GenOrchestra: An Interactive Evolutionary Agent for Musical Composition

Fabio De Felice, Fabio Abbattista.
VALIS Group, Dipartimento di Informatica,Università di Bari, Italy.
e-mail: febo74@libero.it, fabio@di.uniba.it

Francesco Scagliola.
Conservatorio di Musica "N. Piccinni", Bari, Italy.
e-mail: fscagliola@libero.it

Abstract

GenOrchestra is a project involving the Dipartimento d'Informatica and Conservatorio di Musica "N. Piccinni" in Bari. This project concern a Creative Evolutionary System, based on Evolutionary Computation (EC) techniques, applied to the field of western tonal music. With GenOrchestra a novel way to evaluate the produced tunes is presented: indeed we adopt a hybrid solution composed for two kinds of fitness functions. The first, called **technique fitness**, evaluates the consonance degree between melodic, harmonic and rhythmic sections, moreover, it defines how well the rhythmic paths is organized into a coherent musical event. The second fitness function called **human fitness**, determine how well the tunes are perceived from a human audience, like in a concert. This task is accomplished by presenting the tunes on the Internet and then gathering the surfers evaluations in a database from which the system take the final population scoring. This, coupled with a no limited musical primordial soup, makes GenOrchestra a promising eclectic artificial composer. The ultimate goal of this project, currently in progress, is the development of a very human-like composer, which can produce music in any musical genre, and which is able to show a "personal style". Samples will be soon available at http://valis.di.uniba.it/GenOrchestra/samples.html

1. Introduction

Today, many systems exist which exhibit human behaviors: from natural language dialogues abilities to art production and feeling expression. Several examples can be found in music [4]. One of the most cited is GenJam, developed by J.A. Biles at Rochester Institute of Technology, GenJam (Genetic Jammer) is an interactively G.A. (IGA) that evolve jazz solos on a given chord paths and is able to duets with human players connected with the system via Midi.
Another interesting system is Conga, developed by N. Tokui of Tokyo Institute of Information and Comunication Engineering and professor H. Iba of Graduate School of Frontier Sciences of Tokyo University. Conga is an interactive system that, combining GA and Genetic Programming (PG), evolves bass/drum rhythmic sequence.
Vox Populi, developed by A. Moroni, Technological Center for Informatics, J. Manzolli, F. Zuben and R. Gudwin of University of Campinas, Vox Populi is a real time music composer that generates a chord coded into the MIDI standard and then evolves it by using the GA paradigm.
In this paper we present a project, involving the Dipartimento di Informatica of the University of Bari and the Music Conservatory "N. Piccinni" of Bari, aimed to the development of an e-

learning web system applied to the field of western tonal music. Currently we have developed the sound engine called GenOrchestra (**Gen**etic **Orchestra**). GenOrchestra is based on a Genetic Algorithm (GA) [6] and presents some modification of the standard GA paradigm. The aim of Genorchestra is the automatic tunes composition, starting from parameters concerning the structure, the number of measures per section, the starting beat, the playing tempo and starting scale. These parameters will be set by the user or automatically chosen by the system itself, by means of pseudorandom rules.
The remainder of the work is structured as follows: Section 2 and 3 describe in more details the GenOrchestra general architecture and the underlying genetic algorithm. In Section 4 the fitness function adopted in Genorchestra is discussed while in Section 5, some results are reported. In Section 6 we briefly describe how to integrate human evaluations with the genetic algorithms evaluation and, finally, in Section 7 some conclusions are drawn.

2. The Architecture of GenOrchestra.

The GenOrchestra underlying evolutionary process has to be an Open-Ended one, i.e. a continued evolution to relative maximum points without nor temporal ending neither evolution to a single absolute maximum point. The main feature of the system is its generality concerning the faceable musical genres and the capacities of self-judging the composed tunes by evaluating the melodic, harmonic and rhythmic qualities based on the ordering of consonance of musical interval. The aforesaid evaluations procedures are integrated with the web surfers and expert human composers evaluations, through the GenOrchestra site. The repeated iteration of these phases should make emerge particularly ways to equilibrium points, to an intelligent musical behavior, to a "style".
The general architecture is composed of six modules (Fig. 1):

- **Composer**: this module handles the system compositional process. It receives the tunes features and the GA parameter from the user and then starts the evolution. This module is strictly correlated with the Maestro module giving it the composed tunes and receiving from it the fitness scores. Moreover, the Composer communicates with the Feedback module for the user evaluation of the tunes.
- **Maestro**: This module embeds the overall consonance fitness functions. Input to this module are the user tunes submitted via web and the tunes produced by the Composer. The Maestro module produces as output the relative fitness scores.
- **Feedback**: The Feedback module is responsible for the human evaluation of composer tunes. It shows the produced tunes on the web, retrieves the surfers evaluations, defines the scores per tunes and sends these values to the Composer module.
- **Arranger**: The Arranger takes the user tunes, submitted via web, and applies musical transformations in order to arrange the musical materials.
- **Learning**: This module handles the pure documentation side of the whole system. It handles the Docs database in which downloadable materials, concerning music theory and computer music, is stored. This module makes a search in the database following the user query.
- **Web Site:** It makes up the system Internet interface; through this interface users can listen the music produced and they can evaluate it, explaining the evaluations through the guestbook and, so, influencing the future musical production.

The Composer module is mainly based on genetic algorithms [6]. Many variations to the original GA paradigm have been proposed in the last years; GenOrchestra is based on a Steady-state GA with tournament selection and multi-cut points crossover. In a Steady-state GA, every new population presents an overlapping between the old and the new generations. In the tournament selection the population is grouped in several individual families, the best two individuals of each family mate with the crossover operator and the new solution substitutes those in the family with worse fitness. Concerning the crossover operator, GenOrchestra applies a multi cut points operator. In our implementation, a cut point is chosen

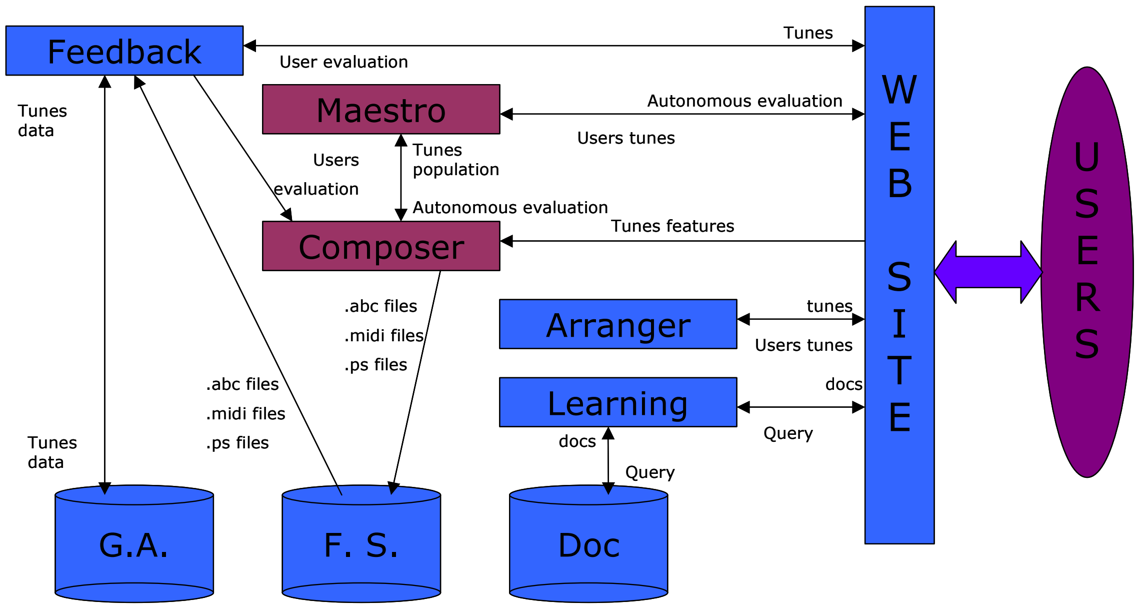

Fig 1. The general architecture of GenOrchestra

for every section in the tune structure and for every layer. The cut point has not been chosen at low level (such as the note or the chord), but at the measure level to avoid the production of new individuals with length measure that differs from the parents one.

3. The chromosome representation in GenOrchestra

In the genetic algorithms, the representation of a solution is usually called the chromosome. The goal of GenOrchestra is to continually evolve populations of tunes, so the chromosomes have to reflect the structure of a musical piece. We can simplify a piece of music as a significant set of sections, differing each other by the melodic theme and possible scale, time and beat variations. Every section can be repeated in the tune execution so if we have three sections, A, B and C, then the structure can be any disposition with repetition of these three sections. Furthermore a tune has some initial features such as: Scale, beat, note unit length (eight note, half note etc.) and playing tempo of the note unit length. These values can be different in a given section and from section to section. Every section is made up of a certain number of measures; a measure is a not unique set of notes where the overall length is equal to the beat value. What we hear in a piece of music is, usually, made up of three sonorous layers: a bass layer, a harmonic layer and a melodic layer. In the first layer we have the bass score, in the second layer we have the chords score and in the last one the tune theme or solo score. So, the chromosome is defined as an array made up with so many components as the

sections in the structure, each of these components points to a three-layered structure containing the aforesaid scores.
The chromosome generation starts from the initial scale; on the ground of this scale the initial chord is built up with a random generated length that cannot exceed the length of the measure. On this first chord a melody with the same length and bass score is generated, then a melodic note is randomly chosen for a new chord with the same initial scale. This process is repeated till the length of the measure is reached; then it is repeated for the number of measures of the current section and for any section of the tune (see fig. 2).
On the ground of the described chromosome, we decided to adopt the same approach as in [1] for the mutation phase that is musically meaningful mutations operators that work at measure

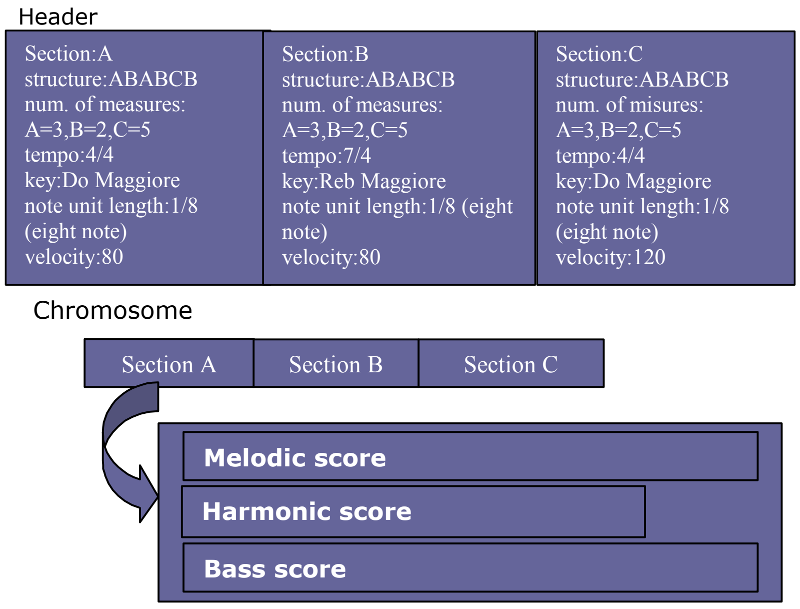

Fig. 2 Chromosome structure for a three-section tune

level. These operators implement classical compositions techniques. With a given mutation probability a chromosome is scanned and measures are chosen for the mutation in every layer. The mutation is randomly chosen among the following 6 types:

- Transposition: Transposes notes and chords in a measure, by a random number of intervals in the given scale. If a note is transposed beyond the allowed range, the count continues according to the scale interval in the upper octave, ignoring rests.
- Reverse: Reverses the events in a measure, rests included.
- Rotate–right: Rotates the events in a measure by a random number of positions to the right.
- Invert: Given an event in the measure, it evaluates the difference between the top position scale note (7) and the scale position of the current note.
- Sort up and sort down: sort the measure and preserve the rhythmic structure
- Invert-reverse: Given a measure, the invert and reverse operators are applied consecutively.

4. The autonomous evaluation of chromosomes

The fitness function used in GenOrchestra is a hybrid solution formed by an autonomous evaluation, judging the consonance qualities among melodic, harmonic and bass layers and a human evaluation for the aesthetic qualities. This approach would be a suitable solution to a critical phase in every musical GA. Indeed actual systems implements the fitness phase by two ways:

a) Completely delegating the individual evaluations to the human ears: this leads to great human-like musical production, but make up a heavy bottleneck for the system and a dull work for the human judge.
b) Adopting autonomous solutions like neural networks [2], implementing physiological aspects in listening music [7] and completely removing the fitness phase [3].

None of these solutions fit the GenOrchestra general purposes but the former resolve the drawbacks of the latter. So, a hybrid solution seems to be a good alternative, moreover it best reflects what happen in the real world.
Indeed, the GenOrchestra consonance fitness function is:
Consonance fitness = melodic-harmonic consonance score + harmonic consonance score + bass-melodic consonance score + bass-harmonic consonance score
To develop the fitness function we start from the fuzzy approach described in [7, 8], where a consonance measure among notes in a chord was defined and we extend it to a consonance measure of notes over chords. By means of this approach we can represent a note as a compound tone consisting of its fundamental tone and upper harmonic series tones. It can be represented as a fuzzy set in which the membership degree of a given tone is proportional to its amplitude. Finally, a note is a fuzzy set made up of couples (x, y) in which *x* is a tone (also called partial), and *y* is the related weight in the note, corresponding to its amplitude. We can now define the consonance between two notes Sm and Sn as follows:

$$\text{(3.4.1)}\quad Co(S_m, S_n) = \sum_{(x,y)\in S_{m\cap n}} y \qquad (1)$$

The consonance measure between two notes is intended as the sum of the intersection of the partials weights, in the range [0,..,1].
Starting from this concept we have defined a set of evaluations to carry out the overall consonance of the tune. We can formalize a note as a couple (pitch, length) and a chord as a set of three notes of the same length but with differing pitches. So, if we have a melodic series of notes $M=\{(m_1,t_1),...,(m_n,t_n)\}$, a series of harmonic set of chords $H=\{(A_1,t_1),...,(A_d,t_d)\}$ where $(A_d,t_d)=\{(a_1^d,td_d),(a_2^d,t_d),(a_3^d,t_d)\}$ and a bass series of notes $R=\{(r_1,t_1),...,(r_z,t_z)\}$, we can define several consonance score functions:

- note-chord consonance score:

$$NC(p,A_d) = \frac{\sum_{i=1}^{3} Co(p,a_i^d)}{3} \qquad (2)$$

, where p is the pitch note;

- chord-chord consonance score:

$$CC(A_i,A_{i+1}) = \frac{\sum_{j=1}^{3} NC(a_j^i,A_{i+1})}{3} \qquad (3)$$

- melodic-harmonic consonance score:

$$F_M = \frac{\sum_{j=1}^{d} \frac{\sum_{i=1}^{n} NC(m_i, A_j) * t_i}{t_j}}{d} \tag{4}$$

- bass-harmonic consonance score:

$$F_R = \frac{\sum_{j=1}^{d} \frac{\sum_{i=1}^{z} NC(r_i, A_j) \cdot t_i}{t_j}}{d} \tag{5}$$

- melodic-bass consonance score:

$$F_{MR} = \frac{\sum_{i=1}^{n} \frac{\sum_{j=1}^{z} Co(r_j, m_i) \cdot t_j}{t_i}}{n} \tag{6}$$

- harmonic consonance score:

$$F_H = (\sum_{i=1}^{m-1} CC(A_i, A_{i+1})) + CC(A_1, A_m) \tag{7}$$

- total consonance score:

$$F_C = F_M + F_H + F_R + F_{MR} \tag{8}$$

Using these functions we carried out the consonance degree between every chromosome layers.
Lets now describe the basic concepts and the resulting function for the rhythmic evaluations of the tunes composed. When we listen to a piece of music we naturally organize the sound signals into meter groups. Furthermore, we infer a regular pattern of strong and weak beats to which relate the actual musical sounds. GenOrchestra evaluates these patterns to judge how well the tune matches the metrical structure defined by the starting meter input. It must be emphasized that beats do not have duration, and we can think about them as an idealization, used by the performer and inferred by the listener from the musical signal. To use a spatial analogy: beats correspond to geometric points rather than to the line drawn between them. But, of course, beats occur in time so an interval of time takes place between successive beats. For such intervals we use the term time-span. Because of the afore said analogy, we can represent beat by dots.

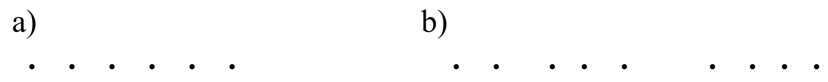

Fig. 3. Beats sequences examples

The two sequences differ in a crucial respect: the dots in the first sequence are equidistant but not those in the second. The meter function is to mark off, insofar as possible, into equal time-spans, this disqualifies the b) sequence from being called metrical. Another aspect of meter is the notion of periodic alternation of strong and weak beats, in a) sequence no such distinction exists. For beats to be strong or weak there must exist a metrical hierarchy. The relationship of strong beat and metrical level is simply that, if a beat is felt to be strong at a particular level, it is also a beat at the next larger level. This is shown in the following figure.

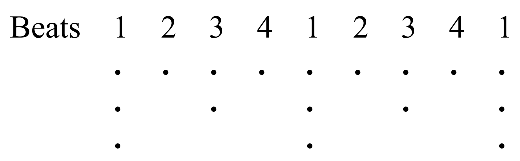

Fig. 4. Metrical structure for a 4/4 meter

So, given a note unit length and a starting meter we can built up the relative metrical structure as follows:

- Define a first note unit length level formed by a number of beats calculated by the following formula:

$$beats = round\left(\frac{num\_val * val\_mov}{unit}\right) \tag{9}$$

  Where *beats* is the number of beat per level, *num_val* is the number of movimento per meter, *val_mov* is the value of the meter, the *unit* is the note unit length and, finally the *round* function round up the ratio result to the next integer, if it is a float.
- While *beat* is greater or equal to 1:
  - Duplicate the *unit* and then calculate the (9) again

We referred to this structure as **perfect metrical structure.** Given this structure, we defined the metrical patterns for every measure in a given tune defining how many pitches start time occur in a given time-span. We refer to this pattern as the **actual metrical structure**. Then the closer the **actual metrical structure** to the **perfect metrical structure** is the better the evaluation.

5. Experimental results

To verify the effectiveness of the GA in evaluating the produced tunes, we performed three different experiments. In the first set of experiments we fixed the parameters concerning the tune features, as described in Table 1 and set different values for GA parameters, as reported in Table 2.
For each configuration of GA parameters, we performed 10 runs with the same population size (100 individuals) but with different initial population. The stop criterion was the

Tune features	*Values*
Structure	**A**
Measures num.	**10**
Beat	**4/4**
Tempo	**40**
Note unit length	**Half note**
Starting Scale	**C Major**

Tab. 1 Tune features for the first experiment

Experiment	*Crossover Prob.*	*Mutation Prob.*	*Generations*
1	0.7	0.2	100
2	0.5	0.2	100
3	0.7	0.3	100
4	0.5	0.3	100
5	0.7	0.4	100
6	0.5	0.4	100

Tab. 2 GA parameters for the first experiment

maximum number of generation allowed (100 in our experiments). The evolution to an effectiveness euphonic music production has been tested with the help of a human composer. As Table 3 shows, the best results have been obtained in the second run, corresponding to a crossover probability equal to 0.5 and mutation probability equal to 0.2. Figure 5 plots the trend of the fitness with respect to the number of generation for the best resulting run. As it can be seen, in the first 50 generations the algorithm evolves very quickly and the fitness increment is near equal to 0.8, while in the second half of the run, the algorithm is quite stable. Tunes produced in the initial population shows very disorganized paths with frequently changes of note and chords lengths, while the best tunes in the last generation show a more relaxed musical events distribution and an effectively more euphonic theme.

Experiment	**Generation**	**Best Fitness**	**Average Fitness**
1	83	1.157228	1.070756
2	**96**	**1.384570**	**1.275823**
3	92	1.202566	0.971285
4	88	1.199713	1.002079
5	49	1.156400	0.935947
6	65	1.035057	0.815731

Tab. 3 Results of the first experiment

In the second set of experiments we used the best the GA parameters found in the first experiments (crossover probability=0.5 and mutation probability=0.2), and set different values for the tune features, the beat values were 3/4, 5/4 and 7/4, the tempo values were 4 and 8, while the values for the other features remain the same of the previous experiment.

For each configuration of tune features we performed 10 runs with the same population size (100 individuals) but with different initial population. As in the first experiment, the stop criterion was the maximum number of generation allowed (100 in our experiments), and the evolution to an effectiveness euphonic music production has been tested with the help of a human composer. As Table 4 shows, the best results have been obtained in the sixth run, corresponding to a Beat of 7/4 and a Note unit length of 8. Figure 6 plots the trend of the fitness with respect to the number of generation for the best resulting run. As the figure shows, in this run the evolution is faster than in the preceding experiments (the maximum fitness value is near 2.0 within the same number of generations). Furthermore, from the graph it can be seen how the maximum fitness score is reached at the end of the evolution, with the possibility of a more high values with more generations.

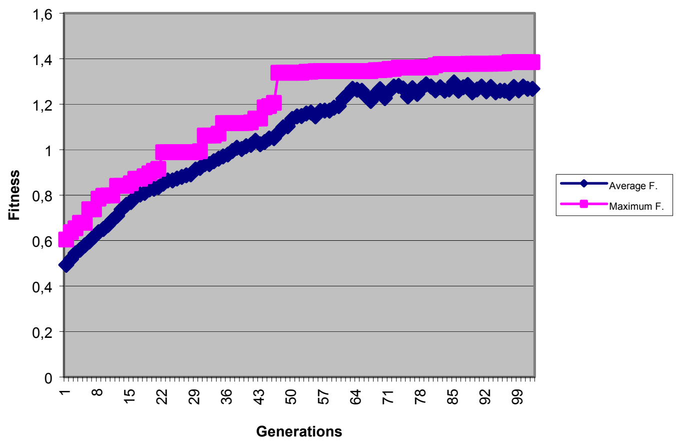

Fig. 5 Best run of the first experiment

Experiment	**Generation**	**Best Fitness**	**Average Fitness**
1	98	1.104048	1.007800
2	95	1.172145	1.090072
3	96	1.101545	1.044355
4	95	1.326399	1.158015
5	96	1.244252	1.146731
6	**92**	**1.923617**	**1.581887**

Tab. 4 Results of the second experiment

To verify this last statement, we decided to repeat the better runs of the two preceding experiments (plotted in Figures 5 and 6), setting a higher generations number (1000). In these further runs we have reached a better fitness score than in the runs limited to 100 generations, but the slight fitness improvement does not seem to justify the strong time effort required. However, in the last generations more individuals with the same fitness score but with different musical structures have been found. This result is consistent with the Open-ended approach and is justified by the fact that identical consonance scores can be reached with different melodic-harmonic-rhythmic structures.

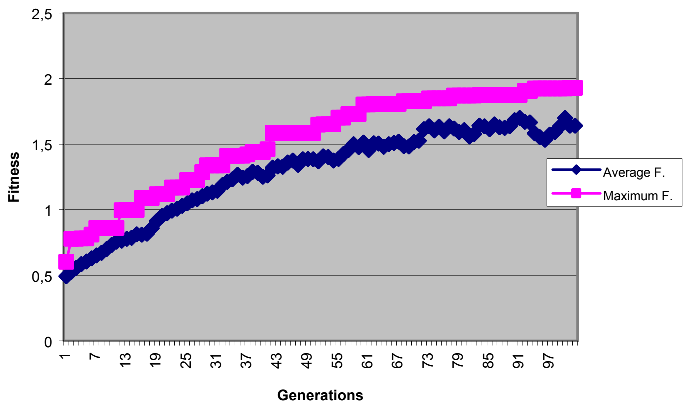

Fig. 6 Best run of the second experiment

6. The human evaluation of chromosomes

The results showed so far refer to the automatic evaluation of the tune features. Nevertheless, a musical composition should be evaluated for its aesthetic qualities. GenOrchestra integrates the GA fitness function with the human evaluations of the produced tunes. In fact, the chromosomes produced by the GA are made available on the GenOrchestra Web site and, users accessing the site, can listen them and provide their subjective scores. Users scores are averaged, for each of the evaluated tunes, and are summed up to the GA evaluations. The involvement of human users is an effective solution to the subjective evaluation of the tunes, but, on the other hand, it represents a bottleneck of the GenOrchestra system, due to the time consuming. To overcome this limitation, we assigned a fixed interval of time for the user evaluation of each generation of tunes. Indeed, not the whole population available on the web will receive a user evaluation for the aforesaid drawbacks. This lead to a speciation of the initial population P after the evaluation phase: the population P_u made up with individuals evaluated autonomously and via web and the population P_t of autonomously evaluated tunes passed unseen on the web. Consequently, each run corresponds to two separate evolutionary processes allowing the selection operator to work on individuals with comparable and homogeneous fitness. The separated populations merge into one after the mutation phase, ready for a new iteration of the GA. It should be noticed that there is no correlation among individuals belonging to the population P_u (respectively P_t), in subsequent generations. Experiments with the web evaluation have been performed in a controlled situation where users are known and their accesses to the site have been monitored. In a few days, we will launch a more wide experiment, allowing the access to the site from the whole Web.

7. Conclusions

In this paper we have described a prototype of an evolutionary based system able to autonomously produce tunes presenting a good consonance degree, as confirmed by a human

expert. However, a main weakness of the system is that the good work of the fitness function adopted is not noticeable to the novice human hear and the produced tunes do not yet correspond to a really human-like musical composition. In the future, several others tunes characteristic have to be studied and formalized into our fitness function.
Comparing GenOrchestra with other creative evolutionary systems we can conclude that it is a more complete system because of its goal (to generate a complete tune) but, on the other hand, currently, GenOrchestra produces a not enough human-like though consonant musical output; in fact a human composer is still needed to arrange the musical output into a finished thematic development.
Further development will be:

- A user tunes consonance evaluation module, by which the system evaluates human composition with the aforesaid function, so an evolution to reach that value can be made.
- Formalization of a given musical genre by means of the above features, in order to make the system able to compose in a given style, without any extensive knowledge.
- Last, we intend to overcome the limitation imposed by the web-based evaluation, implementing an aesthetic wit based on an emotive component, able to emulate a human listener and to feel human-like emotions.

References

[1] J. Biles, GenJam : A Genetic Algorithm for Generating Jazz Solos. In Proc. of the International Computer Music Conference, 1994.
[2] J. Biles, P. G. Anderson, L. W. Loggi, Neural Network Fitness Functions for a Musical IGA. In Proc. of the International ICSC Symposium on Intelligent Industrial Automation and Soft Computing, 1996, pp. B39--B44.
[3] J. Biles, Autonomous GenJam: Eliminating the Fitness Bottleneck by Eliminating Fitness, Workshop on Non-routine Design with Evolutionary Systems, Genetic and Evolutionary Computation Conference, 2001.
[4] A. Camurri, P. Ferrentino, R. Dapelo, A Computational Model of artificial emotions. In Proc. of Kansei-The Technology of Emotion, AIMI International Workshop, 1997, pp 16-23.
[5] J. Cassell, J. Sullivan, S. Prevost, E. Churchill, Embodied Conversational Agent. The MIT Press, 2000.
[6] J. Holland, Adaptation in natural and artificial systems. Ann Arbor: The University of Michigan Press, 1975.
[7] A. Moroni, J. Manzoulli, F. Von Zuben, R. Gudwin, Vox Populi: Evolutionary computation for music evolution. In P.J. Bentley, D. W. Corne, *Creative evolutionary Systems*, Morgan Kaufmann, 2002, pp. 205-221.
[8] G. Vidyamurthy, J. Chakrapani, Cognition of tonal centers: A fuzzy approach. *Computer Music J.* Vol. 16, n° 2, 1992, pp. 45-50.