Spieldose: An Interactive Genetic Software for Assisting to Music Composition Tasks

Ángel Sánchez, Juan José Pantrigo, Jesús Virseda, and Gabriela Pérez

Departamento de Ciencias de la Computación
Universidad Rey Juan Carlos, C/Tulipán, s/n,
28933 Móstoles, Madrid, Spain
{angel.sanchez,juanjose.pantrigo,j.virseda,gabriela.perez}@urjc.es

Abstract. We describe a new software tool, called Spieldose (in English, musical box), suited to the automatic music composition task. Our system is based on the paradigm of Interactive Genetic Algorithms (abbreviated as Interactive GA) where the parent selection stage in a typical GA is made by the user according to his/her subjective criteria. The tool permits to integrate the interaction between the system and the potential users when they create their melodies. One important contribution of this work is the proposal of specific musical genetic operators (different types of crossover, mutation and improvement operators) which ensure that the generated melodies are in concordance with Music Theory and they are also nice to listening. Moreover, our software tool can be customized to a particular musical style by including the specific musical knowledge domain in the system. For validation purposes, we used Spieldose to compose different pieces corresponding to the classicism.

1 Introduction

In general, Computer Music is related to the theory and application of different techniques to the musical generation (or composition) with the aid of computers. Musical information analysis from different sources (like digital audio, digital partitures or metadata) is also a component area in Computer Music. Therefore, it is a multidisciplinary field related to disciplines like Digital Signal Processing, Artificial Intelligence, Acoustics, Mathematics or Image Processing, among others.

A good survey on automatic music composition is presented by López de Mantaras and Arcos [1]. This paper describes a set of representative Computer Music systems (related to compositional, improvisation and performance aspects) that use AI techniques. The pioneering work in automatic music composition is due to Hiller and Isaacson in 1958. Many fundamental compositional processes in music may be described as taken an existing musical idea and changing it to produce a new related piece [2]. Modern musical composers follow many times some simple rules during their creative process. These rules refer to the intervals between notes, to the notes used in each tone and in each musical style and to the types of rhythmic articulations [3].

J. Mira and J.R. Álvarez (Eds.): IWINAC 2007, Part I, LNCS 4527, pp. 617–626, 2007.

In general, music composition can be a very complex task for the people without musical knowledge or skill. To overcome this difficulty, different automatic music composition systems have been proposed since the appearance of computers [4].

Several computational intelligence methods have been applied to music synthesis and analysis tasks. Neural Networks (NN), Genetic Algorithms (GA), and Genetic Programming (GP) are the main ones. A survey of the application of Artificial Intelligence (AI) methods to music generation can be found in [5]. Genetic and evolutionary algorithms have demonstrated to be an effective method to find solutions in complex search spaces. Marques et al [3] have applied GA to the generation of musical sequences using melodic and musical theory concepts. Khalifa and Foster [6] proposed a composition system in two stages: first, musical sound patterns are identified and then they are combined in a suitable way. The MusicBlox project [4] is based on small music fragments or blocks which are combined using GA producing musical successful results. The combination of GA and NN (in particular, Multilayer Perceptrons) for music composition has also been exploited in the work by Göksu et al [7]. Some other AI techniques like constraint programming provide a suitable tool for automatic music generation. Henz et al [8] developed an experimental platform called COMPOzE for intention-based composition. Jewell et al [9] have described the architecture of an agent-based distributed system which is oriented to musical composition.

This paper presents an interactive music composition system based on the application of Genetic Algorithms (GA) to assist the user in this complex task. Our work takes into account the fact that when several people listen to the same musical piece, their impressions are not necessarily the same. In a related work, Unehara et al [10][11] have also remarked that: “any music generation system must reflect composers subjectivity towards music”. To achieve this goal, these authors have proposed the application of Interactive GA to music composition. In these algorithms the parent selection stage is performed the human. In this way, our adapted GA is used as an optimization method to generate and evolve a population of musical pieces (the individuals in a standard GA) by the application of some specific genetic operators which are introduced to hold the principles of Musical Theory. At each iteration during the genetic evolution, the user selects several melodies or pieces considered as “good” candidates according to musical subjective criteria [10]. In this way, an initial population of automatically generated compositions is evolved until a termination condition is met. These musical works, which can comply with a given musical style or author, are generated by considering some specific compositional criteria. The developed software tool, called Spieldose, aims to include criteria of musical composition into the Interactive GA. This integration is the core of the proposed work and it will be detailed in successive sections.

The rest od the paper is organized as follows. Section 2 offers a global description of the presented musical composition system. This description is focused on the detailed presentation of our specific Interactive GA and on the application Graphical User Interface (GUI). The different components of our Interactive GA

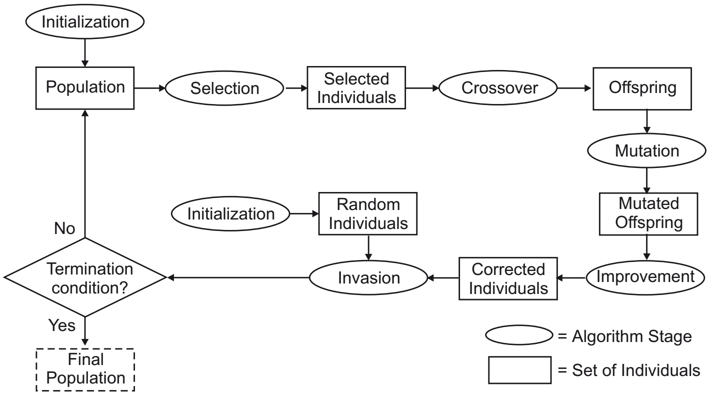


Fig. 1. Overview of the proposed Interactive Genetic Algorithm

are detailed in Section 3. Experimental results are resumed in section 4. Section 5 outlines the conclusions and provides future research lines.

2 System Overview

Our approach is focused on an specific Interactive GA for the automatic music composition task. These kind of GAs were initially applied to fields like the industrial design where the fitness functions were difficult to be defined. We have followed a similar approach as presented in [10][11] by Unehara and Osinawa. Spieldose is a prototype tool which extends the work of these authors. We created new types of genetic operators, in particular, new mechanisms for crossover, mutation, improvement and invasion operations. A complete tool GUI aimed to assist both unskilled and expert user in the music composition stages has also been developed. Our proposed Interactive GA includes the stages represented by the Figure 1:

- *Initialization*: The set of individuals that form the initial GA population are generated in this stage. This operation is defined to produce an initial set of musical works or melodies that are created using some specific knowledge from music theory. In our application, we have only produced 8-bar length musical melodies.
- *Selection*: The best melodies (individuals) in the population are selected by a human expert at each iteration of the GA. In this stage, a variable-size subset of melodies is chosen by the user (by listening them one by one) considering his/her musical preferences or guided by the characteristics of a predetermined musical style.
- *Crossover*: Those individuals selected are combined using three possible crossover mechanisms (according to crossover probabilities), to produce a new generation of child melodies (offspring).

- *Mutation*: This operation enables the modification of some fragments (chromosomes) in the new generated melodies according to a mutation probability.
- *Improvement*: It permits the automatic correction of some musical errors which could be caused by the previous operators. This stage is automatically performed and It strongly takes into account the considered criteria of musical theory. It can also consider the feature of a particular musical style.
- *Invasion*: This stage is included to add new randomly generated individuals to the population of musical works. It is needed to avoid the loss of diversity in the collection of melodies after a number of evolution iterations in the Interactive GA.

In our framework, each individual is an 8-bar melody which is codified using the differential or relative notation as described in [12]. A melody is represented by a sequence of notes, where each note has its pitch, length and type attributes (as shown in Figure 2)). The corresponding data structure to represent a melody is a vector where each position has two fields: the first one describes the note pitch expressed in half tones (where a zero value represents a ligature of the same pitch), and the second one implicitly codifies the note (or silence) length using three possible capital letters: *'N'* represents a note, *'S'* represents a silence and *'L'* stands for a note (or silence) ligature. Some complementary remarks are now pointed out:

- Note types are also implicitly represented and each one holds a determined number of vector positions equals to its length using as minimal reference unit a semiquaver note (for example a black note requires four vector positions, a quaver note requires two vector positions, and so on).
- Each note pitch is expressed with respect to the previous note except for the first note of the melody (represented by an absolute number of half tones with respect to a reference octave). The pitch is given by a positive or negative number of half tones required to obtain the note sound from the previous note in the piece.
- In order to simplify the implementation of GA operators, we set the size of all individuals (melodies) of the population to 128. This value corresponds to 8-bar pieces at 4/4 measures, where the minimal considered note length is a semiquaver. Therefore, length of a note is determined by adding one to the number of consecutive *'L'* vector positions preceded by a *'N'* vector position.

One important contribution of this works is the variety and its effective implementation of the operators in the Interactive GA. We have properly combined in this work the knowledge of experts in Musical Theory and experts in Combinatorial Optimization methods. The following section describes in detail each of the involved GA stages and their components. Another contribution of this work is the complete graphical user interface (GUI) of Spieldose that offers the user the appropriate functionality for the musical composition task and also the

4	0	0	0	1	0	2	0	0	0	0	0	4	0	0	0
N	L	L	L	N	L	N	L	S	L	L	L	N	L	L	L

Fig. 2. Example of the codification of the first bar in a given melody

GUI hides the implementation details of the interactive GA. Figure 3 shows the main GUI window of Spieldose (left) and the initialization stage window (right). This GUI offers the user the following main options:

- Initialize a population of melodies (both initial and invader populations).
- Interactively listen to the created musical works in order to choose those ones that are considered the best ones according to the user preferences.
- Select the best subset of melodies using a tournament algorithm.
- Save the favorite melodies at each iteration in Waveform Audio Format (wav) and/or in text format (tex).
- Interactively edit a musical piece in text format such that the corresponding audio is also modified at the same time (there exists a complete updated equivalence between both formats for each melody).
- Modify different features of the proposed Interactive GA, such as the crossover and mutation mechanisms, the population size, etc.

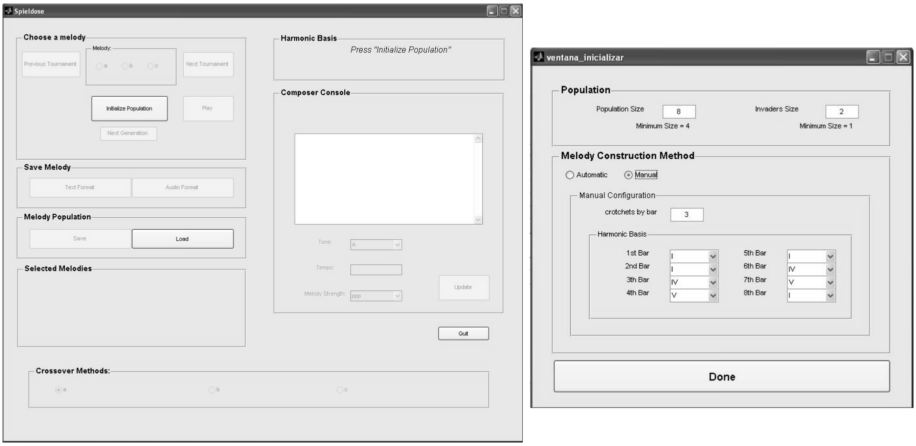
3 Main Components of the Interactive GA

This section describes the main components and the GA operators of our algorithm, as represented in Figure 1.

3.1 Algorithm Initialization

For the generation of the initial set of melodies (individuals), we used the knowledge of the experts in Musical Theory. Two main substages have been considered for this goal: (a) initialization of the common features corresponding to all individuals in the population and (b) initialization of particular aspects from each specific individual. All the individuals share the same time signature or rhythms and also the harmonic structure in order to simplify the application of the GA operators on the melodies. Of course, individuals differ in the melody they represent (their chromosome structure). The common information to all the individuals is generated as follows.

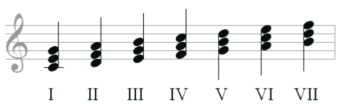
- The permitted rhythms are 2/4, 3/4 and 4/4, and they can be set manually or random.
- The considered harmonic structure assigns a fundamental chord to each of the melody bars in the musical piece. This structure determines that



(a) (b)

Fig. 3. A view of Spieldose application GUI

a melody is correctly created and also that it is human listening. Figure 4(a) represents a octave of three-note chords. For example, if tonic (or first) note is C in major-chord, then this chord I is composed by the notes: C, E and G, respectively. Figure 4(b) describes the considered tonal possibilities for each of the 8 bars in the harmonic basis of the melodies. The first bar always starts with a chord I (and it also represents the major/minor tonality of the melody). Note that the bar positions in the second and third rows with respect to the corresponding ones in the first row of Figure 4(b) means that these chords in a given bar can be exchanged.



I	I	IV	V	I	IV	V	I
	III	II	VII	III	II	VII	
	IV	VI		IV	VI		

(a) (b)

Fig. 4. Solution initialization: the harmonic structure

As pointed out, each individual in the GA stores an 8 bars melody. Next, ee describe the stages followed by the system to create a musical piece.

– *Rhythmic structure*: First, the melody rhythm is constructed. It consists in a vector of notes as represented by the Figure 2. For this purpose, three possible values can be assigned to the second component in each vector position: *'N'*, *'S'* and *'L'*. The first position in a bar of this 2-tuple vector is assigned a value *'N'* to ensure the independence of each bar in the melody

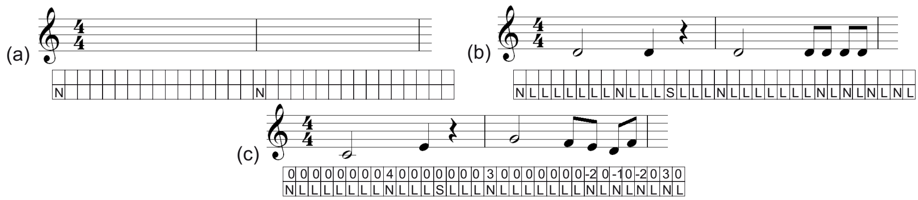


Fig. 5. Sketch of an example of the automatic creation of a melody using the Spieldose tool (for shortness, the first two bars are only represented)

thus discarding the ligatures between notes of different bars (Figure 5(a)). Next, the rest of vector positions are completed by raffling the three possible values (*'N'*, *'S'* and *'L'*) as shown in Figure 5(b). These values can have different probabilities of occurrence.

– *Pitch sequence*: The corresponding pitch values are assigned to the notes in the rhythmic sequence (see Figure 5(c)). These values are subsequently added to the melody such that they are chosen with a probability proportional to an assigned pitch weight. This weight is computed according to the considered tonality and following the compositional rules of a concrete musical style (in our framework we have focussed on the classicism). Therefore, those notes belonging to a considered triad chord have a higher probability to be chosen, and later those notes with pleasant dissonances (in particular, the seventh and ninth notes of the chord). The probabilities associated to each tonality jump are predefined and they have been suggested by the experts in Music Theory.

3.2 Parent Selection Scheme

This stage in our Interactive GA is performed by a tournament method and it is the only operation than necessarily requires from the human participation. A number of N different melodies automatically generated by the system are presented in groups of M (where $M < N$). Then, after listening the M melodies, the user chooses one in each group according to his/her musical preferences. These selected pieces will be the parents for the next generation of individuals, and the rest of the melodies are discarded. The tournament is a very appropriate mechanism for implementing the selection stage in an Interactive GA since the user is only responsible to select those individuals are the best ones in the actual population. This task, as performed manually, requires that user listen to all the generated musical pieces. However, it results much easier than assigning a fitness value to each individual. While performing this stage, the user can save those melodies considered as "good". Moreover, there also exists the possibility to edit and modify on-line the created melodies. This way, the user can interactively transform some parts or aspects of the musical pieces.

3.3 Crossover

The crossover operators aims to define new melodies using chromosomes from two or more parents with a crossover probability. In this way, a child melody is constructed by copying segments of chromosomes from their parent melodies. We have implemented three different types of crossover operators:

- Type A: Two new melodies are created from two parents. For each of the two selected 8-bar musical works, we determine two crossing points (one in each parent), and all the bars for the two child individuals are generated by exchanging the corresponding chromosomes of both parents.
- Type B: In this case, two new melodies are also created from two parents. However, several (in general, more than two) crossing positions are selected at different places in the parent melodies, and the two children are formed by the alternate selection of chromosomes from both parents.
- Type C: This is a generalization of crossover type B that is extended to a large number N ($N > 2$) of parents. As a result, N child melodies are created by selecting several crossing points in the parents and, in a similar way, we alternate the selection of parent chromosomes to build the N children.

It is important to remark that when using all types of crossover operators, the new resulting melodies are automatically repaired to preserve their corresponding tonality.

3.4 Mutation

It is possible to apply different types of mutation operators to the chromosomes of the individuals. We have implemented the following ones: (a) rotation of the notes in a segment (sequence of notes) of a melody, (b) melodic inversion of a segment and (c) the variation of a note (in its length and/or pitch). A given melody can mutate its chromosomes according to one of these three possibilities with a given mutation probability.

3.5 Improvement

This is one of the most interesting contributions of our Interactive GA. In the improvement stage, it is possible to use different heuristics to model the expert's musical knowledge domain with the aim to create musical works that are correct respect to musical theory and also pleasant to be listened. As an example of these improvements, we correct the large pitch jumps between notes of different bars to reduce these differences to less than one octave.

3.6 Invasion

This operator also generates new melodies by a similar procedure of the initialization stage. These new created individuals are added to the actual population.

The aim of this operation is to prevent the loss of diversity and also to avoid a premature convergence of the proposed Interactive GA after a small number of iterations.

4 Experimental Results

Due to the subjectivity of each particular human user when composing musical pieces with Spieldose, it is not very appropriate to show quantitative results in this work. A way to demonstrate the validity of our proposal, is creating a repository of musical pieces generated by Spieldose. This can be found in WAV format at the following URL: `http://gavab.escet.urjc.es/recursos.html`. All the pieces correspond to the classicism musical style. These 8-bar melodies have been produced by applying the different crossover and mutation operators considered by our Interactive GA. Spieldose source code in Matlab and the application tool can also be downloaded from the previous URL.

5 Conclusion and Future Work

This paper has presented a software tool called Spieldose for assisting the human user in the automatic music composition task. Our approach is based on the paradigm of Interactive GA where the parent selection stage is directly performed by the user. The application permits to integrate the interaction between the system and users when they compose their own musical pieces. This principle is useful both for experts in Music Theory and also for musically unskilled people. The aim when developing this software was also to show the natural integration of automatic music composition tasks into The GA. One main contribution is the proposal of new genetic operators (in special, different types of crossover, mutation and improvement operators) in the music context, to guarantee that composed melodies respect the principles of Music Theory and they are also pleasant to listening. This software can be customized to a particular musical style by including in it the specific musical knowledge domain. In particular, in order to validate our approach, we have used Spieldose to create different 8-bar pieces emulating the classicism style. The developed GUI of Spieldose offers to an unskilled music user the appropriate functionality for the compositional tasks. It also enables an immediate equivalence between the textual and audio formats of the musical work being composed.

As future work, we propose two types of improvements for Spieldose: those referred to the Interactive GA itself, and those related to the application interface. With respect to the first type, we intend to develop a system of modular GA components which enable the user to configure his/her own Interactive GA. As an example of this feature, new types of fitness functions could be included for both unskilled and expert users in the musical field. These functions would provide different weighted fitness criteria combining the user subjectivity and the objective musical quality for melody being composed. With respect to the application GUI, we plan to embed into the tool a musical editor which would display in musical notation a composed piece.

Acknowledgements

This research has been partially supported the grant URJC-TIC-C51-1, from Universidad Rey Juan Carlos (URJC) and Comunidad de Madrid (CAM), Spain.

References

1. López de Mantaras R., Arcos, J.L. AI and Music: From composition to expressive performance. AI Magazine, vol. 23, no. 3, pp. 43-57, 2002.
2. Gartland-Jones, A., Copley, P. The Suitability of Genetic Algorithms for Musical Composition. Contemporary Music Review vol. 22:3, 43-55, 2003.
3. Marques, M., Oliveira, V., Vieira, S., Rosa, A.C. Music composition using genetic evolutionary algorithms. Proceedings of the 2000 Congress on Evolutionary Computation, Vol 1, pp. 714-719, 2000.
4. Gartland-Jones, A. MusicBlox: A Real-Time Algorithmic Composition System Incorporating a Distributed Interactive Genetic Algorithm. LNCS 2611, pp: 490-501, 2003.
5. Burton, A.R., Vladimirova, T. Generation of Musical Sequences with Genetic Techniques. Comput. Music J., vol 23 pp:59-73, 1999.
6. Khalifa, Y., Foster, R. A Two-Stage Autonomous Evolutionary Music Composer. LNCS 3907 pp: 717-721, 2006
7. Hüseyin Göksu, Paul Pigg and Vikas Dixit. Music Composition Using Genetic Algorithms (GA) and Multilayer Perceptrons (MLP). LNCS 3612, pp:1242-1250, 2005.
8. Henz, M., Lauer, S., Zimmermann, D. COMPOzE-intention-based music composition through constraint programming. Proceedings Eighth IEEE International Conference on Tools with Artificial Intelligence. pp:118-121, 1996.
9. Michael O. Jewell, Lee Middleton, Mark S. Nixon, Adam Prgel-Bennett and Sylvia C. Wong. A Distributed Approach to Musical Composition. LNCS 3683, pp: 642-648, 2005
10. Unehara, M. Onisawa, T. Music composition system based on subjective evaluation. IEEE International Conference on Systems, Man and Cybernetics Vol. 1 pp. 980-986, 2003.
11. Unehara, M., Onisawa, T. Construction of Music Composition System with Interactive Genetic Algorithm. Proc. of 6th Asian Design Int. Conf. 2003.
12. Cruz-Alcázar, P.P., Vidal-Ruiz, E. Learning Regular Grammars to Model Musical Style: Comparing Different Coding Schemes. Lecture Notes in Computer Science, Vol. 1433/1998 pp. 211-222,1889.