2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

A Novel Genetic Algorithm Considering Measures and Phrases for Generating Melody

Chia-Lin Wu, Chien-Hung Liu, and Chuan-Kang Ting
Department of Computer Science and Information Engineeringand and
Advanced Institute of Manufacturing with High-tech Innovations
National Chung Cheng University
Chia-Yi 621, Taiwan
Email: {wcl101m, lch101p, ckting}@cs.ccu.edu.tw

***Abstract*—Composing music through evolutionary algorithms has received increasing attention recently. To establish a standard of composing, some studies were proposed on the basis of analysis on musicians, statistics of music details, and rule of thumbs. These methods have achieved some promising results; however, generating melody is still a formidable challenge to computer composition because of the considerable permutations of notes. This study develops a genetic algorithm (GA) based on music theory to generate melody. In particular, we use the rhythm of existing songs as the basis to generate new compositions instead of generating music from scratch; that is, the GA keeps the rhythm of an existing song and rearranges the pitches of all notes for a new composition. Three crossover operators are further proposed to improve the performance of GA on composition. The experimental results show that the GA can achieve satisfactory compositions. The three crossover operators outperform 2-point crossover in the fitness of resultant compositions.**

I. Introduction

Music is an important communication method for people to express their mood, emotion, and feeling. Composing music for different scenes becomes more and more mature; especially, music theory has been developed and studied as a standard for music composition for many years. It integrally records the regulations of music and provides the criteria for composing music. Melody is a key element in music composition because it conveys a strong impression and plays an important role in distinguishing and memorizing compositions. Owing to these features, melody usually guides the direction of composition. However, generating melody is a big challenge and a versatile effective standard for composing melody is still lacking.

Genetic algorithm (GA) is a well-known evolutionary algorithm and has been utilized in automatic composition and accompaniment system in view of its successes in dealing with large-scale search and optimization problems. The GA-based composition systems ordinarily evaluate the generated compositions based on its interaction with humans. The dependence on human feedback, nevertheless, suffers from the fatigue and decreased sensitivity after long-time listening. This issue makes the fitness evaluation very exhaustive and even impractical. Recently, Liu and Ting [17] proposed a GA using fitness evaluation based on music theory. The proposed method can resolve the issue of human feedback in evaluation of music compositions. According to music theory, pitches are involved with permutation combination as well as connection between each other. Matic [19] analyzed the variation relation between intervals and consonance in the progression of melody. Freitas and Guimaraes [10] considered multiple objectives to balance the conflicts between music theory and composer's experience. They discussed the trade-off between different note combinations according to the members of pitches and specific chords.

In this study, we propose a novel GA to generate melody. Specifically, the proposed GA uses the rhythm of an existing song as the basis and rearranges the pitches of all notes for a new melody. The new composition adopts the musical form and accompaniment from the original song. In addition, the proposed GA considers measures and phrases in fitness evaluation according to music theory. This study proposes three crossover operators to enhance the performance of GA.

The remainder of this paper is organized as follows. Section 2 reviews related work. The proposed GA is described in section 3. Section 4 presents the experimental results and section 5 gives the conclusions of this study.

II. Related Work

Genetic algorithm has been used to compose and recombine new scores because of its high capability of search and optimization. The GA for computer composition include two major approaches: rule-based GA and interactive GA. Rule-based GA uses explicit rules to determine the fitness function according to music elements, namely, pitch, rhythm, phrase, scale or chord, from personal experience or music theory. Horner and Goldberg [12] proposed composing music by GA using static patterns. McIntyre [20] used the four-part Baroque harmony to establish a stable progression by chords. Tzimeas and Mangina [26] modified the input melodies to construct a new melody based on the patterns generated by jazz scale and rhythms. They also proposed a new design for generating music from the rhythm patterns given by users [27]. Furthermore, Ozcan and Ercal [22] provided improvised melody composition using predetermined chords and rhythms. They focused on pitches rising and falling of melody structure. Liu and Ting [16] designed the evaluation rules based on music theory in polyphony compositions to enhance reasonable progression. In general, rule-based GA can efficiently generate compositions of a specific music style; however, the

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

Table I: Integer-coded representation for notes

Pitch Name	Integer
C	1
$C^{\#}$/ D^{b}	2
D	3
$D^{\#}$ / E^{b}	4
E	5
F	6
$F^{\#}$ / G^{b}	7
G	8
$G^{\#}$ / A^{b}	9
A	10
$A^{\#}$ / B^{b}	11
B	12
C^2 (higher one octave than C)	13
$C^{2\#}/D^{2b}$ (higher one octave than $C^{\#}/D^{b}$)	14
$\vdots$	$\vdots$
C^3 (higher two octaves than C)	25

fixed rules may limit the flexibility for different aims of music composition.

Interactive GA adopts the listener's evaluation on the created music phrases or compositions. Biles [3] developed a real-time human evaluation on the jazz solo segments generated by GA. This study uses Genetic Jammer (GenJam) with the chords, scales, and rhythms from the accompaniment to generate melody. In addition, Biles [4], [5] presented the GenJam architecture for training from human feedback and further utilized the chords, measures, rhythms, and phrases in music composition. He also discussed the differences between the GenJam system and other art generation systems [6]. Jacob [13] proposed combining good pitch sequences from different steps, i.e., acquiring information from composer's motives such as the weighted variation rate, phrase lengths, and transposition table; selecting melody phrases from valid chords and combining them by a human evaluator; and finally outputting the new melody. Johanson and Poli [14] developed an interactive genetic programming (GP) that uses the tree structure to generate music. Tokui and Iba [25] combined GA and GP to generate melodies in real-time. Although interactive GA can prevent unpleasant music phrases, the inevitable fatigue caused by repeatedly listening will gradually run out human evaluators' patience and sensitivity in music evaluation.

In addition to GA, some recent studies use machine learning techniques for music composition and classification. Basili et al. [2] adopted Naïve Bayes classifier, voting feature intervals, and J48 to classify the genres according to melodic intervals, instruments, and meter, respectively. Dannenberg et al. [9] used the classification techniques such as Bayesian classifier, linear classifier, and artificial neural network (ANN) to recognize musical styles. In addition, they proposed an algorithm to generate music scores based on the results. The learning-based GA [7] integrates GA and ANN to increase the accuracy of fitness function by ANN and to reduce the low-creative features by GA. The study also presents the method for training neural network by human mentors interactively. Burton and

notation: Ionian mode on C

chromosome: {1, 5, 12, 15, 13, 20, 17, 15, 8}

Figure 1: An example chromosome

Vladimirova [8] employed ANN in rhythm composition and applied GA to perform the drum machine. Gibson and Byrne [11] presented a combination of ANN and GA to generate four-part music. Manaris et al. [18] used ANN to construct features of classical music and used them to generate music by GP. Spector and Alpern [24] devised an ANN trained from Charlie Parker melodies to obtain the features for GP to compose music. Moreover, Acampora et al. [1] used the fuzzy system and data mining to construct GA rules. Ramirez et al. [23] used machine learning techniques to build the model for fitness function.

III. Proposed Method

This paper aims to construct a GA using western music theory in the evaluation rules to address the fatigue and sensitivity issues of human feedback. More specifically, we adopt music theory [21] and jazz music theory [15] to construct GA operators, analyze the input information, and evaluate the generated melodies. Additional details about the proposed GA are described below.

A. Representation

This study adopts the integer representation for the GA. The genes of a chromosome are encoded by integers ranging from 1 to 25 to represent the pitches. An octave contains twelve pitches according to the twelve-tone equal temperament. Table I lists the integers corresponding to each pitch; for instance, integer 1 stands for pitch C and 15 denotes pitch D^2. Additionally, any scale of different key signature is indicated by the same integer. For example, integer 1 represents C in Ionian mode as well as D in Ionian mode, despite their difference in pitch. Figure 1 presents an example chromosome and its corresponding notes in a measure.

B. Fitness Function

The proposed fitness function uses the information of scale and chords. Two factors are considered in evaluating the arrangement of pitches: measures and phrases.

1) Measures: The main idea of fitness evaluation based on measures is to handle the proportion of scale notes, chord notes, passing notes, and semitones in a measure. Table II list six rules to be examined for each measure. Therefore, a measure can get the full score (6) if it matches all the six rules, whereas it gets a negative score if matching the fifth or sixth rule. Specifically, the fifth rule calculates the score by

$$-\sum_{i=1}^{m} f_i\left(x\right) \quad \text{with} \quad f_i(x) = \begin{cases} x-7 & x>7 \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Table II: Evaluation rules based on measures

No.	Rule	True	False
1	scale member$<$chord member	+1	-
2	passing tone$<$chord member	+1	-
3	passing tone$\leq$scale member	+1	-
4	appear at least one root or fifth chord member	+1	-
5	all semitone ≤ 7	+1	eq. (1)
6	all semitone$\neq 13$	+1	eq. (2)

where x denotes a semitone and m represents the total number of measures. The sixth rule computes the score by

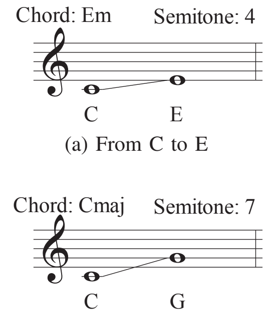
$$-\sum_{i=1}^{m} g_i(x) \quad \text{with} \quad g_i(x) = \begin{cases} x & x = 13 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

These rules are used to control the composition of notes in a measure. The chord notes can stabilize the sound of a measure while the scale notes diversify it. Accordingly, the first rule is defined by limiting the number of scale notes lower than that of chord notes for increasing stability. The fifth and sixth rules are used to prevent the leaps between notes and control the number of semitones.

2) Phrases: The phrase-based evaluation focuses on the consonance between notes. According to harmonics, chord progression serves as the basis of music variation. The chord progression comprises three essential functions in the diatonic system: tonic function, subdominant function, and dominant function. The tonic function is composed of the intervals with high consonance and is usually employed in the start or the end of a phrase. The subdominant function includes the intervals of consonance and imperfect consonance, which are used to dominate the state of music processing. Finally, the dominant function consists of the dissonance that implies the end of music. The music theory suggests using cadence, a progression from dominant to tonic, to hint the approach of sentence or music end. In order to guide the dissonant state of dominant function to the consonant state of main chords, the neighbor pitches have to be limited by resolution. The music can be diversified by the changes between consonance and dissonance states as long as the dissonant state can be resolved.

Figure 3 presents an authentic cadence example that constructs the chord progression from dominant to tonic. The example indicates that the fourth and seventh notes must be resolved using a limited interval. According to music theory, the third and root notes of the subsequent chord can resolve the fourth and seventh notes of the current chord, respectively. In addition, the connection between dissonant and consonant notes usually uses short intervals.

Table III lists 17 rules that identify the undesirable progressions of notes. Symbols $R1_j$, $R2_j$, $R3_j$, and $R4_j$ denote the amount of matched rules, where j indicates the number of phrases. Rule 1 specifies the current note of chord identity without cadence. Rules 2–9 indicate the cases of cadence when the current note belongs to a dominant, half-diminished, or diminished chord and the subsequent note belongs to a major or minor chord. Rules 10–15 show the scale identity of the



(a) From C to E

(b) From C to G

Figure 2: Examples of fitness evaluation based on phrases

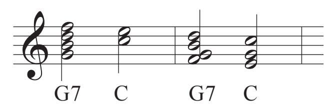


Figure 3: Cadence: from dominant to tonic function

current note. The semitones for rules 10–11 are set to be more than 2 to ensure the subsequent note can resolve the current note. Rule 16 denotes the disallowed passing tone. Rule 17 requires that the final note in a phrase must be a chord identity to ensure the end of phrase consonant.

The phrase-based evaluation checks if an adjacent note matches the rules. Figure 2a illustrates a case of matching rule 11: without cadence, the Aeolian mode on G and E minor chords has the current note C of fourth scale identity and the subsequent note E of root chord identity. Since this progression satisfies the condition and requirement (4 semitones), it obtains $R2_j + 1$. In the example of Figure 2b, note G of fifth chord identity follows note C of root chord identity, but the distance (7 semitones) between C and G does not satisfy the requirement of rule 1.

In summary, the fitness Ph based on phrases is calculated by

$$Ph = \sum_{j=1}^{p} h(R1_j) + h(R2_j) + h(R3_j) + h(R4_j) \tag{3}$$

with

$$h(y) = \begin{cases} -(y \times l_j) & y > 0 \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where l_j denotes the length of the current phrase, and p is the number of phrases in the music score. Finally, the fitness of a chromosome is defined by the sum of fitness based on measures and phrases.

C. Genetic Operators

The genetic operators of GA include parent selection, crossover, mutation and survivor selection. For the parent selection, this paper adopts 2-tournament selection to pick parents from the population. Each pair of selected parents is further performed with crossover and mutation to produce

Table III: Illegal fitness rules with phrase

No.	Rules	Requirement	Count
1	Without the cadence, a current note of chord identity, and next note of any identity	semitone > 7	$R1_j + 1$
2	In cadence, a current note is root of chord identity by dominant quality, and next note is root or fifth of chord identity	semitone > 7	$R1_j + 1$
3	In cadence, a current note is root of chord identity by half-diminished or diminished quality, and next note is root or third of chord identity	semitone > 7	$R1_j + 1$
4	In cadence, a current note is third of chord identity, and next note is root or third note of chord identity	semitone > 2	$R1_j + 1$
5	In cadence, a current note is fifth of chord identity, and next note is root, third, or fifth of chord identity	semitone > 2	$R1_j + 1$
6	In cadence, a current note is seventh of chord identity, and next note is root, third, or fifth of chord identity	semitone > 2	$R1_j + 1$
7	In cadence, a current note is ninth of chord identity and next note of chord identity	semitone > 2	$R1_j + 1$
8	In cadence, a current note of scale or passing tone, and next note of any identity	always true	$R1_j + 1$
9	In cadence, a current note of chord identity, and next note not conform to rules from 2 to 7	always true	$R1_j + 1$
10	Without the cadence, a current note is fourth or seventh of scale identity, and next note of chord identity	semitone > 2	$R2_j + 1$
11	Without the cadence, a current note is fourth or seventh of scale identity, and next note of scale identity neither fourth nor seventh	semitone > 2	$R2_j + 1$
12	Without the cadence, a current note is fourth or seventh of scale identity, and next note is fourth or seventh of scale identity	always true	$R2_j + 1$
13	Without the cadence, a current note of scale identity neither fourth nor seventh note, and next note of chord identity	semitone < 7	$R2_j + 1$
14	Without the cadence, a current note of scale identity neither fourth nor seventh note, and next note of scale identity	semitone > 4	$R2_j + 1$
15	Without the cadence, a current note of scale identity, and next note of passing tone identity	always true	$R2_j + 1$
16	Without the cadence, a current note of passing tone and next note of any identity	always true	$R3_j + 1$
17	the current note is a last note in current phrase; the current note of scale or passing tone identity, and no next note	always true	$R4_j + 1$

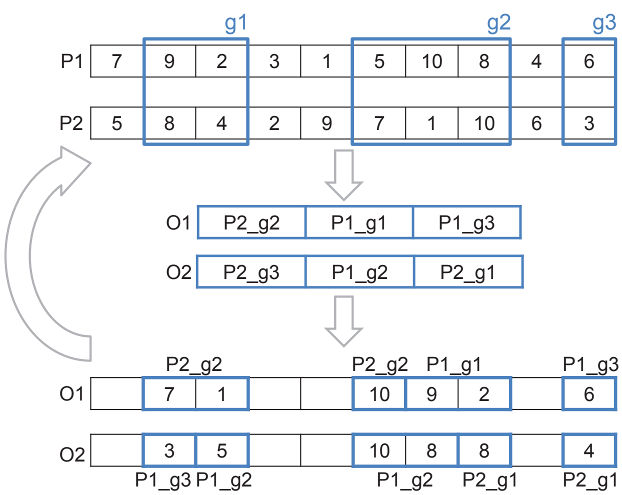


Figure 4: Crossover operation

their offspring. The crossover operator needs to be specially designed to prevent destroying the structure of composition. This paper proposes three methods for selecting the fragments to be switched in the crossover operation.

1) Chord-based Switch (CS): The chord-based switch selects only the fragments with absolutely identical chords.
2) Root-based Switch (RS): The root-based switch selects the fragments that have the same root of chords and differ by less than one pitch in the chords.
3) Analogous Switch (AS): The analogous switch is similar to the root-based switch but permits selecting the fragments with different root of chords.

The proposed crossover first exchanges the selected fragments of two parents in the way of uniform crossover and then rearranges their sequences randomly. As illustrated in Figure 4, three fragments (g1, g2, and g3) are selected to exchange between parents. The sequences of the fragments in the offspring are further rearranged randomly.

The mutation operator slightly changes the offspring for exploring the problem space. In the light of integer chromosome representation, we adopt the random resetting mutation to replace one randomly-picked note with a random value. For the survivor selection, this paper adopts the $\mu + \lambda$ strategy to compete the parent and offspring populations for survival into the next generation.

D. Musical Form

Musical form can make the music composition structural and complete [17]. In this study, we reserve the musical form of the original song for the generated compositions. After the evolutionary process of GA, the fragments of the best chromosome will be copied or replaced according to the music form of the original song. Figure 5 shows an example of improving the composition by musical form. Measures 4–7 are the same as measures 25–27 in the musical form. According the rules of phrase-based evaluation, the former is better in that it induces fewer rules; thus, this fragment is selected to replace measures 25–27. This modification using musical form can keep the structure of the original song and improve the results of GA.

IV. Experiments

This study conducts several experiments to evaluate the performance of the proposed GA. Table IV summarizes the parameter setting for the GA in the experiments. The sample results (MP3 files) can be downloaded via

Figure 5: Improvement by musical form

Table IV: Parameter setting

Parameter	Value
Representation	$c_i \epsilon \{1, 2, \ldots, 25\}$
Population size	400
Initialization	Random
Parent selection	2-tournament
Mutation	Random resetting
Mutation rate	$P_m = 1/\text{chromosome\_length}$
Crossover rate	$P_c = 0.9$
Survival selection	$\mu + \lambda$
Termination	2000 generations

http://cilab.cs.ccu.edu.tw/GAMusic2014.zip. Figure 6 compares the progress of mean best fitness (MBF) over 100 runs of test GAs, for which the classic 2-point crossover and the proposed three crossover (CS, RS, and AS) operators are considered. The song used as the basis is "Reunion" from the album "Song to Fly" composed by Yoko Kanno in 1998. We modify the song into a 3-minute version (cf. Table V).

Figure 6 shows that the proposed CS, RS, and AS crossover operators outperforms 2-point crossover in convergence speed. Table VI compares the MBF values obtained from random initialization and the evolutionary process of GA. These results indicate the advantages of the proposed crossover operators in solution quality and convergence speed. Furthermore, Figure 7 shows the resultant melodies from GA using different crossover operators. These crossover operators achieve similar fitness but the structures of their compositions are different. The CS crossover switches only the fragments with the same chords, which can effectively prevent the risk of placing wrong notes but decreases the diversity of crossover results. The RS crossover enables switching fragments with the same root of chord. The root and fifth chord member are thus more likely to be retained . The AS crossover resembles the RS crossover in retaining the root and 3rd chord member; however, the former gives an opportunity to separate sequences to improve their intervals and increase the diversity of sequences.

Figures 8 and 9 compare the compositions obtained from different stages of the GA. Since the initial compositions are generated randomly, the melody is chaotic and has plenty

Table V: The song used in the experiments

Year	1998
Music Name	Song to fly - "Reunion"
Total measure	59
Total phrase	18
Total note	199
Total chord quantity	11

Table VI: Mean best fitness (MBF) for GA using different crossover operators

Method	MBF
Initialization	-1395.00
GA with 2-point crossover	318.27
GA with CS crossover	323.82
GA with RS crossover	324.25
GA with AS crossover	322.67

illegal identity notes and discordant intervals. For example, in the fourth measure, by Ionian mode on F and F major chord, the second note G and the third note D^b get negative scores according to rule 15 of phrase-based evaluation. This dissonant combination accounts for the low fitness of initial compositions. Through evolution, GA can effectively improve the entire composition: passing tones on inappropriate locations are greatly reduced and the intervals between two notes are generally decreased to prevent huge leaps. Finally, the musical form systematically enhances the structure of compositions.

V. Conclusions

This study proposed a GA using fitness evaluation based on music theory to generate melody. The harmony of intervals is adopted in the fitness function. More specifically, the fitness function considers the arrangement of measures and phrases according to music theory. Three crossover operators using the features of chord are presented to improve the performance of GA. Given the chords and intervals, the proposed GA can effectively rearrange the permutation of pitches to create new melody. Experimental results show that the GA can achieve satisfactory compositions. The adopted music theory, moreover, provides an objective measure for the evolutionary system to generate melody.

Some directions remain for future work. According to the experiments, the fitness reflects the level of coincidence with the rules from music theory. This study considers measures and phrases in the fitness evaluation. However, more aspects of music, such as music form and rhythm, can be further considered. Musical analysis will be helpful in this regard.

Acknowledgment

This work was supported by Ministry of Education, Taiwan.

References

[1] G. Acampora, J. M. Cadenas, R. De 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*, pages 202–209, 2011.

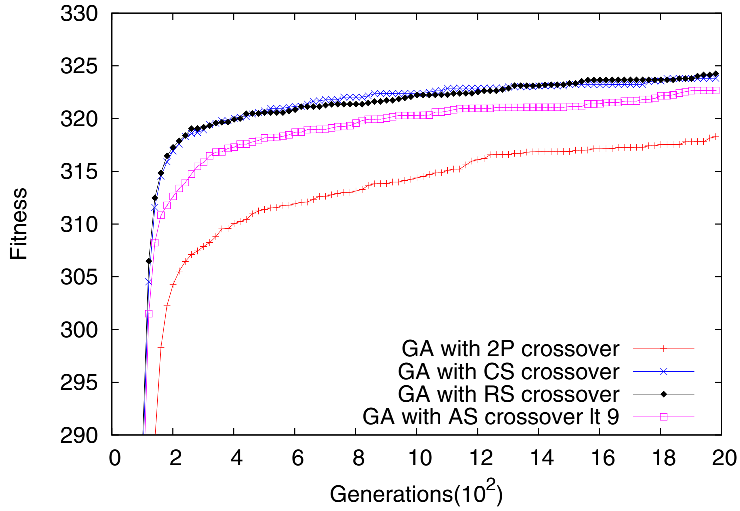


Figure 6: Progress of the mean best fitness

Figure 7: Comparison of the compositions obtained from GA using different crossover operators

[2] R. Basili, A. Serafini, and A. Stellato. Classification of Musical Genre: A Machine Learning Approach. In *Proceedings of The Internationa Society for Music Information Retrieval*. Citeseer, 2004.

[3] J. A. Biles. GenJam: A Genetic Algorithm for Generating Jazz Solos. In *Proceedings of the International Computer Music Conference*, pages 131–131, 1994.

[4] J. A. Biles. Life with GenJam: Interacting with a Musical IGA. In *Proceedings of 1999 IEEE International Conference on Systems, Man, and Cybernetics*, volume 3, pages 652–656. IEEE, 1999.

[5] J. A Biles. GenJam: Evolutionary Computation Gets a Gig. In *Proceedings of the Conference for Information Technology Curriculum, Rochester, New York, Society for Information Technology Education*, 2002.

[6] J. A. Biles. GenJam in Perspective: A Tentative Taxonomy for GA Music and Art Systems. *Leonardo*, 36(1):43–45, 2003.

[7] J. A. Biles, P. G. Anderson, and L. W. Loggi. Neural Network Fitness Functions for a Musical IGA. In *Proceedings of International Computing Sciences Conferences*, 1996.

[8] A. R. Burton and T. Vladimirova. Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition. In *Artificial Neural Nets and Genetic Algorithms*, pages 219–223, 1998.

[9] R. B. Dannenberg, B. Thom, and D. Watson. A Machine Learning Approach to Musical Style Recognition. In *Proceedings of International Computer Music Conference*. University of Michigan, 1997.

[10] A. R. R. Freitas and F. G. Guimaraes. Melody Harmonization in Evolutionary Music Using Multiobjective Genetic Algorithms. In *Proceedings of the Sound and Music Computing Conference*, 2011.

[11] P. M. Gibson and J. A. Byrne. Neurogen, Musical Composition Using Genetic Algorithms and Cooperating Neural Networks. In *Proceedings of Second International Conference on Artificial Neural Networks*, pages 309–313, 1991.

[12] A. Horner and D. E. Goldberg. Genetic Algorithms and Computer-assisted Music Composition. *Urbana*, 51(61801):14, 1991.

[13] B. Jacob. Composing with Genetic Algorithms. In *Proceedings of International Computer Music Conference*, pages 452–455, 1995.

[14] B. Johanson and R. Poli. *GP-music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters*. Citeseer, 1998.

[15] M. Levine. *The Jazz Theory Book*. Sher Music; Unk edition, 1995.

[16] C.-H. Liu and C.-K. Ting. Polyphonic Accompaniment Using Genetic Algorithm with Music Theory. In *Proceedings of IEEE Congress on Evolutionary Computation*, pages 1–7, 2012.

[17] 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*, pages 63–70. IEEE, 2013.

[18] B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro, and J. Romero. A Corpus-based Hybrid Approach to Music Analysis and Composition. In *Proceedings of the National Conference on Artificial Intelligence*, volume 22, page 839. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

[19] D. Matić. A Genetic Algorithm for Composing Music. *The Yugoslav Journal of Operations Research*, 20(1):157–177, 2010.

[20] R. A. McIntyre. Bach in a Box: The Evolution of Four Part Baroque Harmony Using the Genetic Algorithm. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 852–857, 1994.

[21] M. Miller. *The Complete Idiot's Guide to Music Theory*. ALPHA; 2nd edition, 2005.

[22] E. Özcan and T. Erçal. A Genetic Algorithm for Generating Improvised Music. In *Artificial Evolution*, pages 266–277. Springer, 2008.

[23] R. Ramirez, A. Hazan, E. Maestre, and X. Serra. A Genetic Rule-based Model of Expressive Performance for Jazz Saxophone. *Computer Music Journal*, 32(1):38–50, 2008.

[24] L. Spector and A. Alpern. Induction and Recapitulation of Deep Musical Structure. In *Proceedings of International Joint Conference on Artificial Intelligence*, volume 95, pages 20–25, 1995.

[25] N. Tokui and H. Iba. Music Composition with Interactive Evolutionary Computation. In *Proceedings of the 3rd international conference on generative art*, volume 17, pages 215–226, 2000.

Figure 8: Resultant composition from initialization

[26] D. Tzimeas and E. Mangina. Jazz Sebastian Bach: A GA System for Music Style Modification. In *Proceedings of International Conference on Systems and Networks Communications*, pages 36–36, 2006.

[27] D. Tzimeas and E. Mangina. Dynamic Techniques for Genetic Algorithm-based Music Systems. *Computer Music Journal*, 33(3):45–60, 2009.

Figure 9: Resultant composition from GA