Soft Comput (2012) 16:1997–2008
DOI 10.1007/s00500-012-0872-y

FOCUS

Controlling interactive evolution of 8-bit melodies with genetic programming

Maximos A. Kaliakatsos-Papakostas · Michael G. Epitropakis · Andreas Floros · Michael N. Vrahatis

Published online: 15 June 2012

Abstract Automatic music composition and sound synthesis is a field of study that gains continuously increasing attention. The introduction of evolutionary computation has further boosted the research towards exploring ways to incorporate human supervision and guidance in the automatic evolution of melodies and sounds. This kind of human–machine interaction belongs to a larger methodological context called interactive evolution (IE). For the automatic creation of art and especially for music synthesis, user fatigue requires that the evolutionary process produces interesting content that evolves fast. This paper addresses this issue by presenting an IE system that evolves melodies using genetic programming (GP). A modification of the GP operators is proposed that allows the user to have control on the randomness of the evolutionary process. The results obtained by subjective tests indicate that the utilization of the proposed genetic operators drives the evolution to more user-preferable sounds.

M. A. Kaliakatsos-Papakostas (✉) · M. G. Epitropakis · M. N. Vrahatis
Computational Intelligence Laboratory (CILAB),
Department of Mathematics, University of Patras,
26110 Patras, Greece
e-mail: maxk@math.upatras.gr

M. G. Epitropakis
e-mail: mikeagn@math.upatras.gr

M. N. Vrahatis
e-mail: vrahatis@math.upatras.gr

A. Floros
Department of Audio and Visual Arts,
Ionian University,
49100 Corfu, Greece
e-mail: floros@ionio.gr

Keywords Interactive evolution · Music composition · Sound synthesis · Genetic programming · Fitness-adaptive genetic operators

1 Introduction

The creation of artistic material with automatic means is a vast area of research that aims at finding connections between human abstract thought and universal mechanisms. Examples of such mechanisms are pure mathematics, physics and biology among others. With the emergence of digital computers, the interdisciplinary study of topics flourished, allowing the connection of different scientific fields. A connection example is analyzed in this paper between genetic evolution, mathematics and music, under the scope of their interaction with humans. This connection results in user-driven evolution, commonly termed as *Interactive evolution* (IE). Systems that utilize IE constitute a fertile field for the automatic creation of music and generally art, with the constant supervision of a human user. However, user fatigue in IE systems is a common impediment for the evolution, especially for artistic purposes.

The use of *genetic programming* (GP) (Koza 1992) for automatic music composition and sound synthesis with an IE scheme has been previously used in various systems. In Spector and Alpern (1994), Tokui (2000) two systems are presented where genetic algorithms (GA) and GP are combined to modify parts of symbolic music compositions and create novel ones. Some works have utilized automatic fitness raters based on artificial neural networks (ANNs) (Manaris et al. 2007; Spector et al. 1995) or self-organizing maps (Phon-Amnuaisuk et al. 2007) that were trained on specified symbolic music features. For a review of systems that compose symbolic music with genetic-based techniques, the interested reader is referred to (Burton and Vladimirova 1999).

In parallel, evolutionary techniques have been used for sound synthesis but they have mainly focused on creating synthesized sounds that assimilate certain target sounds (Garcia 2000). GP has been used to evolve combinations of sinusoidal oscillators and filters to simulate a target sound too (Garcia 2001). In the project described in Putnam (1994), an IE system is utilized for synthesizing sounds through functions that directly shape waveforms, which are evolved according to fitness values provided by users. However, for these functions it was reported that they "*produced little more than irritating noise and evolved (if at all) very slowly*". Finally, an IE system that is used for measuring possible esthetic measures for sound has been formulated in Kaliakatsos-Papakostas et al. (2012). User fatigue is commonly reported not only in IE systems that create sounds or music, but also graphical art (Lewis 2008). User fatigue in IE systems is an important factor since it not only affects the user's engagement to the rating task, but may also consequently mislead the evolutionary process (Yan and Min 2011).

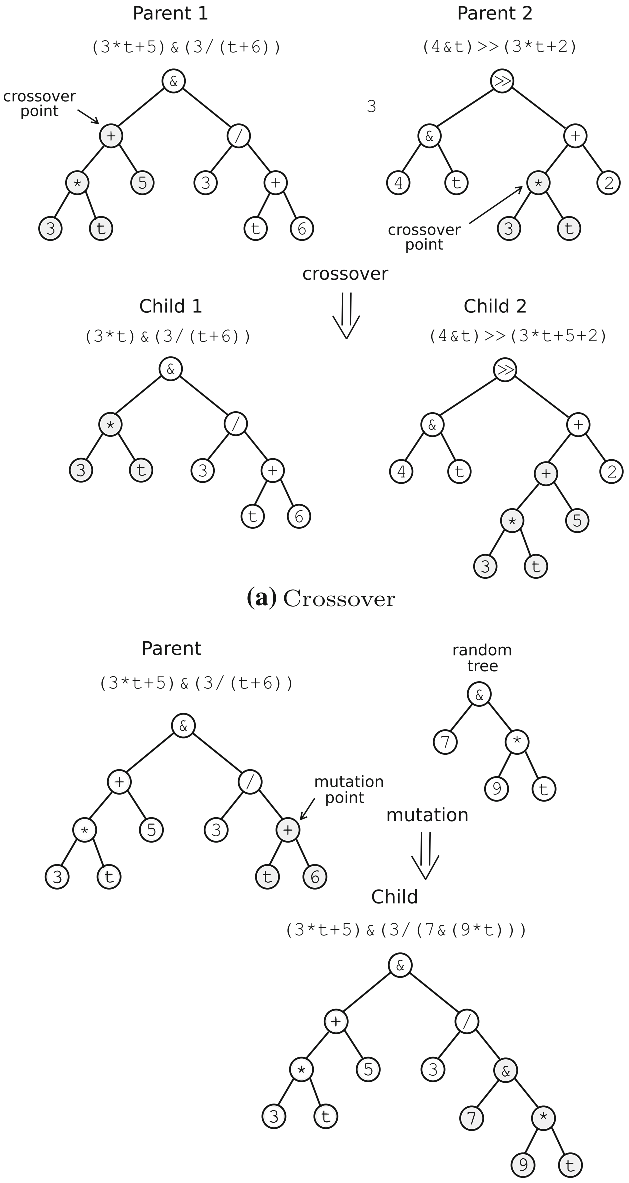
This work proposes a modification of the genetic operators used in GP for reducing the effect of user fatigue in the presented IE system by providing the user with a sense of control over the evolutionary process and thus leading the evolution to subjectively more pleasing output. This is accomplished by employing constraints on the depth that the GP operators act. In the formulation of the problem at hand, the depth of action of the GP operators is indicated to have an impact on the results of the evolutionary process, specifically in the differences between parents and offspring. The presented IE system uses GP to evolve functions which create waveforms with pleasant and interesting sonic and melodic output. These functions, together with some prerequisites for GP are presented in Sects. 2, 3 examines the variability of the melodies produced when applying certain constraints on the genetic operators that evolve them. Motivated by the aforementioned examination, the proposed methodology for applying user control on the evolution is also presented in this section. In Sect. 5, the results obtained through subjective tests are presented, indicating that better melodies are expected to emerge, when the proposed methodology is applied. Some conclusions are discussed in Sect. 6 together with pointers for future work.

2 Background material

This section presents a short introduction to genetic programming (GP) and a description of the mathematical objects that create the melodies for the presented interactive evolution (IE) system. For the GP we describe the genetic operators and the utilized selection methods and also we provide the terminology that is followed throughout the paper. Based on this methodological framework, the next sections discuss the relation of depth constraints on GP operators to the melodic differentiation of offspring.

2.1 Genetic programming

Genetic programming (GP) is a methodology introduced in Koza (1992), and has been widely used in a wide range of applications (Poli et al. 2008). With GP the evolution of



(a) Crossover

(b) Mutation

Fig. 1 Graphical examples of the crossover and mutation operators

programs or functions is possible, through the stochastic combination or alteration of their parts. This is possible with the representation of programs as tree structures. Through this procedure new programs are created that provide better solutions to a certain problem. Next, with the utilization of a selection procedure based on the quality of results provided by each program/tree-clone, new combined or altered programs are chosen. This procedure stops when a program that provides a satisfactory solution is created, or if a maximum number of program-creation epochs has been reached.

The GP methodology, being a part of the general methodology of GA, is inspired by the natural selection introduced in the Darwinian theory of evolution. This fact has established a nomenclature similar to the biological theory counterpart. Thus, each program is referred to as an *individual*, the group of individuals that are candidate for selection is called the *population* (or the *gene pool*) and the individuals created at each stage of the evolution form a *generation*. The individuals that are combined or altered are named as *parents* and the resulting individuals *offspring* or *children*. The aptness of a program is measured by a numeric value that is called *fitness value*. The structural representation of a program is referred to as the *genotype* and its output as the *phenotype*.

Program combination or alteration of programs is realized through a set of genetic operators. In this work, we use the standard *crossover* and *subtree mutation* [or *headless chicken crossover* (Angeline 1997)]. The crossover combines two individuals by swapping a random subtree of each and thus producing two new individuals. The subtree mutation, swaps an entire random tree with a random subtree of an individual. A graphical example of the effects of these two operators is provided in Fig. 1.

The selection methods used in this work are *roulette*, *tournament* and *elitist*. The roulette method acts as if a roulette with random pointers is spun, and each individual owns an angular portion of the roulette that is proportional to its fitness value. Tournament chooses each parent by randomly drawing a number of individuals from the population and selecting only the best of them. Finally, the elitist method selects as parents only the fittest individuals of the population. A thorough description of GP algorithm along with its main operations and characteristics can be found in (Koza et al. 1992; Poli et al. 2008).

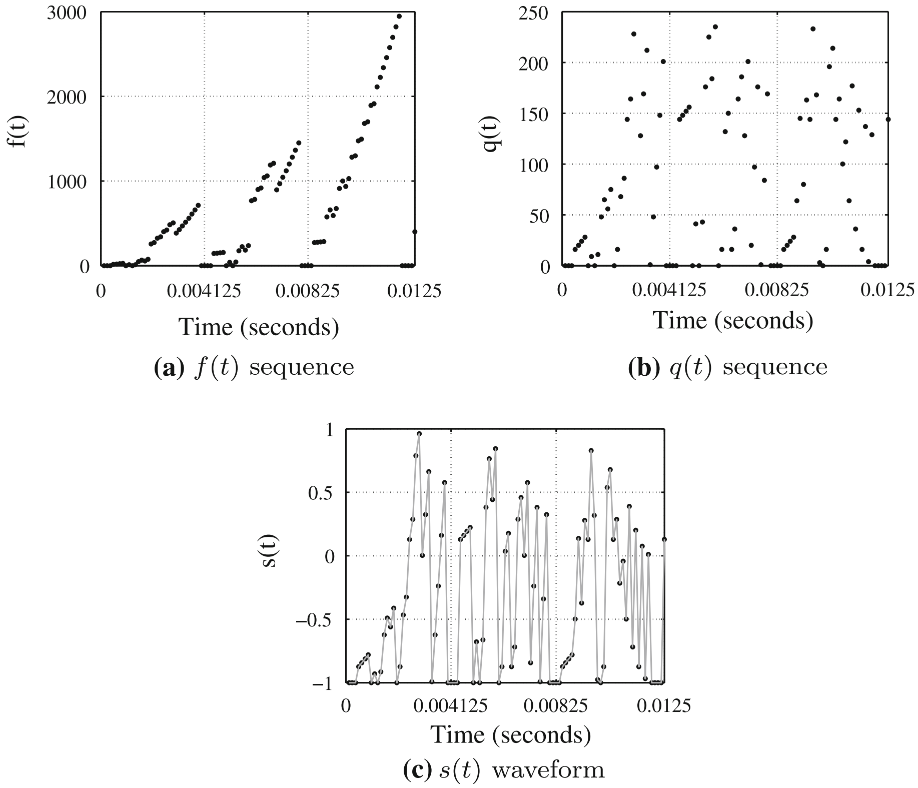
2.2 Simple functions that produce structured sonic output

The IE system described in the paper at hand uses a class of functions which create waveforms with interesting structural coherence from the level of sound texture to the level of musical compositions. Thus, hereby we will be using the terms *waveform* and *melody* interchangeably with similar meanings. These functions are rapidly attracting the attention of many programmers-composers (Heikkilä 2011) in the 8-bit music community (Heikkilä 2011) and they have mainly been used to produce music content with sampling frequency and quantization resolution equivalent to the early PCM digital coding. Greater sampling frequency and/or quantization resolution could be used but the utilized setup provides an esthetic reference point that makes noisy sonic content acceptable, e.g., short segments of noise are perceived as percussion sounds. These functions use not only the standard arithmetic operators ("+", "−", "*", "/"), but we have also experimented with a subset of the available C bitwise operators, namely the bitwise AND (&), OR (|), XOR (ˆ), left shift (<<) and right shift (>>). Furthermore, we have examined the use of a single variable, as described in the next paragraph. For a thorough analysis on the sound properties of the audio signals created by these functions, the interested reader is referred to (Heikkilä 2011). Examples of the tree representations for different such functional expressions are given in Fig. 1.

Algorithm 1 Construction of an 8–bit waveform of 8000Hz sample rate through a function $f(t)$

```
Input: i) A functional expression $f(t)$ and ii) time duration in seconds ($d$)
Output: The waveform of an audio signal, $s(t)$, with $d$ seconds duration
1: for $t = 1$ to $d \cdot 8000$ do
2:     if $f(t) ==$ Not a Number (NaN) then
3:         $q(t) \leftarrow 0$
4:     else
5:         $q(t) \leftarrow \mathrm{mod}([f(t)], 256)$
6:     end if
7:     $s(t) \leftarrow 2\frac{q(t)}{255} - 1$
8: end for
```

Algorithm 1 provides a thorough description for the construction of sound waveforms through the examined functions, while a graphical example is given in Fig. 2. These waveforms have an 8-bit resolution and a sample rate of 8,000 Hz and are constructed with the use of an integer counter, *t*, that takes values between 1 and $d \cdot 8000$, where *d* is the desired duration of the sound output in seconds. The counter, *t*, represents the generated music sample indices. Then the functional expression *f*(*t*) is evaluated for every counter value $t \in \{1, 2, \ldots, d \cdot 8000\}$. The function *f* is computed in an integer C-like context, which means that *f*(*t*) provides integer values for integer input and hereby we consider $f(t) = [f(t)] = \lfloor f(t) + 0.5 \rfloor \in \mathbb{Z}$. Possible division by zero during the computation of *f*(*t*) is assumed to provide a zero value, as indicated in the second line of Algorithm 1. For simulating the wrapping overflowing behavior of 8-bit computer systems, we form the "*quantized*" sequence $q(t) = \mathrm{mod}(f(t), 256)$ for $t \in \{1, 2, \ldots, d \cdot 8000\}$. Finally, we normalize *q*(*t*) in the range



(a) $f(t)$ sequence

(b) $q(t)$ sequence

(c) $s(t)$ waveform

Fig. 2 Example of the transformation of the sequence `f(t) = t*(t>>8*(t>>15|t>>8)&(20|(t>>19)*5>>t|t>>3))`, `t` $\in \{1, 2, \ldots, 100\}$ to $q(t)$ and finally to the waveform $s(t)$

$[-1,\ 1]$ by $s(t) = 2 \cdot (q(t)/255) - 1$ and thus obtain the waveform $s(t)$. The construction of the waveform through Algorithm 1 is unstable, in a sense that minor modifications of the f function may lead to vast differences in the waveform. However, in Sect. 3 indications are presented that the "depth" of the changes in function f is related to the differentiation of the resulting waveforms.

In Sect. 3, we compute the perceived sound distance of the sonic output of these functions, which is also the phenotypical distance of the individuals that represent the respective function. Therefore, we need a measure to compute the distance between the derived sounds. To this end, we extract some features from the waveforms created by individuals and use them as indicators of similarity. Two categories of features were collected. The first, named as *waveform information* features, concerns the information capacity of the quantized sequences that form the waveform. The second category is named as *spectral and cepstral* features and describes some frequency domain characteristics. The cepstral coefficients, expressed through the Mel-Frequency Cepstral Coefficients (MFCCs) have proven to be an effective measure for measuring perceptual distance of audio signals (Terasawa et al. 2005). All the features are demonstrated in Table 1. They are all single numbers, except from the MFCCs which are represented by 12-dimensional vectors. Hence, the vector of waveform features for each waveform-melody is represented as a 19-dimensional vector.

3 Controlling evolution with fitness-adaptive genetic operators

The paper at hand aims to create an IE system which reduces user fatigue by making the process of evolution controllable by the user. Indications are provided that the control of offspring variability during evolution may be realized by the utilization of depth constraints in the application of the GP genetic operators. Therefore, we examine the phenotypical behavior of children that have been evolved with genetic operators which act in a certain depth of the tree (genotypical) representation of the parents. Specifically, we present two experiments which indicate that higher levels of operator action produce individuals that are less similar to their parents. The phenotypical distance among individuals is computed with the use of the sound features presented in Sect. 2.2. Motivated by these experiments, we present a methodology that

Table 1 The features used as indicators of audio similarity

Waveform information	
Fractal dimension (FD)	Fractal dimension of the quantized sequence with the Higuchi (1988) algorithm
Shannon information entropy (SIE)	Shannon information entropy (Shannon 2001) of the normalized (to unit sum) histogram of the quantized sequence
Compressibility through compression rate (CR)	Ratio of the size of the compressed quantized sequence with the Lempel–Ziv algorithm (Ziv 1977) over the size of the uncompressed sequence
Spectral and cepstral features (Tzanetakis 2002)	
Spectral centroid (SC)	The "center of weight" of the spectrogram
Spectral centroid standard deviation (SCSD)	SD of the spectral centroids within short time segments(of 0.1299 s)
Mean spectral flux (SFm)	The mean value of spectral fluxes (Euclidean distances of the spectrogram of short consecutive segments) of segments of 0.1299 s
Spectral flux standard deviation (SFSD)	The standard deviation of the aforementioned spectral fluxes
Mel-frequency cepstral coefficients histogram (MFCCh)	The histogram of 12 mel-frequency cepstral coefficients(MFCCs) (Davis and Mermelstein 1980), produced by computing 13 and removing the DC component

allows the user to have control on the randomness of the evolution derived by the presented system. It has to be noted that we do not attempt to generalize the results of the findings presented here to a wider GP methodological context. Nevertheless, we argue that the proposed methodology is applicable on the presented system. This argument is further amplified by the findings in Sect. 5.

3.1 Depth of operator action and phenotype variation

To examine the dependency of the depth in which a genetic operator acted and the phenotypical variation of the resulting offspring, we have followed the methodology described below. We have created a random *initial generation* of NP individuals, here NP $= 10$, under the constraint that their representation had a depth level greater than 5. To study the phenotypical variation for the crossover and mutation operations in correspondence to the depth level that they act, we have conducted two specific experiments. In the first experiment, we evolved the initial generation for ten generations by only applying the crossover operator and always selecting the offspring as parents for the next generation. During the second experiment, we followed an analogous evolution scheme by applying only the mutation operation. For each experiment, 4 scenarios were applied independently, with the operators acting on different depths. If we suppose that an individual had a depth D, then according to the 4 scenarios the operators acted on the $D-0, D-1, D-2$ and $D-3$ depths, respectively.

At each evolution generation and for every depth scenario we measured the mean phenotypical distance between the generations' offspring and their respective *initial parents*. As a measure of similarity we utilize the Euclidean norm of the vector of relative differences[1] between the feature vectors of the initial parents and their respective offspring. The mean values and the standard deviations (SD) of the distances that describe these differences are illustrated in Fig. 3. It is clear that when an operator acts on the leaf nodes, the sonic output is less likely to change, since the relative differences are smaller. Hence, for the presented methodological context we observe that as the depth in which the operator acts decreases, the differences are getting larger.

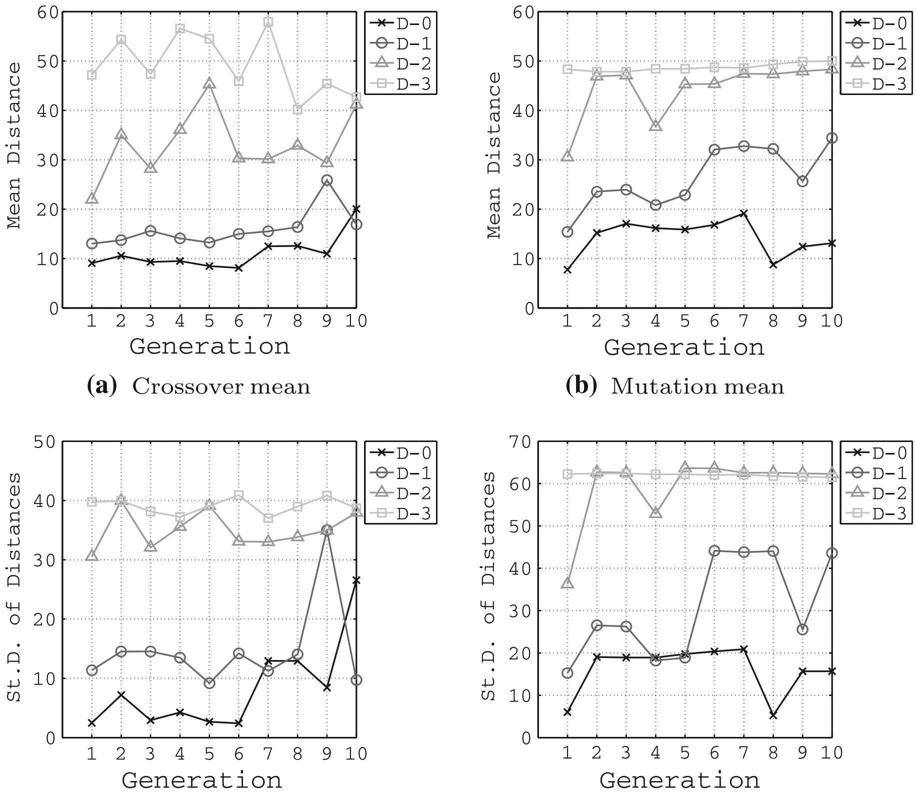
3.2 Fitness-adaptive genetic operators

Our primary concern is making the presented system as user-friendly as possible, letting the user have some control over the randomness of evolution and thus reducing user fatigue, which is a common problem in IE systems that create sound. One of the main reasons for user fatigue in former versions of our system is the creation of individuals that produce uninteresting melodies or noisy sounds. A first idea towards reducing these phenomena was the employment of constraints in the depth of the offspring and the depth that the operators acted. To this end, the selected genetic operator was being re-applied while the offspring had a minimum of depth levels below 3 and above 10. Furthermore, the nodes that were available to be selected for the application of the operator, were the ones located at a depth level above 2.

The evolution of melodies was progressing slowly, given the deteriorated number of generations that a user is able to evolve during a trial. Therefore, the vast alteration of high-rated individuals would prolong the evolution, making the rating procedure more fatiguing for the user. Additionally, we considered the fact that the depth of the node that is chosen for the genetic operation has an impact

[1] The vector of relative differences, $\mathbf{r}$ between two vectors, $\mathbf{x_1}$ and $\mathbf{x_2}$ is computed as $\mathbf{r} = (\mathbf{x_2} - \mathbf{x_1})./\mathbf{x_1}$, were . / is the Hadamard or componentwise division. Every dimension of r describes the relative difference between the components of each dimension of $\mathbf{x_1}$ and $\mathbf{x_2}$. In the examined case the use of relative differences demises scaling issues that emerge by the inhomogeneous vector of features. For example, for the examined individuals the fractal dimension is between 1.1 and 1.9, while the spectral centroid is between 800 and 1,200.

Fig. 3 Mean and standard deviation of distances between the phenotypes of 10 individuals and their descendants from 1 to 10 generations. The descendants were created with **a–c** crossover and **b–d** mutation acting on a specified depth on each descendant



(a) Crossover mean

(b) Mutation mean

(c) Crossover St.D.

(d) Mutation St.D.

on the amount of variation of the melodies, i.e., the deeper the node the lesser the variation. Since a higher rated individual should be less altered, the depth on which the genetic operator acts should be larger. Considering a minimum depth for the genetic operation denoted as m, the depth of the tree representation of the ith individual, D_i, its fitness value f_i, a minimum fitness value equal to 0 and a maximum fitness value F, the depth of the operation, d_i, is chosen to be the integer rounding value of

$$\beta_i = m + \frac{f_i \cdot (D_i - m)}{F}, \tag{1}$$

thus $d_i = \lfloor \beta_i + 0.5 \rfloor$. Equation 1 admits a linear adaption of depth according to fitness which we have chosen as an initial approximation. Further research on the nature of adaption might provide better results.

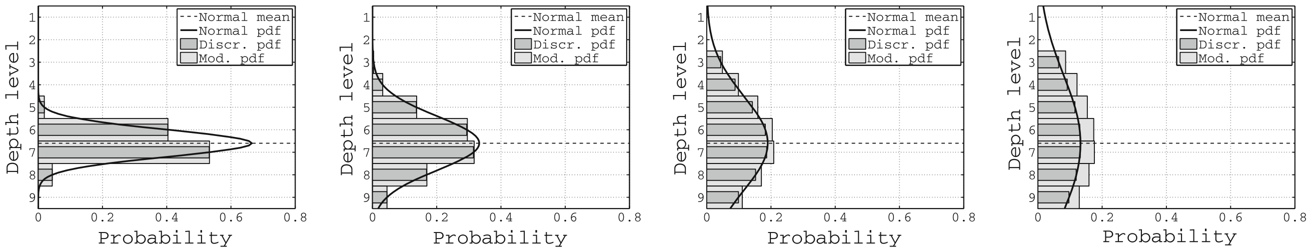
3.3 Controlling evolution

The rounded value of Eq. (1) defines a certain depth for the operation to act on each individual. A potential drawback of this approach is that the evolution becomes more deterministic, depriving the user of the effect of surprise. To this end, we have employed a non-deterministic parameter that introduces uncertainty to the depth that is chosen for the genetic operation. This parameter is adjusted by the user before the genetic operations apply to the selected individuals. We have named this parameter *Risk Factor* (RF), to point out that there is also a hazardous potential in the effect of surprise that high RF values impose.

The depth on which the operator acts depends on two factors: the depth according to fitness, as estimated in Eq. (1), and the value of the RF. Specifically, the RF value modifies the width of a Gaussian bell that constitutes a continuous normal probability density function (pdf) estimated by

$$p(x; \beta_i, \sigma(\mathrm{RF})^2) = \frac{1}{\sqrt{2\pi\sigma(\mathrm{RF})^2}} \mathrm{e}^{-\frac{(x-\beta_i)^2}{2\sigma(\mathrm{RF})^2}}, \tag{2}$$

where β_i is the depth according to fitness, given by Eq. (1), that acts as the mean of the distribution and $\sigma(\mathrm{RF})$ is a function of RF that acts as its variance. The pdf, $p(x; \beta_i, \sigma(\mathrm{RF})^2)$, is continuous, and provides probabilities for non-integer depths, x. We thus form the discrete equivalent of $p(x; \beta_i, \sigma(\mathrm{RF})^2)$ by computing the probabilities at each discrete depth level allowed by a minimum and a maximum depth of operation action, m and D_i, respectively. The discrete probabilities are finally adjusted to have unit sum, forming the final *modified pdf*. A graphical example of the probability transformations is given in Fig. 4, with $\sigma(\mathrm{RF}) = 3 \cdot \mathrm{RF}$.



(a) RF = 0.2 **(b)** RF = 0.4 **(c)** RF = 0.7 **(d)** RF = 1.0

Fig. 4 In this figure we assume an individual rated with 6, its tree representation has 9 depth levels and the minimum depth that a genetic operator can act is 3. The probabilities that a genetic operator will act on a certain depth according to a RF are depicted, specifically **a** RF = 0.2, **b** RF = 0.4, **c** RF = 0.7 and **d** RF = 1.0. The discrete depth probability is computed by the *modified pdf*

4 Overview of the proposed interactive evolution system

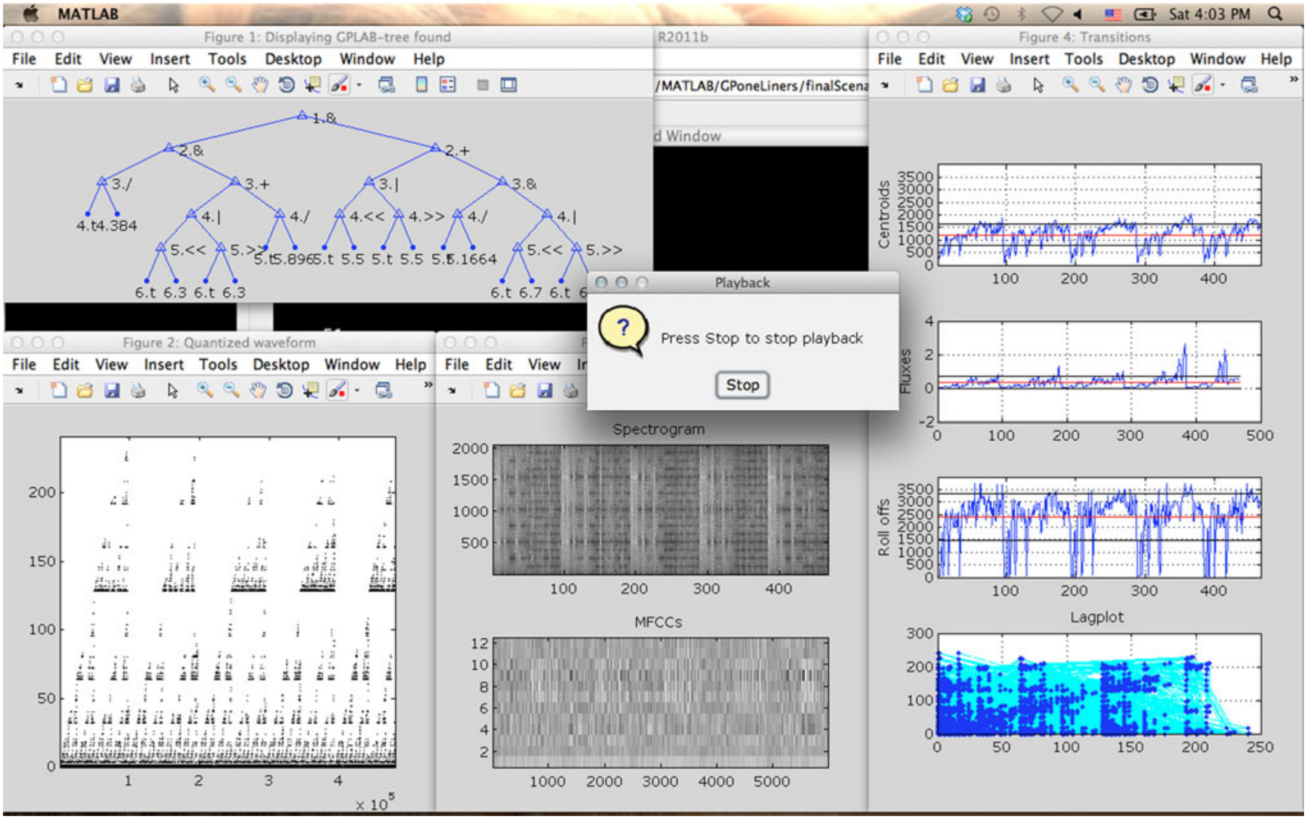
The proposed IE system incorporates GP to evolve the functional expressions discussed in Sect. 2. The user hears the sound output (phenotype) that an individual produces for as long as she/he wishes and then assigns a fitness value for this individual according to hers/his taste. Some individuals exhibit interesting melodic content, with several melodic and rhythmic variations, while others produce rather uninteresting music forms. Since the user may not be sure about the potential of each melody to change from uninteresting to interesting, she/he should spend a considerable amount of time hearing them, a fact that increases fatigue. For this reason, the user is advised to consult several visualizations that are provided in parallel with the sound playback so that the expected variation potential of the individual she/he hears is somehow anticipated. These visualizations include the spectrogram, the MFCCs visual representation and the plot of the quantized sequence ($q(t)$) among others. Figure 5 illustrates a screen shot of the visualizations that are produced during the playback of an individual.

A standard GP methodology is followed for the presented IE system, with the individuals of the current population going through a selection stage, where the parents of the next generation are specified. Two variations of the main IE system were created and tested, one with random selection of the depth of nodes and another that utilized the RF. For each variation, three versions were created with different selection schemes that are previously described (in Sect. 2), namely *roulette*, *tournament* and "*elitist*" methods. The motivation behind the "*elitist*" selection was to examine the possibility of further reducing the user fatigue by eliminating emerging offspring that produced rather noisy or uninteresting phenotype, with a subsequent tradeoff in population variability. The random depth selection is similar to the one presented in (Kaliakatsos-Papakostas et al. 2012).

In the RF variation, the users are asked to provide a RF value before the application of the genetic operators. The available RF values are $0, 0.1, \ldots, 1$, and the variation function was chosen to be $\sigma(\mathrm{RF}) = 3 \cdot \mathrm{RF}$. A value of RF near 1 allows an almost uniformly distributed selection of depth, as illustrated in Fig. 4d. The genetic operator was selected randomly with crossover and mutation probability equal to 0.9 and 0.1, respectively. The minimum depth that the genetic operation is allowed to act is set to $m = 3$. After experimenting with the genetic operators, we observed that individuals of extremely small and large depth tended to produce uninteresting and noisy sounds, respectively. For this reason, we employed depth constraints to the offspring by re-performing the selected operation with the selected individuals until the depth of their offspring was between 3 and 10.

We have firstly designed the system to create a random initial generation, but the individuals' phenotypes created in a random manner were most commonly uninteresting or noisy. A rating procedure with this kind of initialization would just discard noisy individuals, failing to produce interesting finding. The initial generation was thus chosen be a random selection of individuals which have been previously certified to produce interesting sonic output. In this way, the evolutionary process was led towards the subjectively chosen direction of the user.

Focusing on the presented IE system, each user was able to select the number of individuals in each generation. For the presented results, however, the users were only allowed to choose 4 individuals for the initial generation (and consequently for each generation), which constitutes a good compromise between population diversity and IE potential. Additionally, for the RF variation, each trial began with 0.5 RF value. The implementation of the IE system was developed in MATLAB using a modified version of "*GPLAB*" (Silva and Almeida 2003). A typical graphical user interface (GUI) was also implemented for the communication between the user and the system in order to facilitate the user.


MATLAB
Sat 4:03 PM
Figure 1: Displaying GPLAB-tree found
File Edit View Insert Tools Desktop Window Help
1.&
2.&
2.+
3./
3.+
3.|
3.&
4.t4.384
4.|
4./
4.<<
4.>>
4./
4.|
5.<<
5.>>
5.t5.8965.t 5.5 5.t 5.5 5.5.1664
5.<<
5.>>
6.t 6.3 6.t 6.3
6.t 6.7 6.t
R2011b
/MATLAB/GPoneLiners/finalScena
d Window
Figure 4: Transitions
File Edit View Insert Tools Desktop Window Help
Centroids
Fluxes
Roll offs
Lagplot
Figure 2: Quantized waveform
File Edit View Insert Tools Desktop Window Help
x 10^5
File Edit View
Spectrogram
MFCCs
Playback
Stop

Press Stop to stop playback

Fig. 5 Screen shot of the visualizations during the playback process

5 Experimental results

The derived results comprise statistics gathered among 11 participants-users, most of whom (7 out of 11) were able to play a musical instrument and 4 of them had at least 5 years of music education. The participants were not aware of the purposes of the research, and were not informed about the way that the system was realized before they started their trial. The only information they were provided with had to do with their interaction with the system through the GUI. The participants that used both the random depth and the RF variations were seven; two participated exclusively in the random depth variation[2] and two exclusively in the RF variation. Thus 9 users participated in each variation. The participants that took part in both variations first ran a trial of the random depth and one week latter, a trial with the RF variation. Before the beginning of each trial, each user heard three sample melodies to become familiarized with the music style and the sound textures of the 8-bit melodies.

The participants were advised to rate each melody according to their taste and were encouraged to feel free to quit the program any time they liked. For the participants that used the RF variation, they were previously informed that higher RF values were expected to introduce greater novelty to the new melodies, under the risk that these melodies could be uninteresting or noisy. The hearing process was controlled by the user, who was prompted to stop the melody any time she/he wished. Before the beginning of each trial, the users were also advised to consult the visualizations for determining the alteration potential of the melody they heard. After hearing each melody, a rating dialog appeared prompting the user to rate the melody just heard. The rating scale was the integer values between 0 and 10, with 0 being the worst, while the participants were advised to freely rate the individuals according to their personal taste. Between each generation, the users were asked to choose an RF value, according to their expectancy for melodic diversity.

5.1 Fitness improvement in both variations

We divided the participants in three groups for each variation so that all possible system conditions were considered. We gathered the fitness values assigned by the

[2] The participants that used the random depth variation were initially 3, but the results obtained by one of them were discarded because he quit the program after the first generation.

participants in the initial generation and the final generation and compared them to assess the improvement made in both variations, for all the system conditions. By studying the rating values provided by different users, we observed great differences in the way that users used the 0–10 rating scale. For example, the best rating given by a user was 5, while other users considered 7 as a moderate rating. In order to make the rating statistics independent of the rating profile of each participant and since we are interested in measuring the improvement of rating and not the rating per se, we have normalized the ratings for all generations of each user to have unit maximum value. This was realized with the division of all the ratings of a user with hers/his maximum rating among all individuals. Sample melodies at different evolution stages that appeared during the simulation of a participant are available online (2012) or upon request.

Tables 2 and 3 demonstrate the overall improvement of the normalized user ratings from the initial to the final generation for the random depth and the RF variations, respectively. We refer to the normalized fitness ratings corresponding to the initial and the final generation with the FI and FL indices, respectively, while the GN index refers to the number generations. The mean value of the aforementioned quantities is denoted by μ, their standard deviation by σ, while max and min denote their maximum and minimum values, respectively. The relative mean fitness change between the initial to the final generation is denoted by r_f, where $r_f = (\mu_{\mathrm{FL}} - \mu_{\mathrm{FI}})/\mu_{\mathrm{FI}}$.

The positive r_f values for each variation and each version demonstrate that the mean fitness value increased from the initial to the final generation. This reveals that the proposed system captures, to some extent, the subjective aesthetics of the user. As indicated by the r_f values in both variations, the depth constraints for the genetic operation action imposed by the RF, are expected to produce better results. This improvement is also captured in Fig. 6. Additionally, Fig. 7 depicts box plots of the accomplished relative improvements in participants' ratings, for the random and the RF variations. To evaluate whether their differences are statistically significant, we apply a two-sided Wilcoxon (1945) rank-sum test between their relative ratings improvement. The null hypothesis in the test is that the samples compared are independent samples from identical continuous distributions with equal medians. The null hypothesis is rejected at the 5 % significance level with a p value equal to 0.0224, showing that the differences are statistically significant.

Moreover, the mean number of generations increased with the RF variation, which reveals that the effect of fatigue is possibly reduced. In turn, this probably exposes the user's sense of control in the evolutionary process. The standard deviation of the normalized ratings in the final generation is decreased in relation to the initial generation in all cases except the RF-Tournament version, which indicates that the fitness values of most individuals in the final generation are expected to be closer than the ones in the initial generation.

5.2 Risk factor values and fitness rating behaviors

The participants in the RF variation had different altitudes with the use of the RF. Figure 8 exhibits the RF values provided by each of the 9 participants who used the RF

Table 2 Results for the improvement of ratings between the initial and final generations for all three versions of the random depth variation

	μ_{GN}	Random depth								r_f
		Initial generation				Final generation				
		$\mathrm{Min}_{\mathrm{FI}}$	μ_{FI}	σ_{FI}	$\mathrm{Max}_{\mathrm{FI}}$	$\mathrm{Min}_{\mathrm{FL}}$	μ_{FL}	σ_{FL}	$\mathrm{Max}_{\mathrm{FL}}$	
"Elitist"	5.5	0	0.431	0.269	0.778	0.222	0.611	0.272	1	0.419
Roulette	6	0	0.522	0.313	1	0.167	0.647	0.272	1	0.239
Tournament	9	0	0.333	0.319	0.875	0.125	0.510	0.157	0.7	0.5312

Table 3 Results for the improvement of ratings between the initial and final generations for all three versions of the RF variation

	μ_{GN}	Fitness-adaptive with risk factor								r_f
		Initial generation				Final generation				
		$\mathrm{Min}_{\mathrm{FI}}$	μ_{FI}	σ_{FI}	$\mathrm{Max}_{\mathrm{FI}}$	$\mathrm{Min}_{\mathrm{FL}}$	μ_{FL}	σ_{FL}	$\mathrm{Max}_{\mathrm{FL}}$	
"Elitist"	9	0	0.208	0.202	0.600	0.400	0.858	0.168	1	**3.120**
Roulette	12.33	0	0.392	0.291	0.800	0.400	0.725	0.196	1	**0.851**
Tournament	8	0	0.391	0.240	0.750	0.111	0.716	0.290	1	**0.831**

variation, together with the normalized mean fitness values of the generation evolved with the respective RF. The mean fitness values were normalized to maximum unit value for convenient visualization. The ratings of the initial generation were omitted for demonstrational purposes. A noticeable behavior about the RF adjustment and the ratings that followed concerns the users in the elitist selection version, and User 1 in the roulette version. These trials have also produced the highest relative improvements between initial generation and final generation. The users in these trials have experimented with different RF values until the final generations, where they gradually reduced the RF. The RF reduction was accompanied by an increase in the mean fitness value, which indicates that the final few generations were continuously producing more pleasant sound output.

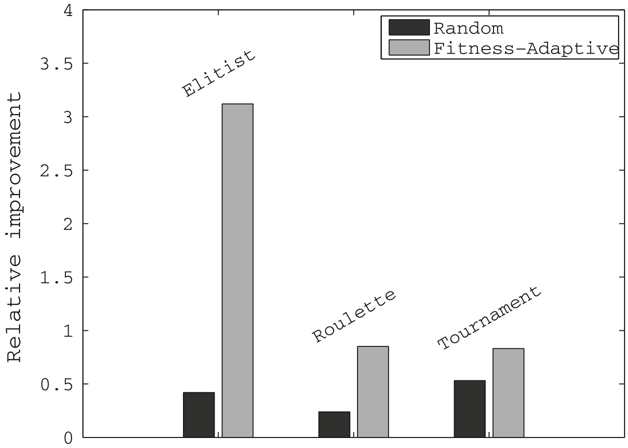


Fig. 6 Relative improvement (r_f) between the initial and the final generation for the random depth and the RF variations

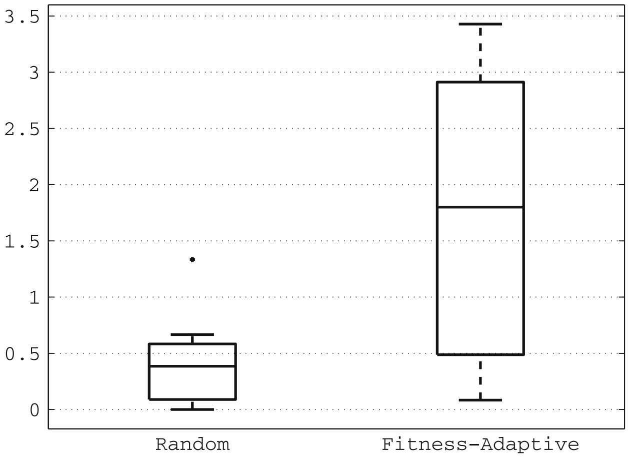
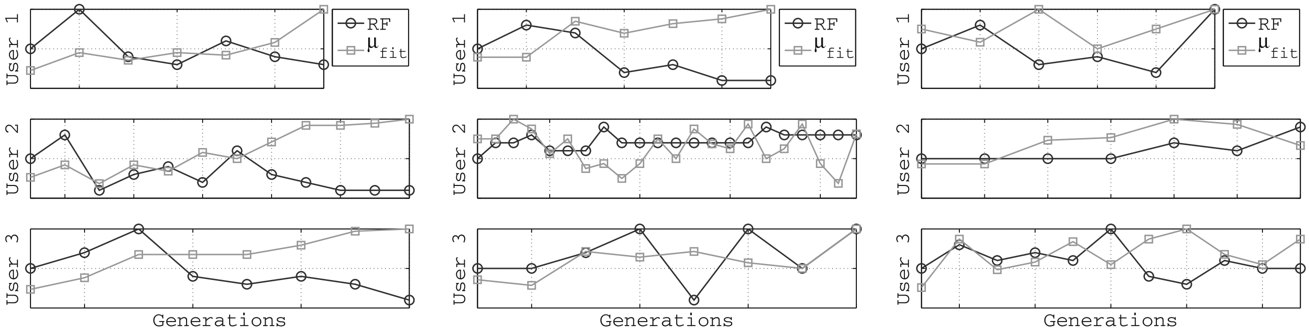


Fig. 7 *Box plot* of the relative improvement of ratings with the random and RF variation

The RF and rating altitude for the rest of the participants exhibits that the impact of the RF in the evolution was imperceptible, probably making the evolution more "random" than they expected. Nevertheless, the RF adjustments by some users indicate that there was a misconception about the role of the RF. For example, User 2 in the roulette version did not apparently take under any consideration the RF, since no adjustments were made with the appearance of better or worse generations. An interesting remark has to do with the fact that all the users of the elitist version had the same RF adjustment attitude, which also led to a vast relative improvement in the fitness ratings. On the other hand, for the random depth variation and the elitist selection method, the relative improvement was not so impressive. Even though the number of participants is very small, indications are provided that the combination of the RF with an elitist selection scheme could lead to more productive interactive evolution of the presented melodies.

6 Conclusions

This paper presents an IE system which evolves melodies using GP with fitness-adaptive genetic operators, allowing



(a) "Elitist" **(b)** Roulette **(c)** Tournament

Fig. 8 RF values and mean ratings per generation ($\mu_{\mathtt{fit}}$) for each participant in the RF variation. The initial ratings are omitted for demonstrational purposes

the user to have some control over the randomness of evolution. Initially, experiments reveal a connection between the depth of the tree representation that a genetic operator acts and the phenotypical distance of the sounds produced by the presented system. These experiments have led us to employ a randomness parameter called "Risk Factor" (RF), which is adjusted by the user before the formulation of each generation. Higher RF values are expected to drive the evolution towards individuals that share greater differences from their ancestors. Subjective tests have been conducted and provided indications that the proposed approach generally facilitates the user towards evolving melodies that are more pleasing. In turn, the presented IE system with the use of the RF is expected to introduce a more productive and less fatiguing experience to the user.

Although the subjective tests were performed on a rather small sample of participants, there are statistically significant differences between the rating improvements in the classical GP and the proposed RF variation. Thus, the results are indicative about the improvement of the system with the use of the RF. Furthermore, there are indications that an elitist selection scheme may be more proper for the presented system, when combined with the control of evolution randomness proposed in this paper. This may result from the fact that the user employs the amount of uncertainty that she/he wishes only to the best-rated individuals. Thus the evolution is accelerated, reducing user fatigue and improving the overall experience. The random "intrusion" of a poorly rated individual, a fact that is possible with roulette and tournament selection, may have a negative impact on the sense of control that the user acquires with the RF.

As a future work, we initially intend to create a web platform for all variations and versions of this IE system in order to make it accessible to many participants. Thus, the availability of results will increase and we will be able to reach safer conclusions about the appropriateness of the RF variation. Moreover, the same methodology should also be tested on other IE systems that use GP to evolve music, sound or any form of art. Controlling the randomness in IE systems that create art might be of vital importance for the potential output that this system may have. Making such systems more user-friendly may not only boost the research towards human conception of sound and art, but also disseminate computer-aided musical composition to a wider spectrum of people.

Acknowledgments The authors would like to thank the anonymous reviewers for their valuable comments and ideas that helped to improve and extend the content as well as the clarity of this paper. We would also like to thank the participants who voluntarily participated in this research.

References

Angeline PJ (1997) Subtree crossover: building block engine or macromutation? In: Genetic programming 1997 proceedings of the second annual conference, Morgan Kaufmann, San Francisco, pp 9–17

Burton AR, Vladimirova T (1999) Generation of musical sequences with genetic techniques. Comput Music J 23(4):59–73

Davis S, Mermelstein P (1980) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Process 28(4):357–366

Garcia RA (2000) Towards the automatic generation of sound synthesis techniques: preparatory steps. In: Audio engineering society convention 109

Garcia RA (2001) Growing sound synthesizers using evolutionary methods. Synthesis M(Garcia 2000):99–107

Heikkilä V (2011) countercomplex-bitwise creations in a pre-apocalyptic world. http://countercomplex.blogspot.com/

Heikkilä V (2011) Discovering novel computer music techniques by exploring the space of short computer programs. CoRR abs/1112.1368

Higuchi T (1988) Approach to an irregular time series on the basis of the fractal theory. Phys D 31:277–283

Kaliakatsos-Papakostas MA, Epitropakis MG, Floros A, Vrahatis MN (2012) Interactive evolution of 8-bit melodies with genetic programming towards finding aesthetic measures for sound. In: Proceedings of the 1st international conference on evolutionary and biologically inspired music, sound, art and design, EvoMUSART 2012, Malaga, Spain, LNCS, vol 7247, pp 140–151. Springer

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection, 1 edn. MIT Press, Cambridge, MA

Lewis M (2008) Evolutionary visual art and design. In: Romero J, Machado P (eds) The art of artificial evolution, natural computing series, Springer, Berlin, pp 3–37

Manaris B, Roos P, Machado P, Krehbiel D, Pellicoro L, Romero J (2007) A corpus-based hybrid approach to music analysis and composition. In: Proceedings of the 22nd national conference on artificial intelligence. AAAI Press, vol 1, pp 839–845

Phon-Amnuaisuk S, Law E, Kuan H (2007) Evolving music generation with SOM-fitness genetic programming. In: Giacobini M (ed) Applications of evolutionary computing. Lecture Notes in Computer Science. Springer, Berlin, vol 4448, pp 557–566

Poli R, Langdon WB, McPhee NF (2008) A field guide to genetic programming. Lulu Enterprises, UK

Putnam JB (1994) Genetic programming of music

Sample of melodies—individuals created during a trial by a participant. http://sites.google.com/site/maximoskp/SampleMelodies.zip (2012)

Shannon CE (2001) A mathematical theory of communication. ACM SIGMOBILE Mobile Comput Commun Rev 5:3–55

Silva S, Almeida J (2003) Gplab—a genetic programming toolbox for matlab. In: Proceedings of the Nordic MATLAB conference, pp 273–278

Spector L, Alpern A (1994) Criticism, culture, and the automatic generation of artworks. In: Proceedings of the twelfth national conference on artificial intelligence, vol. 1. AAAI '94, American Association for Artificial Intelligence, Menlo Park, CA, USA, pp 3–8

Spector L, Alpern A (1995) Induction and recapitulation of deep musical structure. In: Proceedings of the IFCAI-95 workshop on artificial intelligence and music, pp 41–48

Terasawa H, Slaney M, Berger J (2005) Perceptual distance in timbre space. In: Proceedings of the 11th international conference on auditory display (ICAD2005), Department of Computer Science and Information Systems, University of Limerick, Limerick, Ireland, pp. 61–68

Tokui N (2000) Music composition with interactive evolutionary computation. Communication 17(2):215–226

Tzanetakis G, Cook P (2002) Musical genre classification of audio signals. IEEE Trans Speech Audio Process 10(5):293–302

Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83

Yan JR, Min Y (2011) User fatigue in interactive evolutionary computation. Appl Mech Mater 48–49:1333–1336

Ziv J, Lempel A (1977) A universal algorithm for sequential data compression. IEEE Trans Inf Theory 23(3):337–343