an auto-accompaniment program to compose music for non-musicians. Simon et al. [15] trained a Hidden Markov Model with a music database and adopted this model to automatically choose chords for a vocal melody. The naive users can create a song by singing into a microphone. They can also experiment with different styles and chord patterns without music knowledge. Liu and Ting [16] proposed using music theory to generate polyphonic accompaniments which can coordinate the melody.

### B. Pattern Mining

Some studies adopt pattern mining to compose music or to classify music styles. Giovanni et al. [17] used a hybrid strategy which combines fuzzy control and data mining to solve the figured bass problem. The strategy is capable of finding harmonious musical solutions. Kuo and Shan [18] developed a music filtering system which learns the user preference by mining the melody patterns from user's behavior of accessing music. According to the preference, the system can recommend suitable music to the user. Aniruddha and Vahida [19] focused on the mood of Indian popular music. They built a system to recognize the mood of music by analyzing and mining spectral and temporal audio features. Chiu and Shan [20] employed data mining techniques to analyze and discover the common patterns or characteristics of music structure, melody style and motif from the given music. The three characteristics are used for generating new music. Ren et al. [21] developed a song tokenization method, which transforms a composition into a sequence of units for music genre classification. Ren and Jang [22] further utilized time-constrained sequential patterns for music genre classification. Christian and Daniel [23] used data mining to classify the music genres of MIDI files. Chiu et al. [24] proposed apriori-based polyphonic repeating pattern discovery (A-PRPD) and tree-based polyphonic repeating pattern discovery (T-PRPD) to discover music patterns. The experiments show that both algorithms are effective for mining polyphonic repeating pat-

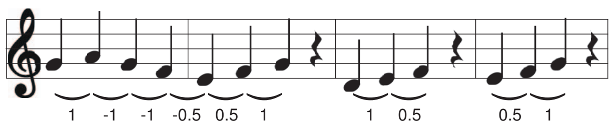


Figure 2. Example melody

terns from synthetic music data and real data. Shan et al. [25] proposed classifying music styles through extracting the chords from existing music and using them to represent the features of melody.

## III. Mining Patterns from Compositions

This study proposes investigating a composer's style through pattern mining. In this study, we focus on the compositions of Jay Chou due to his prolificacy and enormous popularity in Asia. In the past twelve years, he published one album and achieved at least a top-one song in the chart every year. He is also named as the most popular and significant male singer in the Greater China region.

The proposed evolutionary composition system consists of three stages: pattern mining, evolutionary composition, and post-processing. As Fig. 1 illustrates, the system first analyzes the compositions of Jay Chou. A melody is separated and converted into several sequences of intervals according to music phrases. These interval sequences render the data for sequential pattern mining. The patterns mined from the interval sequences are used as chromosome representation for the proposed GA. With the pattern representation, the evolutionary process of GA is conducted to evolve the candidate compositions. The best composition obtained from GA is further processed with musical form and accompaniment to enhance its structure and euphony. More details about the proposed evolutionary composition system are given below.

### *A. Music Data Representation*

This study adopts sixty compositions of Jay Chou. Considering the fact that Pop music usually repeats some parts, we discard the repeated parts to ensure that each musical phrase will not be overemphasized in the music data for pattern mining. Table I lists the information of Jay Chou's compositions to be mined. As aforementioned, each composition is divided into several sequences according to its musical phrases, where a sequence is composed of intervals of the melody and the interval is calculated by the distance of adjacent notes. For example, a piece of music in Fig. 2 is represented by three sequences of intervals: {1, -1, -1, -0.5, 0.5, 1}, {1, 0.5}, and {0.5, 1}. The 60 compositions of Jay Chou are separated into 1108 interval sequences. The data representation based on intervals gains the advantage that the sequences are not affected by the difference in scale and key between compositions.

### *B. Sequential Pattern Mining*

Pattern mining is an important topic in data mining. The Apriori algorithm is widely used to mine association rules for

Table I
Information of music data

| Title of song | #Phrases |
|---|---|
| All the Way North | 19 |
| Shanghai 1943 | 20 |
| Secret | 19 |
| Rainy Mood | 20 |
| Coral Sea | 17 |
| Wounds of War | 10 |
| Cowboy On the Run | 21 |
| In the Name of Father | 21 |
| Peninsula Ironbox | 15 |
| Adorable Lady | 16 |
| White Windmill | 19 |
| Back to the Past | 28 |
| Her Eyelashes | 38 |
| Silence | 24 |
| Little Blacksmith in Milan | 18 |
| You Hear Me | 23 |
| Perfectionism | 19 |
| Not Good Enough for You | 14 |
| Territory | 11 |
| Ye Qu | 13 |
| Chapter Seven | 4 |
| Dong Feng Po | 17 |
| Dad, I'm home | 9 |
| Rain All Night | 22 |
| Blue and White Porcelain | 16 |
| William Castle | 19 |
| Starry Mood | 28 |
| Drifting Poet | 23 |
| Orbit | 16 |
| Romantic Cellphone | 20 |
| Rosemary | 30 |
| A Step Back | 18 |
| Rainbow | 20 |
| The Last Battle | 12 |
| Sweet | 20 |
| A Song-length of Time | 21 |
| Chrysanthemum Terrace | 24 |
| Hard to Say it Out | 18 |
| Sunshine Nerd | 19 |
| The Longest Movie | 20 |
| Love before BC | 13 |
| Maple Leaf | 20 |
| Grandpa's Tea | 4 |
| A Dandelion's Promise | 16 |
| Fade Away | 13 |
| Rice Field | 16 |
| The Promised Love | 20 |
| Black Sweater | 16 |
| Fun Fair | 19 |
| Secret Sign | 21 |
| Simple Love | 23 |
| Excuse | 23 |
| Basketball Match | 22 |
| Ancient Indian Turtledove | 24 |
| Hair Like Snow | 17 |
| Fearless | 9 |
| Tornado | 22 |
| Step Aside | 13 |
| Qi Li Xiang | 16 |
| Brokem String | 20 |

Table II
PATTERNS OBTAINED FROM MINING

| Music pattern ($k = 1$) | Support value | Music pattern ($k = 2$) | Support value | Music pattern ($k = 3$) | Support value |
|---|---|---|---|---|---|
| {0} | 0.84 | {0, 0} | 0.64 | {0, 0, -0.5} | 0.19 |
| {0.5} | 0.34 | {0, 0.5} | 0.19 | {0, 0, -1} | 0.4 |
| {-0,5} | 0.42 | {0, -1} | 0.57 | {0, 0, 1} | 0.29 |
| {1} | 0.70 | {-0.5, 0} | 0.22 | {-0.5, -0.5, -1} | 0.04 |
| {-1} | 0.78 | {-0.5, -0.5} | 0.09 | {-0.5, 2, 0} | 0.02 |
| {2} | 0.15 | {-0.5, 1} | 0.17 | {-0.5, 1, -1} | 0.08 |
| {-2} | 0.16 | {1, 0} | 0.42 | {2, -1, 0} | 0.06 |
| {2.5} | 0.22 | {1, 2} | 0.05 | {2, 1.5, 0} | 0.01 |
| {-2.5} | 0.13 | {1, -3.5} | 0.04 | {2, -2, 0} | 0.02 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

| Music pattern ($k = 4$) | Support value | Music pattern ($k = 5$) | Support value | Music pattern ($k = 6$) | Support value |
|---|---|---|---|---|---|
| {0, 0, 0, -0.1} | 0.24 | {0, 0, 0, -0.5, -1} | 0.06 | {0, 0, 0, 0, 0, -0.5} | 0.03 |
| {0, 0, -1, 0} | 0.23 | {0, 0, -1, 0, -0.5} | 0.04 | {0, 0, 0, 0, 0, -1} | 0.08 |
| {0, 0, 2, 1} | 0.01 | {0, 0, 1, -1, -1} | 0.07 | {0, 0, 0, 0, 0, 1} | 0.06 |
| {1.5, 0, 0, -1.5} | 0.04 | {1, 1, -1, -1, 0} | 0.04 | {-0.5, 0, 0, 0, 0, 0} | 0.01 |
| {1.5, 0, -1, -1.5} | 0.02 | {1, 1, -1, -1.5, 0} | 0.01 | {-0.5, 0, 0, -1, 0, 0} | 0.01 |
| {1.5, 0, 1.5, 1.5} | 0.02 | {1, 1, 1.5, 0, 0} | 0.02 | {-0.5, 0, -1, 0, -1, 0} | 0.01 |
| {2.5, 0, 0, -1} | 0.05 | {1.5, 0, 0, 0, -1} | 0.03 | {-1, 0, 0, 0, -1, 0} | 0.04 |
| {2.5, -0.5, 0, -1} | 0.02 | {1.5, 0, 0, -1.5, 1.5} | 0.01 | {-1, 0, 0, 0, -1, 1} | 0.02 |
| {2.5, -0.5, -1, -0.5} | 0.01 | {1.5, 0, -0.5, -1, -1} | 0.01 | {-1, 0, 0, 0, 1, 0} | 0.01 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

item sets. However, this algorithm cannot directly apply to music pattern mining for two reasons: 1) the same interval (item) may appear many times in one sequence; 2) the adjacency of intervals is important because a jumping interval is inappropriate to present the composition character.

This study develops the revised Apriori algorithm for music pattern mining. The algorithm consists of two phases:

1) L-itemset Phase: This phase generates the large segments with respect to the minimum support. For instance, a minimum support 1% requires a segment to appear at least 12 times among the 1108 phrases. Note that the items are repeatable in the music data.
2) Sequence Phase: This phase involves two steps, i.e., join and pruning, to derive the large $k$-segments based on large $(k-1)$-segments with $k \geq 2$. The join step generates all the possible candidate segments and the pruning step then eliminates the unqualified candidates of low support values.

Table III gives an example of mining five music sequences. Assume the minimum support proportion is 0.4. The L-itemset phrase first finds the satisfied sets: {0}, {0.5}, {1}, {2}, {-0.5}, {-1}, {-2}. Afterward, the joint step of sequence phrase explores all possible candidates: {0, 1}, {1, 0.5}, {1, -1}, {1, 2}, {2, 0}, {2, 1}, {-0.5, 1}, {-0.5, -1}, {-1, 1}, {-1, -0.5}, {-1, -2}, {-2, 0}. Then the pruning step eliminates {1, 0.5}, {-0.5, 1}, {-0.5, -1}, {-1, 1}, {-1, -0.5} since the support of these candidates 0.2 is lower than the minimum support 0.4. This process continues until no further sequence can be found.

From the compositions of Jay Chou, the revised Apriori algorithm results in 5074 patterns in total. Table IV presents the numbers of patterns $k_1$ to $k_6$. Due to space limitation, Table II lists only some patterns obtained. Note that the elements of patterns represent the intervals of melody rather than the absolute pitches.

Table III
EXAMPLE SEQUENCES FOR PATTERN MINING

| No. | Music sequence |
|---|---|
| 1 | 1, 0.5, 2, 0, 1, -1, 1 |
| 2 | 2, 1, 2, 2, 0, 1, 0.5 |
| 3 | 1, 2, 1, 0.5 |
| 4 | 1, -1, -0.5, 1, |
| 5 | -0.5, -1, -2, 0 |

Table IV
NUMBERS OF PATTERNS OBTAINED FROM MINING

| Pattern length | Pattern number |
|---|---|
| 1 | 20 |
| 2 | 176 |
| 3 | 764 |
| 4 | 1461 |
| 5 | 1616 |
| 6 | 1037 |

## IV. EVOLUTIONARY COMPOSITION

The proposed evolutionary composition system is based on GA with the pattern mining results. Specifically, the GA uses the patterns in chromosome representation and utilizes the evolutionary process to evolve the candidate compositions that follow the rhythm and musical form from a given composition. The design of GA involves chromosome representation, fitness function, and genetic operators. The following subsections describe these designs.

### A. Representation

In the proposed GA, a chromosome is represented by a series of numbers to indicate the patterns for a composition.

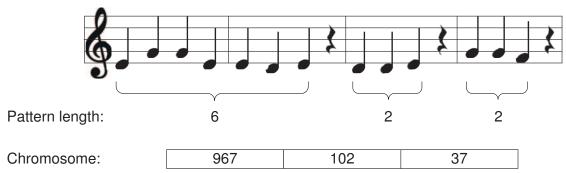


Figure 3. Example of chromosome representation

Given sample music, the system first analyzes its composition and lengths of phrases. The GA aims to find suitable patterns corresponding to the phrase lengths. As Fig. 3 illustrates, the first phrase contains six intervals; therefore, its corresponding gene value ranges from 1 to 1037 since the number of patterns of length 6 is 1037 (cf. Table IV). The second and third phrases contain two intervals and thus have gene values 1–176. For converting a sequence of intervals into a composition, the first note is determined randomly at initialization and the remainder notes can then be derived according to the intervals. For example, the third gene in Fig. 3 indicates the 37th pattern, i.e., {2.5, 1}. Suppose the first note of this phrase is randomly determined to be C, the subsequent two notes (F and G) are derived from the intervals 2.5 and 1, respectively.

### B. Fitness Function

A key issue at the use of GA for music composition is the design of fitness function, which guides the search direction of GA. This study presents a fitness function that focuses on the connection between phrases and the consonance with chords. Especially, we utilize music theory as the basis for the fitness evaluation rules. Table V lists the evaluation rules used in this study. The first three rules favor the phrases starting with a chord note, where the chord root note is more important than the second and third notes. Rules 4–6 encourage using chord notes for the last note of phrases in that a chord note makes the music section complete. The last two rules regulate that the distance between phrases should be in a reasonable range since an excessively large leap usually destroy the structure of music.

The proposed evaluation method holds two major advantages: First, the evaluation criterion is consistent. Since the compositions are evaluated by the music theory rules, the evaluation is objective. Restated, this evaluation is not affected by personal experience or preference as that in the traditional human-machine-interaction evaluation. Second, the evaluation is stable. The human-assisted evaluation usually suffers from human fatigue. In addition, the musical sensitivity of humans will decrease after long-time listening. The evaluation using the proposed rules based on music theory can address these issues and afford an effectual guideline for the GA to generate satisfactory compositions.

### C. Genetic Operators

The genetic operators in GA include parent selection, crossover, mutation, and survivor selection. GA selects chromosomes as parents from the population and then performs crossover and mutation operations to generate their offspring.

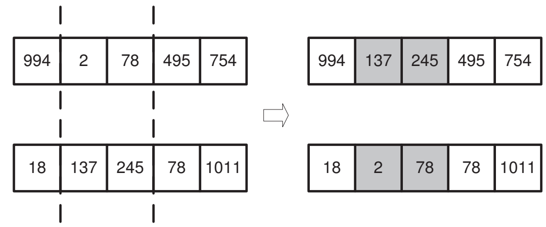


Figure 4. Crossover operation

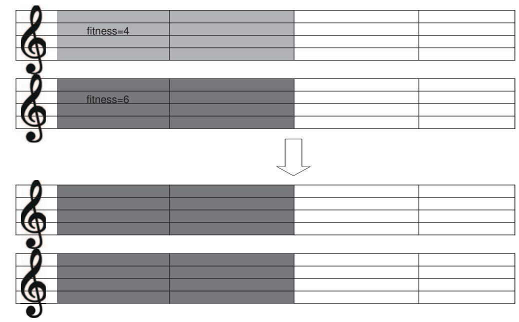


Figure 5. Example of musical form

This study adopts the binary tournament selection in view of its good performance. The binary tournament selection chooses the fitter of two randomly picked chromosomes from the population as a parent. Performing this selection twice yields a pair of parents for the following crossover and mutation operations.

The crossover operator generates offspring by recombining the parental information. For the music composition, the crossover needs to be specially designed since arbitrary recombination of two parents can hardly result in acceptable compositions. This study introduces the notion of crossover unit into the 2-point crossover to address this issue. In the modified 2-point crossover, the cutting points can only lie between bars. Accordingly, the crossover randomly cuts two selected parents and then exchanges the bars in the way of order crossover to generate offspring (cf. Figure 4).

Next, the mutation operator slightly changes the offspring for exploring the problem space. In light of the integer chromosome representation, this study uses the random resetting mutation. This mutation operator probabilistically replaces one randomly-picked note with a random value, which means a pattern will be replaced with another pattern of the same length.

For the survivor selection, this study adopts the $(\mu + \lambda)$ strategy that merges the parent and offspring populations to compete for survival into the next generation.

### D. Post-processing

The best chromosome obtained from the GA will be performed with post-processing to enhance its structure and euphony. The post-process follows that used in [16]: First, the segments of the best chromosome are copied or replaced according to the music form of the sample composition.

Table V
FITNESS EVALUATION RULES AND WEIGHTS

| No. | Evaluation rule | Weight |
|---|---|---|
| 1 | The first note of phrase is the chord root note | +3 |
| 2 | The first note of phrase is the chord 2nd note | +1 |
| 3 | The first note of phrase is the chord 3rd note | +1 |
| 4 | The lase note of phrase is the chord root note | +1 |
| 5 | The lase note of phrase is the chord 2nd note | +1 |
| 6 | The lase note of phrase is the chord 3rd note | +1 |
| 7 | The semitone between the connected notes of phrases is less than 8 | +2 |
| 8 | The semitone between the connected notes of phrases is more than 11 | –2 |

Table VI
PARAMETER SETTING OF GA

| Parameter | Value |
|---|---|
| GA type | Generational |
| Representation | Integer (patterns) |
| Population size | 100 |
| Selection | Binary tournament |
| Crossover | 2-tournament |
| Crossover rate | 0.9 |
| Mutation | Random resetting |
| Mutation rate | 1/chromosome_length |
| Survivor | $\mu + \lambda$ |
| Termination | 10000 generations |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 3 | 4 | 5 | 3 | 4 | 4 | 5 | 3 |
| 3 | 4 | 5 | 3 | 4 | 4 | 5 | 4 |
| 6 | 7 | 7 | 4 | 6 | 7 | 7 | 3 |

Figure 6. Phrase structure of Chrysanthemum Terrace

Table VII
CHORD PROGRESSION OF CHRYSANTHEMUM TERRACE

| Measure | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
|---|---|---|---|---|---|---|---|---|---|
| Chord | C | C | C | Dm | G | G | C | Dm | |
| Measure | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | |
| Chord | C | C | C | Dm | G | G | C | Dm | |
| Measure | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| Chord | C | Am | Dm | Em | C | Am | Dm | Em | C |

Figure 5 shows that, according to the musical form, measures 1–2 should be identical to measures 5–6. Since the latter has a higher evaluation score, it is selected to replace the former. Second, we add accompaniment using the given chords for the resultant compositions.

## V. EXPERIMENTAL RESULTS

This study carries out several experiments to evaluate the performance of the proposed evolutionary composition system. Table VI summarizes the parameter setting for the GA in the experiments. The sample composition used here is "Chrysanthemum Terrace" composed by Jay Chou in 2006. This song has 24 phrases and 123 notes. Figure 6 shows its phrase structure; Table VII lists its progression of chords, which is used as the given chords for evolutionary composition. The generated compositions (WAV files) can be downloaded via http://cilab.cs.ccu.edu.tw/PatMusic2015.zip.

Firstly, we investigate the effect of minimum support values on the performance of the evolutionary composition system. Figure 7 compares the progress of mean best fitness over 30 runs of the proposed GA for generating music, where different minimum support values in pattern mining are experimented. The GA using patterns of support value 0.01 results in the worst fitness and slowest convergence speed. For the GA using patterns of higher support values (i.e., minimum support 0.01), the resultant fitness and convergence speed improve. This outcome indicates the advantage of using high-support patterns. However, an even higher minimum support value ($\geq$ 0.05) shrinks the search space and leads to worse performance than minimum support 0.01, revealing a high minimum support does not guarantee highly fit chromosomes. The high fitness of minimum support 0.01 also reflects that low minimum support values may allow GA to achieve better connection between phrases than high minimum support values.

Figures 8 and 9 show the compositions obtained from different stages of the GA using minimum support 0.01. The initial composition has many discordant notes and a loose music structure. The leaps between phrases are disorganized; in particular, the phrases barely have chord notes at the beginning and the end, reflected on the low fitness values of initial compositions. The proposed GA substantially improves the composition: First, the leaps are mostly in the reasonable range and well-connected. Second, the first and last notes of phrases evolve to be chord notes, enhancing the harmony of compositions. Third, the musical form improves the structure of the compositions generated by GA.

Regarding the fitness values, random compositions at initialization score near 90, whereas the average score (fitness value) of proto-compositions obtained from GA ranges from 120 to 125. The musical form further increases the score by 5 to 10 points on average.

## VI. CONCLUSIONS

This study proposes an evolutionary composition system using GA and pattern mining. The sequential pattern mining is adopted to explore the composer's style and character from his/her works. The patterns serve as the basis for chromosome representation. By this way, the style is considered in the evolutionary composition. In addition, a novel fitness function

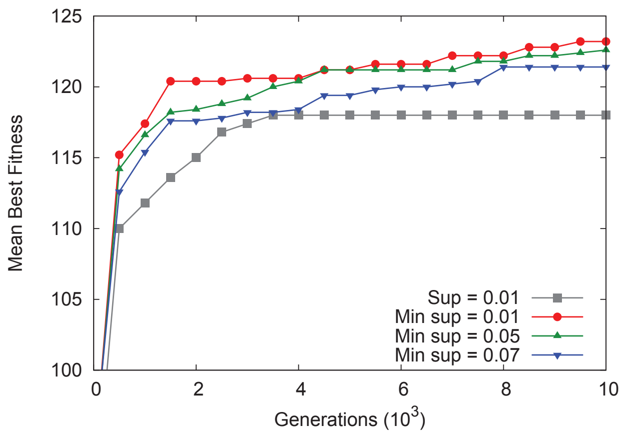


Figure 7. Progress of mean best fitness against generations for GA using patterns of support value (Sup) 0.01 and minimum support (Min Sup) 0.01, 0.05, and 0.07, respectively

based on music theory is proposed to smooth the connection between phrases for the evolutionary composition system.

The proposed evolutionary composition has three major advantages. First, the sequential pattern mining helps to find out the preference of Jay Chou in composition. Second, the chromosome representation based on patterns substantially reduces the numerous possible combinations of notes and reserves the music style of the composer. Third, the fitness evaluation is objective and can overcome the fatigue issue of human-assisted interactive evaluation. The use of musical form further improves the structure of resultant compositions.

Experimental results show that the proposed method can generate satisfactory compositions. The resultant compositions behave well in the connection of phrases and the correspondence with chords. According to the audience with music background, the resultant compositions sound harmonious and follow the sense of melody progression. Some tasks remain for future work: more rules and factors can be considered in the fitness evaluation. In addition, future study can investigate the strategies and applicability of the patterns. The patterns obtained from different genres of a single composer can be further distinguished for better indication.

## References


[1] R. A. McIntyre, "Bach in a box: The evolution of four part baroque harmony using the genetic algorithm," in *Proceedings of IEEE Congress on Evolutionary Computation*, 1994, pp. 852–857.

[2] P. Laine and M. Kuuskankare, "Genetic algorithms in musical style oriented generation," in *Proceedings of IEEE World Congress on Computational Intelligence*, 1994, pp. 858–862.

[3] A. Pazos, A. Santos del Riego, J. Dorado, R. Caldalda *et al.*, "Genetic music compositor," in *Proceedings of IEEE Congress on Evolutionary Computation*, 1999.

[4] M. Marques, V. Oliveira, S. Vieira, and A. Rosa, "Music composition using genetic evolutionary algorithms," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2000, pp. 714–719.

[5] M. Towsey, A. Brown, S. Wright, and J. Diederich, "Towards melodic extension using genetic algorithms," *Educational Technology & Society*, vol. 4, no. 2, pp. 54–65, 2001.

[6] J. Schoenberger, "Genetic algorithms for musical composition with coherency through genotype," Master's thesis, College of William and Mary, 2002.

[7] Y. Khalifa, B. Khan, J. Begovic, A. Wisdom, and A. Wheeler, "Evolutionary music composer integrating formal grammar," in *Proceedings of the Conference on Genetic and Evolutionary Computation*, 2007, pp. 2519–2526.

[8] Y. Chen, "Interactive music composition with evolutionary computation," *NCLab Report*, 2007.

[9] T. Fu, T. Wu, C. Chen, K. Wu, and Y. Chen, "Evolutionary interactive music composition," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, 2006, pp. 1863–1864.

[10] C.-H. Liu and C.-K. Ting, "Evolutionary composition using music theory and charts," in *Proceedings of IEEE Symposium on Computational Intelligence for Creativity and Affective Computing*, 2013, pp. 63–70.

[11] C.-L. Wu, C.-H. Liu, and C.-K. Ting, "A novel genetic algorithm considering measures and phrases for generating melody," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2014, pp. 2101–2107.

[12] L. Luo, P.-F. Lu, and Z.-F. Wang, "A real-time accompaniment system based on sung voice recognition," in *Proceedings of International Conference on Pattern Recognition*, 2008, pp. 1–4.

[13] H.-T. Chen, M.-H. Hsiao, W.-J. Tsai, S.-Y. Lee, and J.-Y. Yu, "A tempo analysis system for automatic music accompaniment," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2007, pp. 64–67.

[14] J.-Y. Jo, Y. ho Kim, H. jo Kang, and J. shin Lee, "Chord-based musical composition and incorporating it into auto-accompaniment instrument," in *Future Generation Communication and Networking*, vol. 2, 2007, pp. 429–432.

[15] I. Simon, D. Morris, and S. Basu, "Mysong: automatic accompaniment generation for vocal melodies," in *Proceedings of SIGCHI Conference on Human Factors in Computing Ssystems*, 2008, pp. 725–734.

[16] C.-H. Liu and C.-K. Ting, "Polyphonic accompaniment using genetic algorithm with music theory," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2012, pp. 1–7.

[17] G. Acampora, J. M. Cadenas, R. D. Prisco, V. Loia, E. M. Ballester, and R. Zaccagnino, "A hybrid computational intelligence approach for automatic music composition," in *Proceedings of IEEE International Conference on Fuzzy Systems*, 2011, pp. 202–209.

[18] F.-F. Kuo and M.-K. Shan, "A personalized music filtering system based on melody style classification," in *Proceedings of IEEE International Conference on Data Mining*, 2002, pp. 649–652.

[19] A. Ujlambkar and V. Attar, "Automatic mood classification model for indian popular music," in *Proceedings of Asia on Modelling Symposium*, 2012, pp. 7–12.

[20] S.-C. Chiu and M.-K. Shan, "Computer music composition based on discovered music patterns," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, 2006, pp. 4401–4406.

[21] J.-M. Ren, Z.-S. Chen, and J.-S. Jang, "On the use of sequential patterns mining as temporal features for music genre classification," in *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, 2010, pp. 2294–2297.

[22] J.-M. Ren and J. Jang, "Discovering time-constrained sequential patterns for music genre classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1134–1144, 2012.

[23] C. Kofod and D. Ortiz-Arroyo, "Exploring the design space of symbolic music genre classification using data mining techniques," in *Proceedings of the International Conference on Computational Intelligence for Modelling Control Automation*, 2008, pp. 43–48.

[24] S.-C. Chiu, M.-K. Shan, J.-L. Huang, and H.-F. Li, "Mining polyphonic repeating patterns from music data using bit-string based approaches," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2009, pp. 1170–1173.

[25] M.-K. Shan, F.-F. Kuo, and M.-F. Chen, "Music style mining and classification by melody," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2002, pp. 97–100.

Figure 8. Resultant composition from initialization

Figure 9. Resultant composition from GA