Memetic Music Composition

Enrique Muñoz, *Senior Member, IEEE,* Giovanni Acampora, *Senior Member, IEEE,*
Jose Manuel Cadenas, *Senior Member, IEEE,* Yew Soon Ong, *Senior Member, IEEE*

***Abstract*—Since their birth, computers and artificial intelligence have always played a key role in the production of artworks through the designing of synthetic agents able to reproduce the capabilities of human artists in assembling high quality artefacts as paintings, sculptures and so on. In this scenario, music composition represents one of the key art disciplines that can greatly benefits from the appropriate use of computational intelligence, as witnessed by the large number of research activities performed in this field over the recent years. Nevertheless, the automatic composition of music is far from completely and precisely perfected due to the intrinsic virtuosity that characterizes human musicians' capabilities. In this paper we try to reduce this gap with the proposal of an intelligent scheme for efficient composition of melodies based on a musical method that is inspired by and strongly characterized by human virtuosity: the unfigured bass technique. In particular, we formulate the music composition technique as an optimization problem and solve it with an adaptive multi-agent memetic approach comprising diverse metaheuristics, the composer agents, that cooperate to create high quality four voice pieces of music starting from a bass line as input. A collection of experimental studies on the famous Bach's 4-voice chorales showed that the cooperation among different optimization strategies yields improved performance over the solutions obtained by conventional and hybrid evolutionary algorithms.**

***Index Terms*—Adaptive Memetic Algorithms, Multiagent Systems, Automatic Music Composition.**

I. Introduction

HISTORICALLY, artificial and computational intelligence methodologies have commonly been employed to support artists/scientists in their production of artworks. Some of the notable efforts involved the design of synthetic agents targeted at reproducing the capabilities of human artists in assembling high quality artefacts such as paintings, sculptures and so on. This technology, known as *Computer Art*, represented a real historical breakthrough in computer applications since it tried to replicate, for the first time, a pure and exclusive human capability: *the creativity*. Since 1965, when the first computer art exhibitions were held simultaneously in New York and Stuttgart, many strides have been made in the realization of the so-called creative algorithms capable of supporting (or replacing) an artist in his composition activities.

E. Muñoz is with the Dipartimento di Informatica, Universit degli Studi di Milano, Via Bramante 65, 26013 Crema, Italy e-mail: enrique.munoz@unimi.it.

G. Acampora is with the School of Science and Technology, Nottingham Trent University, NG11 8NS, Clifton Lane, Nottingham, United Kingdom. e-mail: giovanni.acampora@ntu.ac.uk.

J. M. Cadenas is with Department of Information Engineering and Communications, University of Murcia, 30100 Murcia, Spain. e-mail: jcadenas@um.es.

Y. S. Ong is with the School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore e-mail: asysong@ntu.edu.sg.

Manuscript received April 19, 2005; revised December 27, 2012.

Today, artificial intelligence literature covers a large spectrum of approaches for artworks assembly and, among them, the music represents the art disciplines that can get great benefits from the use of intelligent methodologies for data processing, as witnessed by the large number of research activities performed in this field during the last decades (see Section II-B). Indeed, it is possible to find research works dating back to well before the computer era aimed at solving different musical problems such as notation programs, sound synthesis, real-time performances, digital instruments, and much more [1]. However, among these music problems, the *Algorithmic (or Automatic) Music Composition* is, surely, what is most fascinating to both computer scientists and musicians. Indeed, different computer techniques have been applied to this challenging topic: random numbers, formal grammars, cellular automata, fractals and so on. In particular, starting from the pioneering research work of Lawrence J. Fogel on *Evolutionary Art* [2], recently, the term *Evolutionary Music* has been coined as a new form of computer art methodology that focuses on evaluating and evolving the aesthetics of a given musical artefact by means of evolutionary algorithms. As a consequence, music composition techniques based on genetic algorithms, genetic programming, particle swarm optimization, evolutionary strategy, are becoming very common in this area of computer art and yielding interesting results [3].

Nevertheless, automatic music composition based on evolutionary approaches is far from being completely established due to two key aspects: 1) the great challenges of artificially reproducing the intrinsic virtuosity that characterises human composers' capabilities and 2) the difficulties in defining a formal measure for the aesthetic evaluation of auto-generated melodies. These drawbacks are particularly relevant in composition techniques such as the *unfigured bass*, where a human composer creates the melody for the bass voice without the need to specify the counterpoint or the ripieno chords, and an interpreter has to improvise those chords (tenor, alto and soprano) that better adjust to the harmony of the piece.

This research is aimed at improving the capability of evolutionary approaches in automatically generating music melodies by introducing a multi-agent system that comprises coadapting memes that emerge to solve the virtuosity and aesthetics pitfalls characterising the unfigured bass composition technique. In science, genes provide the instructions for making proteins, while a meme is the sociocultural equivalent of a gene containing instructions for carrying out behavior [4]. Taking inspiration from nature, here a meme is modeled as instructions that specify the procedure of a search. In particular, our system has a collection of population-based and local-based meta-heuristics as memes, labeled here as *memetic composer agents*, that by using an opportune learning mechanism based

on fuzzy decision trees, are able to evaluate the strength and specialty of each composer agent in composing unique kinds of unfigured bass compositions. In performing the learning and composition activities, each memetic composer agent employs an innovative musical fitness function based on well-known harmony rules that enables each agent to eastetichally evaluate its music composition virtuosity in a precise and formal way. Successively, the composer agents use this mined knowledge to cooperate and exchange information among them to efficiently explore the space of three voice melodies and achieves composition performance that are significantly better than those obtained by the individual composer agents [1]. At the same time, the cooperation and competition among memes (for the fixed resources available) help facilitates the emergence of creative music pieces (that can never transpire with individualistic efforts) in the search.

As will be shown in the section on experimental results, the cooperation among diverse memes or meta-heuristics strategies, allows our artificial composers to yield improved solutions and performances on the famous Bach's 4-voice compositions over those obtained based on conventional state-of-the-art (both hybrid and non-hibrid) evolutionary algorithms.

The remainder of the paper is organised as follows. In Section II a short survey about memetic algorithms and evolutionary music is reported. Section III show how to model the unfigured bass technique for music composition as an optimization problem. Then, in Section IV the proposed adaptive memetic composer is presented. Next, in Section V we show some results obtained using the composer, and finally in Section VI we present some conclusions. Appendix A introduces some musical concepts necessary to understand the figured bass problem.

II. Related Works

This section shortly introduces some recent research activities related to memetic approaches and evolutionary music.

A. A Brief Introduction to Memetic Algorithms

Memetic algorithms (MAs) are metaheuristics designed to find solutions to complex and difficult optimization problems [6]. They are extensions of the conventional evolutionary algorithms that include a stage of local search optimization as part of their search strategy. MAs have arisen as a response to the problems showed by EAs, which generally suffer from slow convergence to locate a precise enough solution because of their failure to exploit local information. This often limits the practicality of EAs on many large-scale real world problems where the computational time is a crucial consideration.

From an optimization point of view [7], MAs have been shown to be both more efficient (i.e., requiring orders of magnitude fewer evaluations to find optima) and more effective (i.e., identifying higher quality solutions) than traditional EAs on several problem domains. As a result, MAs are gaining wide acceptance, in particular, in well-known combinatorial optimization problems where large instances have been solved to optimality and where other metaheuristics have failed to produce comparable results. Krasnogor and Smith [7] show a complete review of the field, gathering many applications to well-known combinatorial optimization problems.

However, despite the interesting results achieved by MAs, the process of designing effective and efficient MAs still shows some drawbacks. For instance, the difficulty pertaining to the fine tuning of their control parameters, which may require extensive tests, and specifically, of finding a problem-specific meme that suits the problem of interest well [8]. In fact, the choice of memes has been shown to greatly influence the search performance of MAs [9]. This evidence has led the research community to develop MAs capable of adapting their behavior to the characteristics of the instance being solved, obtaining third generation MAs or adaptive MAs. Ong et al. [10] have presented a classification and comparative study of adaptive MAs.

B. Automatic Music Composition and Evolutionary Music

Automatic composition of melodies is one of the most challenging problems in the area of computer art and it is almost as old as the computer itself: the *ILLIAC* suite [11], [12] is a musical piece composed by a computer program and dates back to 1957 just a few years after the introduction of the first computer.

In past years several evolutionary approaches were proposed and some of them also successfully used in the composition of ear pleasing musical pieces. For instance, a genetic algorithm based automatic composition tool named *GenDash* [13], [14] has been used by a famous american composer, Rodney Waschka II, during his live performances. Other genetic approaches have been proposed for music composition: Jacobs system [15], [16] introduced a genetic algorithm based on an "ear" module for listening and evaluating composed melodies in contrast to Waschka's system which completely omitted this fitness evaluation stage. *GaMusic* [17] is an evolutionary system for automatic composition of two-octave melodies on the basis of aesthetic genetic algorithms. Johanson and Poli presented the GP-Music system, a genetic programming approach that evolves short melodic sequences according to both mentor's preferences and autonomous measures by the system itself [18]. Spector and Alpern [19], [20] proposed the *GenBebop*, a system for composing short jazz improvisations, the so called "trading fours", i.e. four bar improvisations on a reply to recently played four bars by another player. A similar approach has also been investigated by Biles who designed *GenJam* [21], [22], [23], [24], an interactive genetic algorithm-based system for trading-fours improvisation. Successively, Thywissen designed *GeNotator*, a hybrid-algorithmic composition tool for computer assisted composition of music [25]. Gibson and Byrne introduced a system for composing short musical pieces using diatonic, four-part Western harmony. The system has been called *NEUROGEN* and utilized a genetic

[1]There has been a long history of research on these different kinds of effort in social evolution since the first study in 1898, where it was shown that cooperation and competition, as compared with individualistic efforts, typically result in higher achievement [5]. The current work thus takes inspirations from here.

algorithm to compose music and neural networks to evaluate the quality of generated results [26]. Unemi created *SBEAT*, an interactive system for breeding short musical phrases (e.g., rhythm patterns) in terms of musical patterns with various rhythm, pitch and velocity [27], [28]. Another proposal to evolve melodies by means of interactive genetic algorithm has also been made by Ralley [29] .

Special attention is here placed on the work of Zaccagnino and De Prisco [43] where they tried to address the figured bass problem with a genetic algorithm. However, they did not formalize the search as an optimization problem mathematically. Furthermore, our proposed evolutionary methodology comprising adaptive memetic agents that are equipped with diverse metaheuristics is capable of uncovering high quality and creative solutions more effectively and efficiently than a simple genetic algorithm. Particularly, our approach employs knowledge that is automatically extracted from the preceding executions of the metaheuristics while the search progress online, thus adapting well to the instances being solved. Further, the cooperation and competition, as compared with individualistic efforts, results in higher achievement and facilitates the emergence of creative music pieces that would never transpire with individualistic efforts.

III. The unfigured music composition technique as a NP-Hard search problem

In this section, a formal representation of the unfigured bass technique as a NP-hard optimisation problem is provided. As aforementioned, with unfigured bass technique, a human composer creates the melody for the bass voice but he/she does not specify the harmony, i.e. the notes corresponding to the other 3 voices: tenor, alto and soprano; only later, when the composition is played, it is the interpreter who has to improvise the chords[2] to suit to the harmony of the piece. As a consequence, informally speaking, this composition technique could be view as an optimisation problem whose solution, for a given input bass line, contains the most appropriate (aesthetically speaking) three voices that complete a 4-voice piece.

However, since no prior knowledge about the three voices necessary for completing a 4-voice piece, it is particularly difficult to formally model the unfigured bass technique as an optimisation problem and solve it through evolutionary techniques. Indeed, the lack of such information makes it impossible to create an initial population of feasible solutions to be evolved. As a consequence, a so-called *figuration step* is necessary to identify a suitable initial three voices knowledge for a given input bass line and finally define the music composition as an optimisation problem in terms of its fitness function and problem constraints.

Figure 1 presents an example of the proposed composition process: Part a) of the figure shows the input (the bass line); part b) outlines the notes that have to be harmonized (called harmonizable notes); part c) adds the symbols that indicate the most suitable chords (the figuration step), while part c) shows the final solution obtained, for example, by using an evolutionary approach.

[2]Each combination of 4 notes interpreted by the 4 voices represents a chord, see Appendix A for a brief introduction to harmony and chords.

A. The figuration step: From unfigured to figured bass

The figuration step decides which chords are those that can produce the most pleasant melody starting from a given input bass line. From an optimisation point of view, this step reduces the search space, which contains all of the possible chords induced by the bass line, to a subset of it, which, according to traditional harmony rules, contains those chords that can produce a better final composition. Precisely, the figuration step is performed with dynamic programming.

However, not all notes included in the bass line are considered in the figuration step. The subset of notes analyzed in the figuration step is determined by the notes in the input bass line and the beat length specified by the rhythm of the composition, also indicated in the bass line. Rhythm is typically specified using a time signature $\frac{n_b}{b_l}$ (in the example in Figure 1, $\frac{n_b}{b_l}$ is $\frac{4}{4}$) that represents the number of beats included in a measure, n_b, and their length, b_l. In particular, we consider as basic time unit the length of a beat. All the notes that are placed on a beat are included in the figuration. We also allow notes to have a length greater than (actually, a multiple of) the beat length. However, if an input note is shorter than the beat length then we consider it only if the note is placed on the beat (otherwise it is only a passage note). By *harmonizable notes* (HNs) we designate those notes that participate in the figuration step.

Hence, in the figuration step we select the most suitable chords for each HN in the bass line. To do that we assign weights to pairs of chords related to two consecutive notes belonging to the harmonizable notes collection. These weights represent well-known harmonic rules that indicate which sequences of chords should sound better (see for example [34]). We have assigned heavier weights to those sequences of chords that are more common and lighter weights to other sequences. Notice that the weights change as a function of the previous chords in the sequence.

The weights are assigned for each pair of consecutive HNs using the following criteria, where c_1 and c_2 represent the chord employed for the first and the second HN in the sequence, respectively:

- c_1 and c_2 are chords in the same major tonality. The assigned weights are given in Table I.
- c_1 and c_2 are chords in the same minor tonality. The assigned weights are given in Table II.
- c_1 and c_2 modulate from a major to a major tonality or from a minor to a minor tonality. The assigned weights are given in Table III.
- c_1 and c_2 modulate from a major to a minor tonality or from a minor to a major tonality. The assigned weights are given in Table IV.

We use a dynamic programming approach (see Algorithm 1) to calculate the most suitable sets of chords C_i that is possible to associate to the i^{th} harmonizable note. The solution given as output by this algorithm is the figured bass version related to the original unfigured bass line. In the next section, the figured bass line is used to define an NP-hard optimisation problem

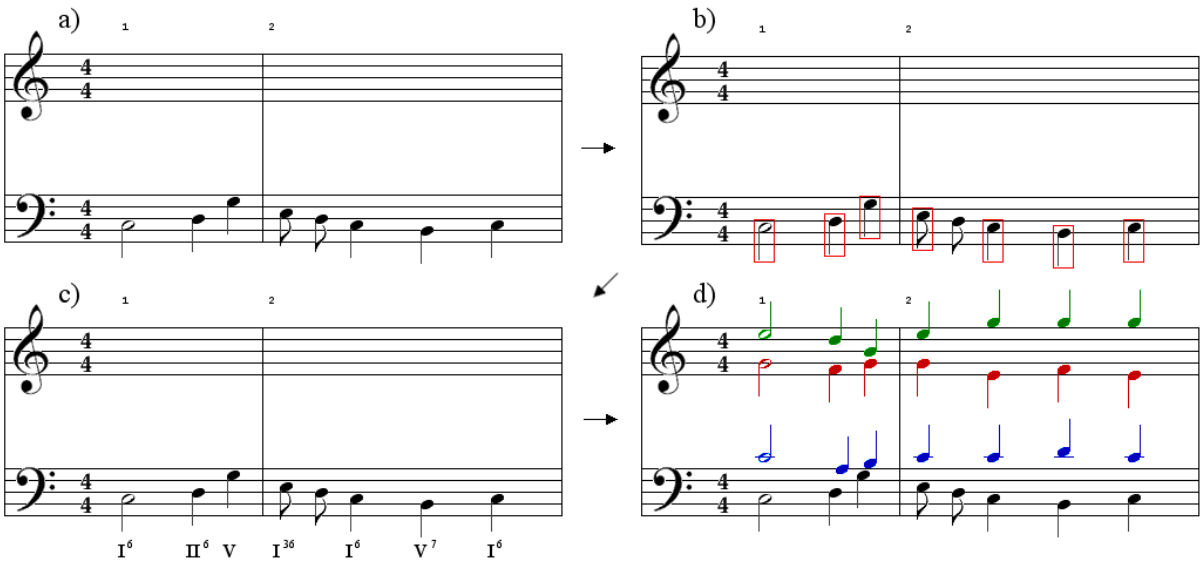

Fig. 1. Example of the automatic unfigured music composition. a) input bass line; b) identification of harmonizable notes; c) figuration; d) final composition.

TABLE III
WEIGHTS FOR MAJOR-TO-MAJOR OR MINOR-TO-MINOR MODULATION.

	C	G	D	A	E	B	G♭	D♭	A♭	E♭	B♭	F
C	2000	0	-500	-1000	-1500	-2000	-2500	-2000	-1500	-1000	-500	0
G	0	2000	0	-500	-1000	-1500	-2000	-2500	-2000	-1500	-1000	-500
D	-500	0	2000	0	-500	-1000	-1500	-2000	-2500	-2000	-1500	-1000
A	-1000	-5000	0	2000	0	-500	-1000	-1500	-2000	-2500	-2000	-1500
E	-1500	-1000	-1000	0	2000	0	-1000	-1000	-1500	-2000	-2500	-2000
B	-2000	-1500	-1500	-5000	0	2000	- 0	-500	-1000	-1500	-2000	-2500
G♭	-2500	-2000	-2000	-1000	-500	0	2000	0	-500	-1000	-1500	-2000
D♭	-2000	-2500	-2500	-1500	-1000	-500	0	2000	0	-500	-1000	-1500
A♭	-1500	-2000	-2000	-2000	-1500	-1000	-500	0	2000	0	-500	-1000
E♭	-1000	-1500	-1500	-2500	-2000	-1500	-1000	-500	0	2000	0	-500
B♭	-500	-1000	-1000	-2000	-2500	-2000	-1500	-1000	-500	0	2000	0
F	0	-500	-1000	-1500	-2000	-2500	-2000	-1500	-1000	-500	0	2000

TABLE I
WEIGHTS FOR STEPPING BETWEEN CHORDS IN THE SAME MAJOR TONALITY.

	I	II	III	IV	V	VI	VII
I	250	200	50	200	250	50	10
II	100	100	100	150	2000	150	10
III	100	100	100	200	100	250	10
IV	250	150	100	200	1500	100	10
V	2000	100	100	100	250	150	10
VI	100	200	150	150	200	200	10
VII	1000	50	150	50	50	100	200

TABLE II
WEIGHTS FOR STEPPING BETWEEN CHORDS IN THE SAME MINOR TONALITY. THE CHORDS ARE DENOTED II^*, III^* ETC. BECAUSE IN A MINOR TONALITY ON THESE DEGREES WE CAN HAVE SEVERAL TYPE OF CHORDS.

	I	II*	III*	IV*	V*	VI*	VII*
I	250	200	50	200	250	50	10
II*	100	100	100	150	2000	150	10
III*	100	100	100	200	100	250	10
IV*	250	150	100	200	1500	100	10
V*	2000	100	100	100	250	150	10
VI*	100	200	150	150	200	200	10
VII*	1000	50	150	50	50	100	200

whose efficient solution corresponds to a high-quality music composition activity.

B. Formalizing the figured bass technique as an optimization problem

Consider a set V of n HNs, where each HN i can be interpreted as a chord c_i from a set of available chords C_i obtained by the figuration step. We define $\mathbf{c} = \{c_1, \ldots, c_n\}$, as the joint harmonization and $\xi(\mathbf{c})$ as the global aestethic payoff obtained *by playing* $\mathbf{c}$. The figured bass problem entails finding the optimal joint harmonization $\mathbf{c}^*$ that maximizes $\xi(\mathbf{c})$.

In this problem, we can approximate $\xi(\mathbf{c})$ as the sum of the rewards locally obtained for each HN. Additionally we can suppose that these rewards depend on c_i (the chord chosen for HN i) and the chords chosen for a subset of V, defined as $\Gamma(i)$, which represents the neighbors of HN i. The set of chords that determine each local reward is defined as $\mathbf{c_i}$ and

TABLE IV
WEIGHTS FOR MAJOR-TO-MINOR OR MINOR-TO-MAJOR MODULATION.

	C	G	D	A	E	B	G♭	D♭	A♭	E♭	B♭	F
C	-500	-200	0	500	0	-200	-500	-700	-1000	-1200	-1000	-700
G	-700	-500	-200	0	500	0	-200	-500	-700	-1000	-1200	-1000
D	-1000	-700	-500	-200	0	500	0	-200	-500	-700	-1000	-1200
A	-1200	-1000	-700	-500	-200	0	500	0	-200	-500	-700	-1000
E	-1000	-1200	-1000	-700	-500	-200	0	-500	0	-200	-500	-700
B	-700	-1000	-1200	-1000	-700	-500	-200	0	-500	0	-200	-500
G♭	-500	-700	-1000	-1200	-1000	-700	-500	-200	0	-500	0	-200
D♭	-200	-500	-700	-1000	-1200	-1000	-700	-500	-200	0	-500	0
A♭	0	-200	-500	-700	-1000	-1200	-1000	-700	-500	-200	0	-500
E♭	-500	0	-200	-500	-700	-1000	-1200	-1000	-700	-500	-200	0
B♭	0	-500	0	-200	-500	-700	-1000	-1200	-1000	-700	-500	-200
F	-200	0	-500	0	-200	-500	-700	-1000	-1200	-1000	-700	-500

Algorithm 1: Pseudo-code used to select the most suitable chords.

```
Input: HN: array of n harmonizable notes.
Output: most suitable chords.
begin
    for i = 1 ; i < n ; i++ do
        A[i] = calculate set of chords that can be
        obtained using HN[i];
    end
    for i = n ; i > 0 ; i-- do
        for Chord c ∈ A[i] do
            for Chord d ∈ A[i+1] do
                weight=calculate weight using Tables
                I, II, III and IV;
                weight+=maxWeight[d];
                if weight > maxWeight[c] then
                    maxWeight[c] = weight;
                    mostSuitableFollowingChord[c] = d;
                end
            end
        end
    end
    mostSuitableChord[1]=chord with maximum weight
    for HN[1];
    for i = 2 ; i < n ; i++ do
        mostSuitableChord[i] =
        mostSuitableFollowingChord [
        mostSuitableChord [i-1]];
    end
end
```

TABLE V
RULES TABLE

Single voice error	**Cost**	**Type**
sixth	-1000	normal
aug. fourth	-1000	normal
aug. fifth	-1000	normal
seventh	-3000	critical
> octave	-3000	critical
Single voice cost	**Cost**	**Type**
jump	-interval	no error
Two voices error	**Cost**	**Type**
hidden unison	-100	normal
hidden fifth	-100	normal
hidden octave	-100	normal
unison	-3000	critical
parallel fifth	-3000	critical
parallel octave	-3000	critical
Error within chord	**Cost**	**Type**
octave leap	-100	normal

takes values from $C_i \times (\times_{j \in \Gamma(i)} C_j)$. Thus, we define the local reward obtained by chord i as $\xi_i(\mathbf{c_i})$, and the global reward as $\xi(\mathbf{c}) = \sum_{i=1}^{n} \xi_i(\mathbf{c_i})$.

In this paper, we consider only reward functions defined over at most two chords. Hence, we can represent the problem as an undirected graph $G = (V, E)$ in which each node $i \in V$ represents a HN, and each edge $(i, j) \in E$ indicates that the corresponding HNs influence each other, and thus $j \in \Gamma(i)$ and $i \in \Gamma(j)$. When considering just relations between two chords, the edges of the graph will just connect two consecutive HNs. This graph is known as a Harmonization Graph (HG). Following this notation the global reward $\xi(\mathbf{c})$ can be rewritten as:

$$\xi(\mathbf{c}) = \sum_{i \in V} f_i(c_i) + \sum_{(i,j) \in E} f_{ij}(c_i, c_j) \qquad (1)$$

where $f_i(c_i)$ is the reward contributed by HN i when interpreted using chord c_i and $f_{ij}(c_i, c_j)$ is the reward contributed by the pair of chords (c_i, c_j).

To define functions $f_i(c_i)$ and $f_{ij}(c_i, c_j)$, we take into account musical composition rules from the common practice period. In particular, we consider a set of aesthetic rules, which are detailed in Table V. These rules indicate penalties for rule violations, instead of rewards, therefore, they are negative. For example, if a voice makes an augmented fourth jump, a penalty of 100 is arrived. For some violations (which we call critical errors), higher penalties are imposed. For example, for a voice jump bigger than an octave or for two voices that create exposed parallel fifths, a penalty of 3000 is considered.

This problem is equivalent to the problem of recovering the maximum a posteriori configuration of random variables in a graphical model, which is an important problem with applications ranging from protein folding to image processing. In general, this problem is NP-hard [42]. Consequently it is difficult to solve using a deterministic approach, especially when the size of the instance is big, as in Figured Bass problem. Thus it is necessary to use flexible approaches, like metaheuristics. In this paper, to cope with the difficulties that Figured Bass problem presents we apply and adapt a method-

ology [35] that employs a set of optimization agents that, through collaboration, constitute a memetic strategy capable of adapting to the different instances of the problem.

IV. Music Composition with Composer Agents Cooperation and Competition

In this section we introduce our proposal for an automatic music composition framework (Fig. 2) capable of obtaining high-quality musical pieces that adhere to scholastic rules by solving the problem introduced in Section III.

From a high-level point of view, the framework is designed as a multi-agent system with two key steps: in the first step, a collection of population based composer agents take a (figured) bass line as input and cooperate in a parallel and adaptive way in order to efficiently explore the space of 4-voice musical pieces and identify a suitable melodies subspace. In the second step, a set of local composer agents cooperate in a parallel and adaptive way to exploit the region computed by the population based agents and identify a high quality 4-voice musical piece completing the input bass line. In particular we have employed up to six different memetic composer agents, three evolutionary composer agents, namely Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), and three local composer agents, which are Tabu Search (TS), Simulated Annealing (SA), and Variable Neighborhood Search (VNS).

The cooperation and adaptability features of the proposed multi-agent system are provided by a so-called *coordinator agent* that learns and becomes aware of the strengths and specialty of each evolutionary and local composer agent in evolving 4-voices melodies starting from a given input bass line. Further, the coordinator agent knowledge is acquired through a training phase where a collection of Fuzzy Decision Trees (FDTs) [38] containing this knowledge, is learned. The coordinator agent uses the knowledge contained in the FDTs and a collection of TSK fuzzy rules to evaluate the quality of melodies composed by each agent and "intelligently move" these melodies across different populations to improve the composition capabilities of each composer agent. The following subsections formally introduce the different sub-modules composing the proposed framework.

A. Learning the coordinator agent knowledge

In order to obtain the FDTs that define the coordination agent's knowledge, it is necessary to apply a knowledge extraction process that gathers and summarizes useful information about the performance of the composer agents. In particular, the coordinator employs two sets of FDTs, which have been named *parameter trees* and *weight trees*. The former set contains knowledge that describes, for each possible bass line, the most convenient values for the parameters of the metaheuristics implemented by the composer agents. While the latter includes knowledge, in the form of weights, that rank the expected performance of each metaheuristic. These weights are exploited by a set of rules that controls the exchange of melodies, which is explained below. The knowledge extraction process is divided into two different phases explained below: data preparation and data mining.

Note that this configuration procedure is applied just once, before the framework can be applied to compose. After that the FDTs are used to configure the framework independently of the bass line being harmonized.

1) Data preparation: The first step to obtain effective FDTs is to create a set of databases that contain highly representative data of the performance of the metaheuristics implemented by the composer agents. These databases have to accurately describe the performance of the metaheuristics with different kinds of instances of the Figured Bass problem. To obtain such databases we employ a procedure that obtains 4-voice melodies using individually the different metaheuristics with diverse parameter configurations.

In particular, we chose a set of 30 representative instances of the Figured Bass problem and solved them using the four metaheuristics. These instances were obtained from J.S. Bach's chorales[3], which represent unreachable examples of harmonization. Each metaheuristic used several configurations of parameters values, which varied depending on the number of parameters, their possible values and previous knowledge about their behavior. 256, 81, 216, 72, 36 and 130 different parameter combinations were respectively employed for GA, PSO, ACO, SA, TS and VNS. In addition, each bass line was harmonized 10 times with each parameter combination. For each metaheuristic we recorded a *raw database* that contained important information about the obtained melodies. In particular, each row included:

- A description of the bass line, composed of:
  - Its musical tempo, or the number of beats per minute, which indicates the speed or pace of the piece.
  - The number of beats indicated by its time signature. For instance, 3 in $\frac{3}{4}$.
  - The beat length indicated by its time signature. For example, 4 in $\frac{3}{4}$.
  - The key signature of the composition, e.g. C Major or A minor.
  - The number of HNs.
  - The number of measures that compose the piece.
- The values of the parameters used to harmonize it.
- The final fitness value obtained by the melody.

Table VI shows an example of raw database related to GA.

However, raw databases are not yet prepared to be exploited by the FDTs' learning algorithms. Because they store an excessive amount of information, which cannot produce the desired outputs (weights and parameters). To reduce the size of raw databases and prepare them for the learning algorithms we applied a refining process that produced the following *refined databases*:

- Evolutionary weights database. This database was employed to train the FDT that produces the weights in the evolutionary phase.
- Local weights database. This database was utilized to train the FDT that produces the weights in the local phase.

[3]The scores of Bach's chorales can be found in http://www.jsbchorales.net/index.shtml

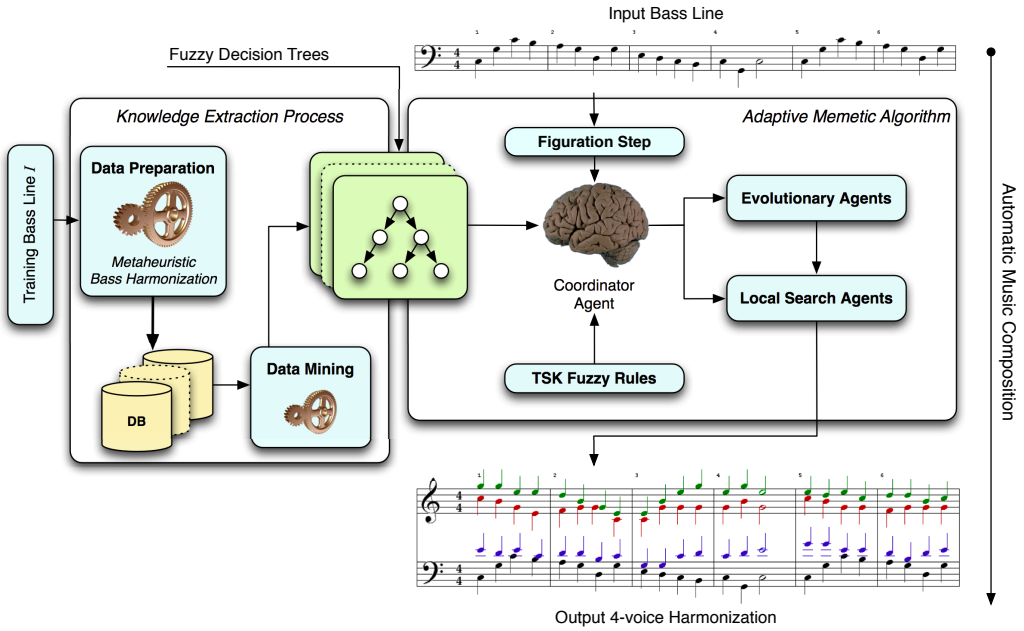

Fig. 2. Memetic Music Composer.

- A set of parameters databases. As many databases as parameters of the different metaheuristics. For instance, in the example shown in Table VI, GA needs 4 parameter databases, one per parameter, and the rest of metaheuristics need similar databases. These databases were used to learn the trees that indicate the most suitable parameters for each instance.

The procedure to create refined databases follows the following steps:

1) We first need to calculate the average fitness value obtained by each metaheuristic with each parameters combination and for each bass line. Following the example shown in Table VI, for Bach's chorale BWV 26,[4] we calculate the average fitness value obtained by GA using parameters combination 1 (Tournament selection, mutation probability 0.01, crossover probability 0.5 and number of individuals 20), then we calculate the same average fitness value for parameters combination 2, 3 and so on. We repeated the procedure for Bach's chorale 70, and so on. We proceed with PSO, SA and TS in an analogous way.
2) Successively, for each bass line and metaheuristic, we isolate the best parameters combination, as the one that has obtained the highest fitness value. We call this combination b_i^a, where a represents the instance and i the metaheuristic. The fitness value associated to b_i^a is $\dot{\xi}_i^a$.
3) For each metaheuristic and each one of its parameters, a parameter database is created that includes one row per bass line. Each row includes the description of the bass line and the value of the parameter in b_i^a.
4) For each phase (evolutionary and local search), a database is generated that contains one row per instance. Each row includes the description of the instance and the weights calculated as

$$\omega_i^a = \frac{\dot{\xi}_i^a}{\sum_{j \in M} \dot{\xi}_j^a}$$

where M represents the set of composer agents involved in the phase.

Refining process is illustrated in Table VII, which presents an example of the two refining steps.

2) Data Mining: Once performance information has been gathered and collected through refined databases, we extract a collection of knowledge models in the data mining phase. These models include two FDTs (evolutionary and local phases) and as many FDTs as configuration parameters of the different metaheuristics. Each one of these FDTs is directly obtained applying FDT learning algorithm [38] to the databases created in the previous step. FDTs have been chosen because of their previous success in similar problems [35], [36], [37], and their interpretability, which favors the understanding of the obtained models.

B. Cooperation and coordination control

After learning the FDTs that define the knowledge of the coordinator agent, it then becomes capable of controlling the cooperation and coordination of the composer agents. However, cooperation and coordination control is non-trivial, since an unrestricted exchange of melodies can lead to the appearance of melodies that do not fully satisfy quality standards (as often encountered with traditional optimization problems [39]). This problem can be avoided by monitoring the exchange of melodies and preventing undesirable interchanges. To do that the coordinator agent uses a set of fuzzy rules that exploit the FDTs to control the exchange. In particular, the

[4]Numeration of Bach's chorales follows the Bach-Werke-Verzeichnis (BWV) catalogue by Wolfgang Schmieder.

TABLE VI
EXAMPLE OF RAW DATABASE

			Attributes						**Parameters**				
Instance	Par. comb.	it.	Tempo	# beats	beat length	key sign.	# NHs	# measures	Sel. strategy	Mut. prob.	Cross prob.	# indiv.	**Final fitness**
Bach's chorale BWV 26	1	1	240	4	4	A min	39	10	Tournament	0.01	0.5	20	92342
		...				...				...			...
		10	240	4	4	A min	39	10	Tournament	0.01	0.5	20	91358
	...	...				...				...			...
	...	...				...				...			...
	256	1	240	4	4	A min	39	10	Roulette	0.5	0.9	40	87964
		...				...				...			...
		10	240	4	4	A min	39	10	Roulette	0.5	0.9	40	90957
Bach's chorale BWV 70	1	1	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124831
		...				...				...			...
		10	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124330
	...	...				...				...			...
	...	...				...				...			...
	256	1	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	132587
		...				...				...			...
		10	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	133641
...	...	...				...				...			...
...	...	...				...				...			...
...	...	...				...				...			...
Bach's chorale BWV 153.9	1	1	240	3	4	C Maj	40	17	Tournament	0.01	0.5	20	76670
		...				...				...			...
		10	240	3	4	C Maj	40	17	Tournament	0.01	0.5	20	78549
	...	...				...				...			...
	...	...				...				...			...
	256	1	240	3	4	C Maj	40	17	Roulette	0.5	0.9	40	69325
		...				...				...			...
		10	240	3	4	C Maj	40	17	Roulette	0.5	0.9	40	71537

TABLE VII
EXAMPLE OF REFINED DATABASE

			Attributes						**Parameters**				
Instance	Par. comb.	it.	Tempo	# beats	beat length	key sign.	# NHs	# measures	Sel. strategy	Mut. prob.	Cross prob.	# indiv.	**Final fitness**
...	...	...				...				...			...
Bach's chorale BWV 70	1	1	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124831
		...				...				...			...
		10	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124330
	...	...				...				...			...
	...	...				...				...			...
	256	1	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	132587
		...				...				...			...
		10	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	133641
...	...	...				...				...			...

↓ ↓ Fitness Average Computation ↓ ↓

Bach's chorale BWV 70	1	avg.	60	3	4	G Maj	62	25	Tournament	0.01	0.5	20	124662.3
	...	...				...				...			...
	...	...				...				...			...
	256	1	60	3	4	G Maj	62	25	Roulette	0.5	0.9	40	134903.2
	...	...				...				...			...

↓ ↓ New entries in the refined databases ↓ ↓

Weights database							Sel. Strategy database					
Tempo	# beats	beat length	key sign.	# NHs	ω_{GA}	ω_{PSO}	Tempo	# beats	beat length	key sign.	# NHs	parameter value
60	3	4	G Maj	62	0.62	0.38	60	3	4	G Maj	62	Roulette

FDTs analyze the annotated bass line and estimate a weight ω_i that represents the expected performance of the metaheuristic implemented by composer agent i, where $\omega_i > \omega_j$ indicates that composer agent i generally obtains better results than composer agent j.

In addition, in each step (evolutionary and local), the coordinator agent employs a collection of TSK fuzzy rules, which exploits the weights, to decide when it is necessary to exchange melodies between two or more composer agents. These rules reflect the following structure:

if $(\omega_1 \cdot d_1)$ is *enough*
then $poor\_musical\_quality_1 = 1$
...
if $(\omega_{h-1} \cdot d_{h-1})$ is *enough*
then $poor\_musical\_quality_{h-1} = 1$
if $(\omega_{h+1} \cdot d_{h+1})$ is *enough*
then $poor\_musical\_quality_{h+1} = 1$
...
if $(\omega_n \cdot d_n)$ is *enough*
then $poor\_musical\_quality_n = 1$

where:

- n is the number of composer agents participating in a given step.
- c_h is the composer agent being evaluated by the rule;
- $d_i = (\xi(c_i) - \xi(c_h))/\max(\xi(c_i), \xi(c_h))$, where ξ is the fitness function defined in Section III;
- $\omega_i \in [0, 1]$ is the weight of c_i;
- $enough$ is a fuzzy set with trapezoidal membership function defined by a quadruplet (a, b, c, d) and whose universe of discourse is the range $[0, 1]$.
- $poor\_musical\_quality_i$ with $i = 1, \ldots, n$, is a TSK variable belonging to $[0, 1]$. Higher values of $poor\_musical\_quality_i$ illustrate that c_h may be benefited from an exchange of melodies with c_i.

Every composer agent included in the framework is associated to a set of fuzzy rules that relate it to the rest of composer agents belonging to same step. For instance, in a system that includes GA and PSO in the evolutionary step we need one rule that relates the composer agent implementing GA to the one implementing PSO and another one that relates PSO to GA. These two rules represent how convenient it is to receive melodies from the other composer agent. A similar set of rules is necessary for the local search step.

However, not every activated rule triggers a melody exchange. Only those rules whose $poor\_musical\_quality$ is higher than a given threshold α do so. When such a situation occurs the exchange of melodies is performed in the following way:

- If the exchange involves composer agents employing evolutionary techniques, then a subset of c_h's population of melodies is replaced by the best individuals in c_i (the metaheuristic that triggered the rule) population. Here the proportion corresponds to ω_i.
- If the interchange engages composer agents implementing local search strategies, then the current melody of c_h is replaced by a melody "similar" to the best melody obtained by c_i. Here "similar" represents the application of a mutation operator.

V. Experimental Analysis

This section studies the validity and effectiveness of the proposed technique to solve the Figured Bass problem. Different tests have been carried out to assess how the inclusion of new metaheuristics can affect the performance of the proposed technique. In particular, the experiments start with non-adaptive MAs that just include one population based algorithm and one local search algorithm. Successively more metaheuristics are added, both population based and local search based, until a cooperative strategy that contains six different metaheuristics is obtained. To finish, the proposed strategy is compared to a method that was previously applied to Figured Bass problem [43].

A. Configuration of the Experiments

The experiments compare the following 13 strategies:

- A collection of non adaptive MAs, that combine one population based strategy and one local search based. In particular:
  - Genetic Algorithm and Simulated Annealing (GA+SA).
  - Genetic Algorithm and Tabu Search (GA+TS).
  - Particle Swarm Optimization and Simulated Annealing (PSO+SA).
  - Particle Swarm Optimization and Tabu Search (PSO+TS).
- A collection of adaptive MAs obtained adding new metaheuristics to the previous combinations. These strategies have been named MMC (Memetic Music Composer), and to distiguish the different combinations we indicate as a superscript the included population based metaheuristics, and as a subscript the employed local search metaheuristics. In particular:
  - Genetic Algorithm, Particle Swarm Optimization and Simulated Annealing ($MMC_S^{G\&P}$).
  - Genetic Algorithm, Particle Swarm Optimization and Tabu Search ($MMC_T^{G\&P}$).
  - Genetic Algorithm, Simulated Annealing and Tabu Search ($MMC_{S\&T}^{G}$).
  - Particle Swarm Optimization, Simulated Annealing and Tabu Search ($MMC_{S\&T}^{P}$).
  - Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing and Tabu Search ($MMC_{S\&T}^{G\&P}$).
  - Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Simulated Annealing and Tabu Search ($MMC_{S\&T}^{G\&P\&A}$).
  - Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing,Tabu Search and Variable Neighborhood Search ($MMC_{S\&T\&V}^{G\&P}$).
  - Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Simulated Annealing,Tabu Search and Variable Neighborhood Search ($MMC_{S\&T\&V}^{G\&P\&A}$).
- Evolutionary Music Composer (EMC). An evolutionary algorithm proposed by Zaccagnino and De Prisco [43].

This is, to the best of our knowledge, the only article that has tried to solve a problem similar to the Figured Bass problem as defined in this paper.

Each strategy had 50,000 solution evaluations to resolve each instance included in the experiments. Instead of using a stopping condition based on time, we decided to use solutions evaluations because it produces fairer comparisons with other techniques. Using this procedure, we can compare the results independently of the programming language used to code the strategy, the computer architecture of the machine where the test were run, or the ability of the programmer to optimize his code.

However, we consider that the time can be considered as an important measure of performance, and in that sense we have to point out that the overhead derived from the execution of the coordinator is negligible and that the execution time of the cooperative strategies is always smaller than that of the slower individual metaheuristic for the same number of evaluations of the objective function.

Regarding execution mode, for cooperative strategies one can resort to parallel schemes if time is important, or one can simulate the parallelism in a one-processor computer. This is the strategy taken here and the procedure is extremely simple. We construct an array of solvers and we run them using a round-robin schema. This implementation uses an asynchronous communication mode that is simulated in this way: solvers are executed during a certain number of evaluations of the objective function. This amount of evaluations is a random number in the interval $[5000, 5500]$ and after this period of time, information exchanges are performed. These steps are repeated until the termination condition, given in terms of objective function evaluations, is fulfilled.

The set of instances employed in the experiments was chosen among J.S. Bach's chorales, and included chorales: BWV 4.8, BWV 8.6, BWV 12.7, BWV 16.6, BWV 17.7, BWV 19.7, BWV 25.6, BWV 28.6, BWV 31.9, BWV 33.6, BWV 39.7, BWV 40.6, BWV 43.11, BWV 47.5, BWV 55.5, BWV 62.6, BWV 67.7, BWV 77.6, BWV 81.7, BWV 88.7. Each individual instance was solved 10 times using each strategy. Results show the averages and the standard deviation (as a subscript).

B. Results

1) Comparing non-adaptive and adaptive MAs: Table VIII presents the results obtained by the different strategies. These results are presented in terms of the average percentage of error with regard to the highest fitness value obtained for each instance, indicating between parenthesis the standard deviation. With the aim of demonstrating the effectiveness of the adaptivity mechanism implemented by the proposed technique, we first analyze these results comparing adaptive and non-adaptive MAs. To perform a more rigorous comparison, we apply the methodology proposed by García et al. [41], which employs non-parametric statistical tests to study the significant differences in the results obtained by different methods.

Initially, using Friedman test, we evaluate if there is any significant difference among the results obtained by every metaheuristic. As a result a p-value of 2.2e-16 is obtained, which indicates that there exist differences with a confidence higher than 99.9%. After that, the performance of each strategy is compared with that of the others in a pairwise way by applying Wilcoxon unsigned rank test. As null hypothesis an equivalent result distribution is assumed. On the other hand, the considered alternative hypothesis is that the true location shift is not equal to 0. When more than 2 techniques are compared, the obtained p-values need to be adjusted. We use Benjamini-Hochberg method to adjust them. Table IX summarizes the results obtained. This table shows symbol '+' when the difference in performance between the two strategies is significant (p-value smaller than 0.05), and symbol '-' when no statistical significance was obtained.

The first conclusion that can be drawn from the obtained results is the usefulness of the adaptivity and cooperation introduced in the proposed system. In particular, all adaptive memetic strategies except two ($MMC_T^{G\&P}$ and $MMC_{S\&T}^{P}$) obtained results significantly better than those obtained by all the non-adaptive memetic strategies. In addition, the two adaptive strategies that could not improve the performance of the non-adaptive strategies obtained results statistically equivalent, except in one case, in which GA+SA outperforms $MMC_T^{G\&P}$.

The results also indicate the scalability of the proposed technique. In most of the cases (10 out of 16), when a new metaheuristic was added to a previous combination, the obtained results were significantly better than those obtained before adding it. In the remaining six cases, the results did not show statistically significant differences. And in no case the results worsened. Furthermore, the strategy that obtained the best results was the one that included all six individual metaheuristics. In particular, the results obtained by this strategy were significantly better than the results of all other tested memetic strategies, adaptive or not.

2) Comparing with a literature technique: We now compare the performance of the memetic algorithms against that of EMC, whose results are shown in Table VIII. Once again, to obtain a fairer comparison, we analyze the results statistically following the same approach as in the previous section. The last column of Table IX presents the results of the statistical tests. As it can be seen all the implemented memetic algorithms obtain results statistically better than those obtained by EMC. These results demonstrate the ability of memetic algorithms to solve a complex problem like the Figured Bass Problem.

C. Analysis of the solutions obtained

Analyzing the results pertaining to the obtained fitness value permits one to identify the best strategy for addressing a problem. However, this does not bring insights to the musical characteristics of the found solutions. In order to fill in this gap, we study the obtained solutions for Bach's choral BWV 25.6 in this subsection. Due to space restrictions, we shall limit this study to the first measures of the piece.

Figure 3 presents the chorale's bass line (the input). If we were to solve this instance with simple random search, we

TABLE VIII
OBTAINED RESULTS - Error percentage with regard to the best obtained fitness function, and standard deviation in parenthesis

Bach's chorale	GA+SA	GA+TS	PSO+SA	PSO+TS	$MMC_S^{G\&P}$	$MMC_T^{G\&P}$	$MMC_{S\&T}^{G}$
n. 408	0,77%$_{(0,01)}$	0,62%$_{(0,00)}$	1,80%$_{(0,02)}$	2,99%$_{(0,04)}$	0,31%$_{(0,00)}$	0,62%$_{(0,01)}$	0,45%$_{(0,00)}$
n. 806	1,22%$_{(0,01)}$	0,96%$_{(0,01)}$	1,69%$_{(0,02)}$	2,01%$_{(0,02)}$	1,22%$_{(0,01)}$	1,52%$_{(0,02)}$	0,86%$_{(0,01)}$
n. 1207	0,23%$_{(0,00)}$	2,18%$_{(0,03)}$	0,42%$_{(0,00)}$	3,55%$_{(0,04)}$	0,49%$_{(0,00)}$	2,08%$_{(0,03)}$	0,09%$_{(0,00)}$
n. 1606	0,32%$_{(0,00)}$	0,24%$_{(0,00)}$	0,32%$_{(0,00)}$	0,41%$_{(0,00)}$	0,18%$_{(0,00)}$	0,74%$_{(0,01)}$	0,15%$_{(0,00)}$
n. 1707	0,67%$_{(0,00)}$	1,12%$_{(0,01)}$	1,46%$_{(0,02)}$	3,92%$_{(0,05)}$	0,36%$_{(0,00)}$	1,13%$_{(0,01)}$	0,59%$_{(0,00)}$
n. 1907	0,25%$_{(0,00)}$	0,57%$_{(0,00)}$	0,25%$_{(0,00)}$	0,98%$_{(0,01)}$	0,13%$_{(0,00)}$	0,30%$_{(0,00)}$	0,09%$_{(0,00)}$
n. 2506	0,58%$_{(0,00)}$	1,20%$_{(0,02)}$	1,62%$_{(0,02)}$	3,90%$_{(0,10)}$	0,46%$_{(0,00)}$	1,86%$_{(0,03)}$	0,15%$_{(0,00)}$
n. 2806	0,46%$_{(0,00)}$	1,98%$_{(0,02)}$	0,80%$_{(0,01)}$	3,30%$_{(0,04)}$	0,16%$_{(0,00)}$	2,19%$_{(0,02)}$	0,38%$_{(0,00)}$
n. 3109	0,40%$_{(0,00)}$	2,13%$_{(0,04)}$	1,57%$_{(0,02)}$	4,32%$_{(0,15)}$	0,62%$_{(0,00)}$	2,28%$_{(0,03)}$	0,42%$_{(0,00)}$
n. 3306	0,77%$_{(0,00)}$	0,37%$_{(0,00)}$	0,42%$_{(0,00)}$	0,64%$_{(0,00)}$	0,34%$_{(0,00)}$	1,17%$_{(0,01)}$	0,40%$_{(0,00)}$
n. 3907	0,49%$_{(0,00)}$	0,66%$_{(0,01)}$	0,64%$_{(0,01)}$	1,28%$_{(0,02)}$	0,09%$_{(0,00)}$	0,60%$_{(0,00)}$	0,17%$_{(0,00)}$
n. 4006	0,47%$_{(0,00)}$	0,44%$_{(0,00)}$	0,38%$_{(0,00)}$	0,75%$_{(0,01)}$	0,37%$_{(0,00)}$	0,58%$_{(0,01)}$	0,45%$_{(0,00)}$
n. 4311	0,33%$_{(0,00)}$	0,31%$_{(0,00)}$	0,86%$_{(0,01)}$	0,39%$_{(0,00)}$	0,20%$_{(0,00)}$	0,37%$_{(0,00)}$	0,44%$_{(0,00)}$
n. 4705	0,46%$_{(0,01)}$	0,12%$_{(0,00)}$	0,11%$_{(0,00)}$	0,12%$_{(0,00)}$	0,12%$_{(0,00)}$	0,25%$_{(0,00)}$	0,11%$_{(0,00)}$
n. 5505	0,24%$_{(0,00)}$	0,43%$_{(0,00)}$	0,39%$_{(0,00)}$	0,86%$_{(0,01)}$	0,19%$_{(0,00)}$	0,31%$_{(0,00)}$	0,23%$_{(0,00)}$
n. 6206	0,21%$_{(0,00)}$	0,06%$_{(0,00)}$	0,37%$_{(0,00)}$	0,06%$_{(0,00)}$	0,05%$_{(0,00)}$	0,38%$_{(0,00)}$	0,24%$_{(0,00)}$
n. 6707	0,67%$_{(0,00)}$	0,42%$_{(0,00)}$	0,19%$_{(0,00)}$	0,26%$_{(0,00)}$	0,25%$_{(0,00)}$	0,59%$_{(0,01)}$	0,16%$_{(0,00)}$
n. 7706	0,71%$_{(0,01)}$	1,26%$_{(0,02)}$	0,75%$_{(0,01)}$	2,04%$_{(0,03)}$	0,22%$_{(0,00)}$	1,09%$_{(0,01)}$	0,40%$_{(0,00)}$
n. 8107	0,68%$_{(0,01)}$	0,24%$_{(0,00)}$	0,80%$_{(0,01)}$	1,56%$_{(0,02)}$	0,21%$_{(0,00)}$	0,76%$_{(0,00)}$	0,60%$_{(0,00)}$
n. 8807	0,29%$_{(0,00)}$	1,85%$_{(0,02)}$	0,23%$_{(0,00)}$	2,58%$_{(0,03)}$	0,23%$_{(0,00)}$	1,35%$_{(0,01)}$	0,40%$_{(0,00)}$
Average	0,51%$_{(0,00)}$	0,86%$_{(0,01)}$	0,75%$_{(0,01)}$	1,79%$_{(0,03)}$	0,31%$_{(0,00)}$	1,01%$_{(0,01)}$	0,34%$_{(0,00)}$

Bach's chorale	$MMC_{S\&T}^{P}$	$MMC_{S\&T}^{G\&P}$	$MMC_{S\&T}^{G\&P\&A}$	$MMC_{S\&T\&V}^{G\&P}$	$MMC_{S\&T\&V}^{G\&P\&A}$	EMC
n. 408	2,05%$_{(0,03)}$	0,36%$_{(0,00)}$	0,02%$_{(0,00)}$	0,51%$_{(0,00)}$	0,01%$_{(0,00)}$	20,56%$_{(1,66)}$
n. 806	2,41%$_{(0,00)}$	0,66%$_{(0,00)}$	0,03%$_{(0,00)}$	0,69%$_{(0,00)}$	0,02%$_{(0,00)}$	5,12%$_{(0,20)}$
n. 1207	0,13%$_{(0,00)}$	0,38%$_{(0,00)}$	0,03%$_{(0,00)}$	0,17%$_{(0,00)}$	0,02%$_{(0,00)}$	0,24%$_{(0,00)}$
n. 1606	0,12%$_{(0,00)}$	0,08%$_{(0,00)}$	0,03%$_{(0,00)}$	0,27%$_{(0,00)}$	0,02%$_{(0,00)}$	1,75%$_{(0,03)}$
n. 1707	1,30%$_{(0,01)}$	0,36%$_{(0,00)}$	0,03%$_{(0,00)}$	0,88%$_{(0,00)}$	0,02%$_{(0,00)}$	49,01%$_{(4,92)}$
n. 1907	0,18%$_{(0,00)}$	0,25%$_{(0,00)}$	0,02%$_{(0,00)}$	0,20%$_{(0,00)}$	0,01%$_{(0,00)}$	16,55%$_{(1,09)}$
n. 2506	0,79%$_{(0,01)}$	0,41%$_{(0,00)}$	0,06%$_{(0,00)}$	0,24%$_{(0,00)}$	0,04%$_{(0,00)}$	10,36%$_{(0,78)}$
n. 2806	0,33%$_{(0,00)}$	0,25%$_{(0,00)}$	0,02%$_{(0,00)}$	0,38%$_{(0,00)}$	0,01%$_{(0,00)}$	2,13%$_{(0,04)}$
n. 3109	1,54%$_{(0,02)}$	0,33%$_{(0,00)}$	0,02%$_{(0,00)}$	0,87%$_{(0,01)}$	0,02%$_{(0,00)}$	15,57%$_{(0,84)}$
n. 3306	0,30%$_{(0,00)}$	0,47%$_{(0,00)}$	0,03%$_{(0,00)}$	0,43%$_{(0,00)}$	0,02%$_{(0,00)}$	29,74%$_{(2,31)}$
n. 3907	0,43%$_{(0,00)}$	0,23%$_{(0,00)}$	0,02%$_{(0,00)}$	0,08%$_{(0,00)}$	0,01%$_{(0,00)}$	0,33%$_{(0,00)}$
n. 4006	1,10%$_{(0,02)}$	0,12%$_{(0,00)}$	0,02%$_{(0,00)}$	0,12%$_{(0,00)}$	0,02%$_{(0,00)}$	0,23%$_{(0,00)}$
n. 4311	0,26%$_{(0,00)}$	0,22%$_{(0,00)}$	0,02%$_{(0,00)}$	0,34%$_{(0,00)}$	0,01%$_{(0,00)}$	33,24%$_{(2,68)}$
n. 4705	0,23%$_{(0,00)}$	0,10%$_{(0,00)}$	0,03%$_{(0,00)}$	0,10%$_{(0,00)}$	0,02%$_{(0,00)}$	0,17%$_{(0,00)}$
n. 5505	0,34%$_{(0,00)}$	0,31%$_{(0,00)}$	0,02%$_{(0,00)}$	0,24%$_{(0,00)}$	0,02%$_{(0,00)}$	0,87%$_{(0,01)}$
n. 6206	0,06%$_{(0,00)}$	0,40%$_{(0,00)}$	0,01%$_{(0,00)}$	0,05%$_{(0,00)}$	0,01%$_{(0,00)}$	0,14%$_{(0,00)}$
n. 6707	0,09%$_{(0,00)}$	0,17%$_{(0,00)}$	0,02%$_{(0,00)}$	0,11%$_{(0,00)}$	0,01%$_{(0,00)}$	0,11%$_{(0,00)}$
n. 7706	1,25%$_{(0,01)}$	0,40%$_{(0,00)}$	0,03%$_{(0,00)}$	0,49%$_{(0,00)}$	0,02%$_{(0,00)}$	5,68%$_{(0,44)}$
n. 8107	0,19%$_{(0,00)}$	0,20%$_{(0,00)}$	0,03%$_{(0,00)}$	0,49%$_{(0,00)}$	0,02%$_{(0,00)}$	6,07%$_{(0,43)}$
n. 8807	0,17%$_{(0,00)}$	0,22%$_{(0,00)}$	0,02%$_{(0,00)}$	0,48%$_{(0,00)}$	0,02%$_{(0,00)}$	0,97%$_{(0,02)}$
Average	0,66%$_{(0,01)}$	0,30%$_{(0,00)}$	0,03%$_{(0,00)}$	0,36%$_{(0,00)}$	0,02%$_{(0,00)}$	9,94%$_{(0,77)}$

TABLE IX
STATISTICAL ANALYSIS '+' significant difference, '-' non significant difference, '=' same strategy

	GA+SA	GA+TS	PSO+SA	PSO+TS	$MMC_S^{G\&P}$	$MMC_T^{G\&P}$	$MMC_{S\&T}^{G}$	$MMC_{S\&T}^{P}$	$MMC_{S\&T}^{G\&P}$	$MMC_{S\&T}^{G\&P\&A}$	$MMC_{S\&T\&V}^{G\&P}$	$MMC_{S\&T\&V}^{G\&P\&A}$	EMC
GA+SA	=	-	-	+	+	+	+	-	+	+	+	+	+
GA+TS	-	=	-	+	+	-	+	-	+	+	+	+	+
PSO+SA	-	-	=	+	+	-	+	-	+	+	+	+	+
PSO+TS	+	+	+	=	+	+	+	+	+	+	+	+	+
$MMC_S^{G\&P}$	+	+	+	+	=	+	-	+	-	+	-	+	+
$MMC_T^{G\&P}$	+	-	-	+	+	=	+	-	+	+	+	+	+
$MMC_{S\&T}^{G}$	+	+	+	+	-	+	=	-	-	+	-	+	+
$MMC_{S\&T}^{P}$	-	-	-	+	+	-	-	=	-	+	-	+	+
$MMC_{S\&T}^{G\&P}$	+	+	+	+	-	+	-	-	=	+	-	+	+
$MMC_{S\&T}^{G\&P\&A}$	+	+	+	+	+	+	+	+	+	=	+	+	+
$MMC_{S\&T\&V}^{G\&P}$	+	+	+	+	-	+	-	-	-	+	=	+	+
$MMC_{S\&T\&V}^{G\&P\&A}$	+	+	+	+	+	+	+	+	+	+	+	=	+
EMC	+	+	+	+	+	+	+	+	+	+	+	+	=

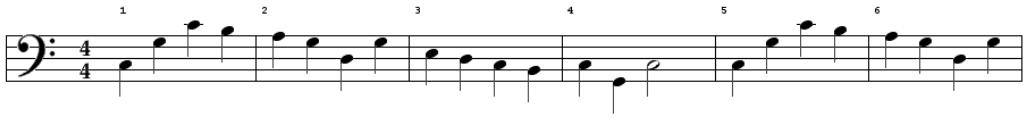

Fig. 3. Input of Bach's chorale BWV 25.6

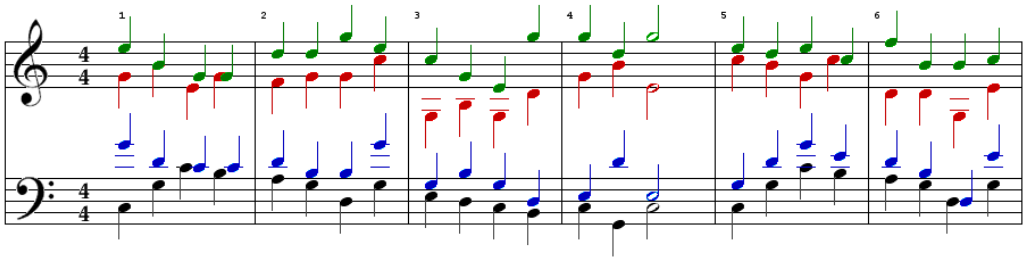

Fig. 4. A random solution of Bach's chorale BWV 25.6

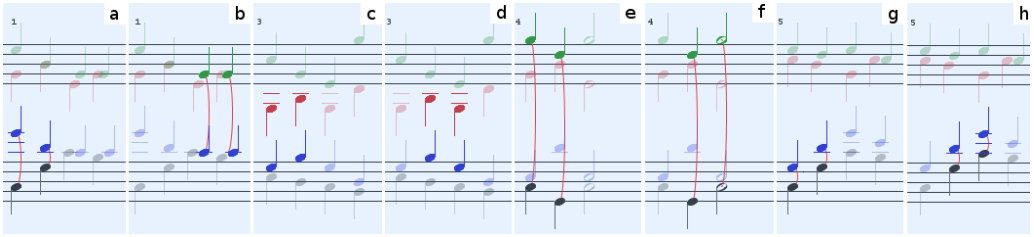

Fig. 5. Harmonization errors exhibited by the random solution.

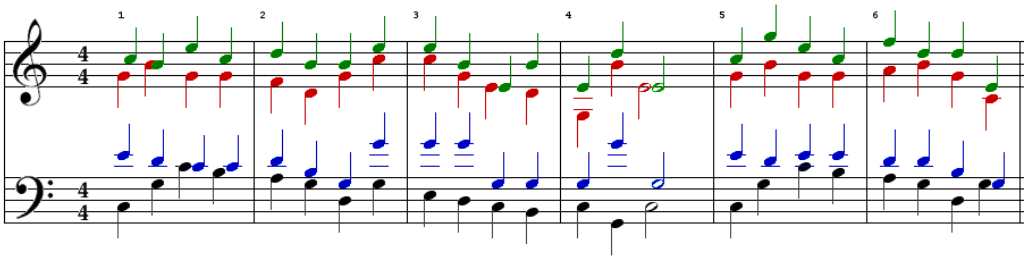

Fig. 6. A solution obtained by EMC.

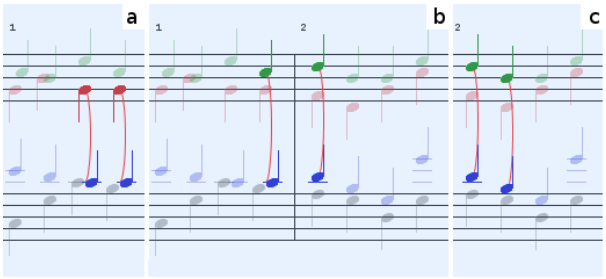

Fig. 7. Critical errors appearing in EMC's solution.

could arrive at a solution such as that shown in Figure 4. This solution was obtained during the initial steps of the resolution of the instance and, as a consequence, it seems quite chaotic at a first glance. In fact it exhibits many of the harmonization errors defined in Section III. And if we could hear it, it would sound discordant and unpleasant.

Figure 5 shows the critical errors that can be isolated in the random solution. A total of eight critical errors appear in the first 6 measures of the randomly harmonized piece. Together with these errors, 12 minor problems (classified as normal in Section III) also appear in the solution, but are not displayed. The notes involved in the errors are outlined using a darker color. The errors that can be appreciated are:

- Consecutive fifths. This error represents the progression of two voices in which an interval of a perfect fifth is

followed by a different interval of a perfect fifth. Appears in parts a, b, e, f, g, h of the figure.
- Unisons. This error, which can be seen in parts c and d of the figure, occurs when two voices move in the same direction by the unison.

Choosing different chords, among the available, or switching the notes interpreted by each voice, it is possible to improve the solution and eliminate the errors. Nevertheless, by doing so, other errors may appear. Figure 6 shows a solution found by EMC for Bach's chorale BWV 25.6. This solution was randomly selected among the solutions obtained employing EMC. It looks more organized than the randomly generated counterpart solution, and sounds better. All the critical errors appearing in the random solution have been removed, although new ones have shown up, as shown in Figure 7. Three critical errors can be appreciated in EMC's solution, together with 8 normal problems. These three errors include:

- A consecutive fifth, shown in part a.
- Parallel octaves. This error, which can be seen in parts b and c, occurs when two voices jump, in the same direction, from a smaller interval to an interval of a perfect octave.

If the appropriated technique is applied, it can eliminate all critical errors, and most, if not all, of the non-critical ones. Figure 8 shows a solution obtained by the strategy proposed in this article. This solution was randomly selected among the solutions obtained employing $MMC^{G\&P\&A}_{S\&T\&V}$. This solution does not present any of the previously defined errors. Consequently, it sound is more pleasant than that of the rest of solutions presented in this section.

D. Musical quality

As we have pointed out in the introduction, the quality of a musical composition is a subjective judgment. However, the solutions provided do not have any of the critical errors that guided the search performed by the proposed strategy.

We have also performed listening tests where we had people, with music backgrounds, listen to the pieces composed by our automatic composer. The average response that we had is that the compositions are sound and adhere to scholastic rule. A human composer would probably do better in terms of originality, but most people were surprised that a computer could actually produce such outputs.

The results obtained fulfill our expectations, returning solutions with notable quality and never discordant. Such results are possible because the cooperative metaheuristic is able to eliminate all critical errors and many of the errors allowed by the used harmonic rules.

VI. Conclusions

In this paper we have presented an automatic composer based on the collaboration of diverse memetic composer agents that mimics the process of multi-cultural evolution of music. This approach has been shown to perform creative musical composition without human intervention.

The specific problem under consideration is the figured bass problem, in which the objective is the generation of a 4 voice piece starting from bass line. We have tackled the problem using a strategy that employs different memetic composer agents, which employ different population-based and local-based metaheuristics as memes. The cooperation and competition among memes allows them to adapt to the changing characteristics of the instances of the problem, and facilitates the emergence of creative music pieces.

To test the validity of the strategy we have performed different tests obtaining interesting results.

Appendix A
Music Background

From a musical point of view, our proposal is based on the *tempered music system*, which is conventionally used among western musicians. This system models the musical notes by using a reference sound (a given frequency, normally 440Hz) and defining *octaves* as the ranges of frequencies between those sounds obtained by doubling/halving the reference sound. More precisely, if a note has frequency f the "next" note has frequency $f \cdot 2^{\frac{1}{12}}$. Each octave is divided in 12 equally spaced notes, denoted by the letters A, A♯ or B♭, C, C# or D♭, D, D# or E♭, E, F, F# or G♭, G and G# or A♭.

Tempered music is based on the notion of tonality. Roughly speaking, a tonality is a group of notes which form a scale. Using each one of the 12 notes of a scale as starting point we can obtain different tonalities (there are different kinds of tonality for each note, like major, minor, ...).

For example, C major scale is C, D, E, F, G, A, B, while D major scale is D, E, F#, G, A, B, C#.

The notes of a scale are often denoted as I, II, III, IV, V, VI, VII, specially when the emphasis is given to the degree of the scale and not to the particular note, which depends on the tonality. Such a notation is possible because all notes are equally spaced.

The interested reader should refer to [33] to get a deeper insight of tempered music.

Usually a tonality is considered the main tonality for a piece, and thus, the notes belonging to the corresponding scale are considered more important than those not belonging to it. Western music, starting from the common practice period, is based on well established harmonic and melodic rules for the tempered music system. We refer the interested reader to any good harmony book as [34] for a discussion about such rules.

A musical piece is composed of a sequence of *measures*, each one divided into a given number of *beats*. In each beat 4 independent voices play a note, *bass*, *tenor*, *alto* and *soprano*. The vertical set of notes in a beat is a *chord*. The notion of chord is fundamental for this work. There are many types of chords.

In this paper we consider 3- and 4-note chords. Chords are built upon each degree of the scale, that is, I, II, III, IV, V, VI, VII. The note upon which the chord is built, called *root note*, is often given to the bass; however the bass can play any other note which is called chord inversion. Chord inversions usually denoted by a superscript which indicates the inversion (e.g., III^6, V^{46}, I^{357}).

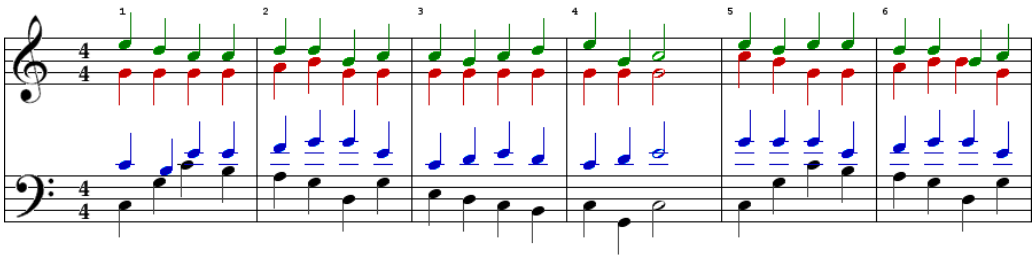

Fig. 8. A solution obtained by $MMC^{G\&P\&A}_{S\&T\&V}$.

The chords considered on this work are described in Table X (for each one of them we give an example in the tonality of C).

TABLE X
CHORDS CONSIDERED - WITH EXAMPLES IN THE TONALITY OF C

Chord	Root note	Set of notes
Major triad	C	C, E, G
Minor triad	E	E, G, B
Major seventh	C	C, E, G, B
Minor seventh	D	D, F, A, C
Dominant seventh	G	G, B, D, F
Half-diminished seventh	B	B, D, F, A

One of the main rules of tempered music system is that the vertical set of notes has to belong to any of the allowed chords. Other rules may affect sequences of chords. Some sequences are "better" than others, where better is difficult to define, as it is a subjective evaluation. In any case it is largely accepted that particular sequences of chords work better than others. Some chords are "more important" than others because they suggest, prepare, device or enforce tonal centers. The art of tonal music consists precisely in arranging chords in such a way that their interplay is pleasant and significant.

In practice, this is translated into simple rules as the following [34]:

I is followed by IV o V, sometimes VI, less often II o III.

...

...

II is followed by V, sometimes VI, less often I,III o IV.

Besides the rules concerning sequences of chords, we can also find rules about melodic lines. A melodic line is the sequence of notes played by each voice. Rules about melodic lines can refer to the movement of a single voice (for example, a jump bigger than an octave is normally not allowed) or also to the movements of two voices (for example, two voices that proceed by parallel fifth are not allowed). Again, we refer the interested reader to a standard textbook, like [34] for a better and more detailed explanation.

ACKNOWLEDGMENT

This work is partially supported by project TIN2011-27696-C02-02 of the Ministry of Economy and Competitiveness of Spain.

REFERENCES

[1] E.R. Miranda, *Composing Music with Computers* , Music Technology series, Focal Press, 2001.
[2] L. Fogel "Evolving Art," Proceedings of the First Annual Conference on Evolutionary Programming, San Diego, pp. 183–189, 1992.
[3] M. Dostl, "Evolutionary Music Composition," *Handbook of Optimization*, I. Zelinka, V. Snàšel, A. Abraham (Eds.), Intelligent Systems Reference Library, vol. 38, pp. 935-964, Springer Berlin Heidelberg, 2013.
[4] X. S. Chen, Y. S. Ong, "A Conceptual Modeling of Meme Complexes in Stochastic Search", IEEE Transactions On Systems, Man and Cybernetics - Part C, Vol. 42, No. 3, 2012.
[5] D. W. Johnson and R. T. Johnson, "Cooperation and competition: Theory and research", Edina, MN: Interaction Book Company, 1989.
[6] P. Moscato, "On evolution, search, optimization, GAs and martial arts: toward memetic algorithms," California Inst. Technol., Pasadena, CA, Tech. Rep. Caltech Concurrent Comput. Prog. Rep. 826, 1989.
[7] N. Krasnogor, J.E. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy and design issues," *IEEE Trans. Evol. Algorithms* vol. 9, no. 5, pp. 474–488, 2005.
[8] A. Torn and A. Zilinskas, *Global Optimization*. Berlin, Germany: Springer-Verlag, vol. 350, LNCS, 1989.
[9] Y. S. Ong, A. J. Keane, "Meta-Lamarckian in memetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 99–110, 2004.
[10] Y. S. Ong, M.H. Lim, N. Zhu, and K.W. Wong, "Classification of adaptive memetic algorithms: a comparative study," IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 36, no. 1, pp. 141152, Feb. 2006.
[11] L. Hiller, Computer music, *Scientific American*, 201(6):109–120, Scientific American Inc., 1959.
[12] L. Hiller, L.M. Isaacson, *Experimental music: Composition with an Electronic Computer*, McGraw-Hill, New York, 1959.
[13] R. Waschka, "Avoiding the fitness bottleneck using genetic algorithms to compose orchestral music," in Proc. of the International Computer Music Conference, pp. 201203, 1999.
[14] R. Waschka, "Composing with Genetic Algorithms: GenDash," *Evolutionary Computer Music*, E. Miranda, J. Biles (Eds.), Springer London, pp. 117–136, 2007.
[15] B. L. Jacob, "Composing with genetic algorithms," in Proc. of International Computer Music Conference, 1995.
[16] B. L. Jacob, "Algorithmic composition as a model of creativity," *Org. Sound*, vol. 1, pp. 157–165, 1996.
[17] J. H. Moore, "Gamusic: Genetic algorithm to evolve musical melodies," 1994
[18] B. Johanson, R. Poli, "GP-music: an interactive genetic programming system for music generation with automated fitness raters," in Proc. of the Third International Conference on Genetic Programming, 1998.
[19] L. Spector, A. Alpern, "Criticism, culture, and the automatic generation of artworks," in Proc. of the Twelfth National Conference on Artificial Intelligence, American Association for Artificial Intelligence, vol. 1, pp. 3–8, 1994.
[20] L. Spector, A. Alpern, "Induction and recapitulation of deep musical structure," in Proc. of the IFCAI 1995 Workshop on Artificial Intelligence and Music, pp. 41–48, 1995.
[21] J. A. Biles, "GenJam: A genetic algorithm for generating jazz solos," Ann Arbor, MI: MPublishing, University of Michigan Library, 1994.
[22] J. A. Biles, "Interactive genjam: Integrating real-time performance with a genetic algorithm," in Proc. of the 1998 International Computer Music Conference, ICMC, 1998.

[23] J. A. Biles, "Life with genjam: Interacting with a musical iga," in Proc. of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999.
[24] J. A. Biles, "GenJam: evolution of a jazz improviser," Morgan Kaufmann Publishers Inc., San Francisco, pp. 165–187, 2002.
[25] K. Thywissen, "Genotator: an environment for investigating the application of genetic algorithms in computer assisted composition," in Proc. of the 1996 International Computer Music Conference, pp. 274277. ICMA, 1996.
[26] P. M. Gibson, J. A. Byrne, "NEUROGEN: musical composition using genetic algorithms and cooperating neural networks," in Proc. of Second International IEEE Conference on Artificial Neural Networks, pp. 309–313, 1991.
[27] T. Unemi, "A design of genetic encoding for breeding short musical pieces," in Proc. of Workshop on Artificial Life Models for Musical Applications II: Search for Musical Creativity, pp. 25–29, 2002.
[28] T. Unemi, "Sbeat3: a tool for multi-part music composition by simulated breeding," in Proc. of the Eighth International Conference on Artificial Life, pp. 410–413, MIT Press, Cambridge, 2003.
[29] D. Ralley, "Genetic algorithms as a tool for melodic development," in Proc. of the International Computer Music Conference, Banff, Canada, pp. 501–502, 1995.
[30] D. Cope, *Experiments in Musical Intelligence*, Computer Music and Digital Audio Series 12, A-R editions, Madison, WI, 1996.
[31] D. Cope, *The Algorithmic Composer*, Computer Music and Digital Audio Series 16, A-R Editions, Madison, WI, 2000.
[32] D. Cope, *Virtual Music: Computer Synthesis of Musical Style*, MIT Press, Cambridge, 2004.
[33] G. Loy, *Musimathics vol. 1: The Mathematical Foundations of Music*, The MIT Press, Cambridge, 2006.
[34] W. Piston, M. DeVoto, *Harmony*, Norton, New York, 1987.
[35] Acampora, G.; Cadenas, J.M.; Loia, V.;E, Muñoz, "Achieving Memetic Adaptability by Means of Agent-Based Machine Learning," Industrial Informatics, IEEE Transactions on , vol.7, no.4, pp.557,569, Nov. 2011
[36] G. Acampora, J. M. Cadenas, V. Loia, and E. Muñoz, "A Multi-Agent Memetic System for Human-Based Knowledge Selection," IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 41, no. 5, pp. 946960, 2011.
[37] J.M. Cadenas, M.C. Garrido, E. Muñoz. "Using machine learning in a cooperative hybrid parallel strategy of metaheuristics," *Inform. Sciences*, vol. 179, pp. 3255–3267, 2009.
[38] P.P. Bonissone, J.M. Cadenas, M.C. Garrido, R.A. Díaz-Valladares. A fuzzy random forest. International Journal of Approximate Reasoning vol. 51, no. 7, pp. 729-747, 2010.
[39] T. G. Crainic, M. Gendreau, P. Hansen, N. Mladenovic, "Cooperative parallel variable neighborhood search for the p-median," *J. of Heuristics*, vol. 10, pp. 293–314, 2004.
[40] D. Henderson, S.H. Jacobson, and A.W. Johnson, The theory and practice of simulated annealing, *In F. Glover and G.A. Kochenberger (Eds.), Handbook of Metaheuristics*, Kluwer Academic, 2003.
[41] S. García, A. Fernández, J. Luengo, F. Herrera, A study statistical of techniques and performance measures for genetics-based machine learning: accuracy and interpretability, *Soft Computing*, 13(10):959–977, Springer, 2009.
[42] A. Globerson and T. Jaakkola, "Fixing Max-Product: Convergent Message Passing Algorithms for MAP LP-Relaxations," Advances in neural information processing systems, 2007.
[43] R. De Prisco and R. Zaccagnino, "An evolutionary music composer algorithm for bass harmonization," in Applications of evolutionary computing, Springer, 2009, pp. 567572.