Exploiting Interactive Genetic Algorithms for Creative Humanoid Dancing

Adriano Manfré[1], Agnese Augello[1], Giovanni Pilato[1], Filippo Vella[1], Ignazio Infantino[1,*]

Abstract

The paper discusses an approach aimed at endowing a cognitive architecture with artificial creativity capabilities in order to make a humanoid able to dance in a pleasant manner. The robot associates movements to music perception creating an aesthetically valuable dance by using a Hidden Markov Model with a nonclassical approach. Two matrices mainly influence the model: a Transition matrix TM, and an Emission Matrix EM. The TM matrix rules the transition between two subsequent movements. The EM matrix constitutes the link between a set of movements and the perceived music features. In order to compute the EM matrix, we exploit a genetic algorithm approach. The approach makes use of two kinds of fitness functions. The first one is an internal evaluation fitness that allows the robot to autonomously learn the association between music and movements. The second one depends on the interaction with a human teacher, leading to the determination of different dance styles, which constitute the robot repertoire. The experimental part discusses the effects on the creativity of different distances to compute fitness.

Keywords: robot, dance, computational creativity, music perception, co-creative tool

---

[*] Corresponding author

Email address: `ignazio.infantino@cnr.it` (Ignazio Infantino)

[1] Institute of High Performance and Networking (ICAR), National Research council (CNR), Via Ugo La Malfa,153, 90146, Palermo, Italy.

1. Introduction

In recent years, interactive and social robots have gained the attention of media and are becoming also accessible to generic end-users. Various reported experiences demonstrate the proficient use of robotic platforms in education, health, entertainment, and other fields where human presence and interaction is an important aspect. Future robotic companions have to enhance their *humanity*, by improving artificial cognitive capabilities, and sharing knowledge processes with people. Computational creativity paradigm, see for example Colton et al. (2011), supported by a cognitive architecture (CA) can pursue such aims because considers relevant aspects of human intelligence and behaviour. Artefacts and artistic performances deriving from creative processes represent powerful vehicles of emotions, intentions, thinkings, and knowledge. If an artificial system aims to be judged capable of creativity by a human, probably all the previous aspects have to be part of a suitable cognitive architecture. Naturally, biological inspiration could be at the level of abstract functionality, or at the level of basilar mechanism, but the final aim is, as stated by the Turing test, to be credible to human beings. A cognitive architecture defines all components required to emulate the interaction between perception, memory, learning, planning and action execution, Goertzel et al. (2010) . Introducing creativity in CA implies to take into account evaluation processes, both exploiting internal and external mechanism that drive the creative process. Creativity cannot ignore external influence given that its mechanisms of association, analogy, and composition strongly depend on the evaluation process, Boden (1998). Creativity is completely subjective and therefore it requires detailed feedback from experts. In these systems, there are two kinds of evaluation: *internal*, related to a self-evaluation, and *external*, bound at audience feedback. In Colton et al. (2011) and in Colton et al. (2012) authors try to formalize artificial creativity to define creative methodologies and metrics, Galanter (2012), Jordanous (2012). How an embodied artificial agent able to interact with humans and the environment might produce creative processes or creative acts through a given cognitive architecture? Jordanous (2012) indicates 14 key components to take into account in order to deal with creativity, and some of them could be partially considered in various components of a CA: *dealing with uncertainty; intention and emotional involvement; generation of results; domain competence; originality; social interaction and communication; variety, divergence and experimentation; value; evaluation*. Other factors are more difficult to manage, requiring more investigation and better definition for real implementation:*active involvement, persistence, general intellectual ability, progression and development, spontaneity and subconscious processing, independence, and freedom, (abstract) thinking*. Several cognitive architectures utilise methods of self-evaluation: the easiest method is to compare the expectation value with the obtained result; these mechanisms can assure valid results in aesthetic sense, Romero et al. (2012). Furthermore, it is essential to capture an external evaluation to obtain an objective feedback about the generated output from an audience.

The present paper focuses on the learning process that determines the creative behaviour. By using an interactive genetic algorithm, an expert or teacher of the given chosen artistic domain influences the evolution according to his/her judgments on the performance. The paper tries to explore the potentiality of computational creativity in the domain of dance using various fitness distances. In Manfrè et al. (2015), we first proposed a robust mechanism allowing a robot to associate elementary movements with perceived music. The explained learning module has been embedded in a cognitive architecture that supports a general framework of computational creativity, Augello et al. (2015b), and it has been tested in various live performances[2]. The cognitive architecture is flexible and extensible, and we have used it also in other artistic domains, see for example Augello et al. (2015a).

---

[2] Robodanza web pages http://www.pa.icar.cnr.it/scarlab/robodance/

1.1. Aim and motivations

A typical human reaction when someone hears a rhythmic sound is reacting spontaneously through body movements synchronized on the musical rhythm. In fact, when we listen to a song, often unconsciously nod our head, beat feet, beat our hands and fingers on the table or on our legs, Todd et al. (2002).

In Toiviainen et al. (2010), authors investigated the movement induced by music in a human beings , focusing on the relationship between patterns and metrical levels of music, revealing that humans embody musical stimuli at several metrical levels.

A similar research has been pursued by authors in Xia et al. (2012); instead, Seo et al. (2013) focus their research on simple rhythmic movements like head nodding or hand shaking.

Usually, for people, it is very difficult or even impossible to dance on a choreography without any practice, but anyone can keep the rhythm with some simple movements. On the other side, it is very simple to program a robot that dances on a choreography with predefined movements. While it is not easy for robots synchronizing and automatically generating sequences of motions well formed for suiting different music genres. An introduction dealing with concepts about dancing robotic system is illustrated in Shinozaki et al. (2007).

It is very hard to reuse a pre-programmed choreography on a different song because it generally requires reprogramming of all the movements sequence to ensure the robot stays in sync with the new song.

In this paper, we propose a creative robotic system able to dance. The robot learns some associations between music and movements through an internal mechanism of evaluation, and by mean of a genetic algorithm, it obtains the creativity. Finally, its originality, naturalness, and the whole performance are judged by the audience.

The evolution is given by an external fitness function, and its value is assigned by a human. The approach is classified in literature as Interactive Genetic Algorithm or Interactive evolutionary computation (IEC), Takagi (2001), and it is used to obtain an aesthetic selection.

The robot is able to autonomously follow the rhythm of the music by emulating a human dancer that coordinates the movements with perceived rhythm. This is obtained by the use of a cognitive architecture loosely inspired to the Psi architecture, Bartl & Dörner (1998). Internal and external evaluation processes, Augello et al. (2015b), drive the variation in order to perform aesthetically acceptable execution, Augello et al. (2013),Augello et al. (2014).

The paper is structured as follows: the first section describes how to introduce a creative process in the Psi cognitive architecture; the second section deals with dance creation, and in the last section the experiments, performed in front of a live audience are reported.

2. Cognitive Architecture of the Humanoid Dancer

The design of a computational artist involves the modeling of several cognitive capabilities and representation levels. Indeed, although creativity is mainly perceived as a high-level cognitive process, Boden (2009), it is also based on low-level mechanisms that manage perceptions and representations, Bach (2009). We have modeled the humanoid dancer behavior and its autonomous, creative capabilities into a proper cognitive architecture. The cognitive architecture is loosely inspired by the Psi model proposed by Dietrich Dörner ,Bartl & Dörner (1998), and it is suitable to model aspects such as motivation and emotions that are integrated with perceptual and reasoning processes Bach (2009).

Cognitive architecture modules deployed in robot platform are shown in Figure 1), and takes into account the peculiarities of the robotic platform used for experiments. The Long Term Memory LTM stores repertories of dance behaviors through emission matrices EMs, and transition matrix TM. During the learning phase, the planning/execution module activates the music perception process, and the dance execution using an HHM approach. The EMs are selected by a teacher by judging some simulation of promising dances at each step of the evolution of the genetic algorithm, Vose (1999). The LTM stores a set of 20 elementary movements, and the TM determines the transition from a movement to another. The movement set and the initial TM are designed by a choreographer. If the real robot during dance execution encounters a problem (e.g. it falls, or the motor temperature is too high), variations on TM values could exclude some movement combinations.

The learning process is directly linked to computational creativity paradigm given that it assures variability and unpredictability thanks to various external evaluations and stimuli arising from the real environment. Teacher evaluations drive learning and determine the dancing style of the robot. The judgments of the audience influence Competence and Certainty parameters, as explained in Augello et al. (2015b), determining motivation, mood, and the dance execution of the robot. Moreover, GAs have been widely used in literature to emulate the results of human creativity into artificial systems, Dostál (2012),Biles (1994).In particular, in the case of music generation, one of the first systems is the one proposed in Horowitz (1994). Furthermore, the use of GAs perfectly fits the four-strategy model of creative interaction proposed by Tubb, in Tubb & Dixon (2014) which outlines interactions between implicit and explicit thinking and between divergent and convergent thinking, Augello et al. (2015a).

In the following, we introduce all the cognitive architecture modules used to create real live performances to better understand the approach used for learning. We implemented the approach in the context of a real embodied agent with physical needs. In general, the activity of the robot is influenced by its needs or primary drives, Bach (2009), and some parameters stored in Short Term Memory STM. The arrows depicted in figure 1 indicate the influences of the environment (including humans), and of the physical conditions of the robot itself . The *Physiological needs* lead the system to satisfy a physiological demand; for example, the need to charge its batteries. The social urge named *Affiliation* triggers a behavior aimed to a socialization with other agents or with human beings. In the specific case, the affiliation urge is satisfied if the robot detects at least one human face. Such aspects extend the predefined *autonomous* behavior of our robot. The cognitive urges Competence and Certainty are particularly relevant for the robot performance. Competence is defined as *the effectiveness*

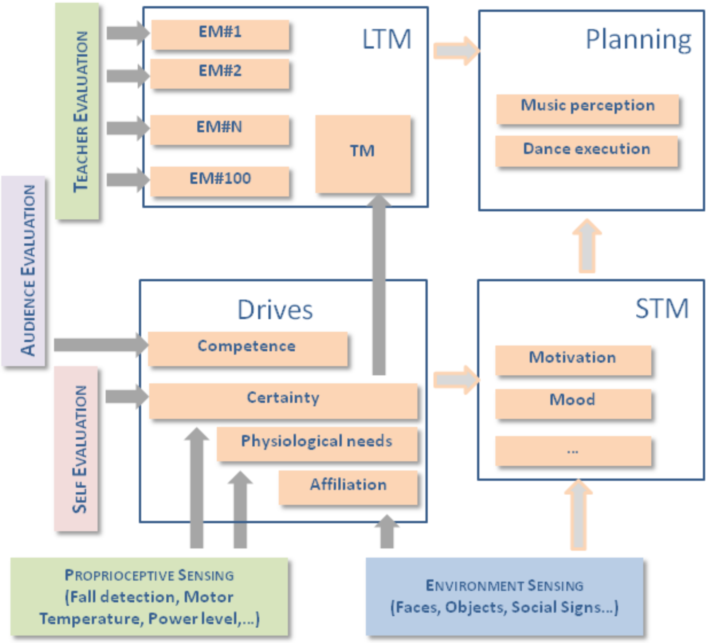


Figure 1: The cognitive architecture modules deployed in robot platform. Long Term Memory LTM stores repertories of dance behaviors through emission matrices EMs, and transition matrix TM. During Learning phase, the planning/execution module activates music perception process, and dance execution using HHM approach. In general, the activity of the robot is influenced by its needs or primary drives, and some parameters stored in Short Term Memory STM. The arrows indicate the influences of the environment (including humans), and of the physical conditions of the robot itself.

at fulfilling the agent needs and Certainty as *the confidence of the agent knowledge*, Cai et al. (2011). In modeling the humanoid dancer we have considered the Competence as the effectiveness at correctly fulfilling a set of creative acts, and the Certainty as the confidence in its creative capabilities. In particular, when the agent generates something new it will increase its Certainty if the result has a positive feedback from the audience. Therefore, we considered the Certainty as a function of the amount of the **repertoire patterns** used for the creation of the dance judged as being *pleasant* by an audience. Competence and Certainty are influenced by the production-evaluation cycle, as shown in Figure 2. The production phase is aimed at creating a dance, obtained by the composition of a sequence of movements that should be harmonically executed with respect to the auditory perceptions of the robot. The robotic dancer has been trained with a set of basic associations between music and movements. Moreover, it is able to create and evaluate its own performance storing significative results into the long-term memory (LTM). These results can be used as input by the planning and execution modules.

In the following section, we will explain how the dance is generated, executed, and how artificial creativity drives all the aspects of the artistic composition process.

3. Dance creation

Our idea is that humanoid dance is a composition of a harmonized movements sequence that satisfies some aesthetic criterions. Under this vision, we make use of a Hidden Markov Model to create an humanoid improvised dance. The best motion sequence associated with the perceived music is estimated through an Hidden Markov Model, and this sequence is used to generate the dance. Other HHM based approaches have been proposed for generating motion such as in Kwon & Park (2008) (combined with principal components), and Tanco & Hilton (2000) with movements clustering by K-means. Our HHM approach is quite different, inverting its logic when we handle observations and states of the model, as explained in the following.

3.1. Music perception module

The Music Perception module processes both rhythmic sounds and music. Usually, audio processing requires complex algorithms and many computational resources that not allow the robot to have a fast reaction to its auditive stimuli. For this reason, we have chosen to detect only musical beats and intensities of sound using fast algorithms. We employ Essentia library, Bogdanov et al. (2013), to obtain the relevant information from perceived audio signal. In detail, the features extracted are: beats locations, beats interval, BPM (Beats Per Minute) and confidence of detected beats. The extracted features are essential for the dance creation because they represent the music perception of the robot, and allow it to synchronize its movements with musical beats.

Beats represent the basic units of time defining a rhythmic pattern for a piece of music, and BPM (Beats Per Minute) represents a measure of its *tempo*. The total number of beats in a music depends on the temporal interval between two successive beats. Naturally, smaller intervals are associated with a piece of music perceived as fast. The temporal localisation of the beat is important for two purposes: to compute the intensity of music fragments and to allow the robot to stay in sync. Generally, the increasing or decreasing of the rhythm is due to the change of speed and the intensity of the music. Humans associate different emotions while listening to a music, defining a sort of musical mood that is mainly influenced by the music intensity, Shiratori et al. (2006). The ability to perceive a musical rhythm and intensity is a natural born skill for a human being. The robot emulates this ability by detecting the beat location and calculating the quantity of intensity between two consecutive music fragments. The obtained value is named *loudness* and it is used to associate a class at each music fragment among nine possible values. In a previous work, Manfrè et al. (2015), we used only three classes (labeled as Low, Middle, High), but empirical experiments have indicated us to use a larger number of levels in order to catch peculiarities of a huge number of musical pieces (we are planning to use a database of thousands of them). The sequence of class labels constitutes our *Model of Music*. Such a model encodes how the robot perceives the music and it is used to find the best matching between movements and music. Naturally, the robot perception has to influence the dance creation, and in fact, the selected movements are chosen by the interpretation of what it hears.

The *Model of Music* represents the observed sequence, and together the Emission Matrix EM and the Transition Matrix TM will be the inputs of our Hidden Markov Model to estimate the hidden states that correspond to basic movements. The composition of these basic movements creates the dance schema performed by the robot.

To execute dance, the robot uses the temporal interval between two consecutive beats to define the duration of each single movement, keeping the tempo and remaining in sync with the music. One single movement is performed by the robot at each beat interval, and the total time of the dance coincides with the duration of the audio input.

At present, the robot has to listen to the music and process the whole audio file before executing a dance. The computed music model is associated with the corresponding piece of music of the database. Using a standard software library for music fingerprint recognition [3], few seconds are sufficient to the robot for retrieving the corresponding music model and starting to dance at the right point of the listened musical piece.

3.2. Movements Selection

The Hidden Markov Model is governed through a doubly stochastic process: the observation is a probabilistic function of the state, allowing to detect the hidden states N_s that produce the set of emitted symbols M_s. The transition among states is characterized by a stochastic matrix, the Transition Matrix TM, that measures the probability that the status of the system will evolve from the state i to the state j. Since from the i-th state, the system can evolve in one

---

[3] https://github.com/worldveil/dejavu

the N_s states, the sum of elements for each row must be equal to 1.

In a similar way, there are emission probabilities from one of the N_s states, defining the probability of observing a symbol at given time. This property is represented by the Emission Matrix EM that provides the probability that an output symbol, chosen among the possible M_s is emitted while the system is in a given state. An EM row has a sum of its elements equal to 1.

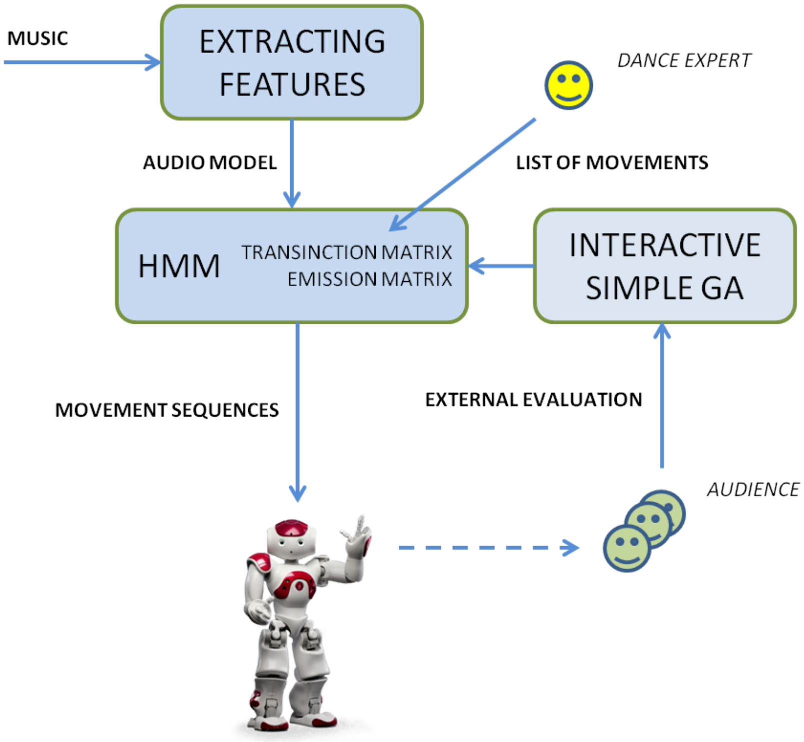


Figure 2: The schema of the dance execution and learning

In our system, we decided to associate the hidden states to the corresponding basic movements, while the observed symbols are associated with the music descriptions. According to this model, we can characterize a sequence of movements as a path along the states of the model so that the transitions among states represent the movements sequence.

The TM has a key role in the selection and the composition of the movements. The chosen movements have been acquired from movements of human dancers, Koenemann & Bennewitz (2012), as a set of elementary movements. The movements have been suitably selected so that the set does not contain movements that could cause the fall of the robot or can damage any of its parts. This hypothesis does not assure to avoid fallings. During our experiments, a fast musical tempo combined with the execution of some movements caused a fall. When the robot experiences such case, the TM is modified by lowering the probability (or zeroing it) of the movement combination involved in the fall.

In our system, the set of fixed elementary movements and TM are decided by a choreographer. In future, we will investigate an autonomous mechanism to extend the set of movements in order to allow the robot to have more space to express its creativity. The design of the initial TM is very important because the goal is to compose a sequence of movements that could be perceived as natural by a human audience. An aesthetic measure could be used to drive the TM modifications: maximizing transition probabilities of harmonic movements, and lowering probabilities of disliked movements.

The TM is independent of the music type and it only refers to the movement transition. The music, its rhythm, and melody determines a unique choreography constrained by movement transition tendencies of the robot. Under this point of view, TM is a fundamental part of the artistic behavior of the robot.

the EM constitutes the link between the music and the movements. In our formulation, the emissions of the symbols are referred to the music observations while the hidden states are referred to the movements. This representation is somehow counter-intuitive and inverts the relationship between cause and effect that one could conceive. Notwithstanding, we found this inversion profitable for the generation of the movements. The representation of the audio input according to the given model of music could be used as a sequence of emission of the Markov model. It is a sequence of observations and it is used to estimate the best sequence of the corresponding states. Since they are not observable, a probabilistic inference is done by employing Viterbi algorithm. There are several possible states that could generate the sequence observed, so we choose to use the Viterbi algorithm, Rabiner (1989), to find the optimal sequence of states associated with the given observation sequence. The output of Viterbi algorithm corresponds to a series of states that are bound to a set of movements. Their composition creates the dance that the robot will perform when it listens to the input music. As stated above, with Markov Models it is possible to detect a match between a state and an observable event. Given a sequence of events, the model can be trained so that the sequence is bound to a state of the system at the different time. The desired result is to find out the status of the system, knowing only the observation of symbols. The Model of Music obtained from the previous step, represents the coding of how the robot perceives music dividing the song into fragments and associating to each part to nine different loudness categories. The output of the Viterbi algorithm corresponds with a series of movements, chosen by the robot, to interpret what it had heard. This series of movements will be used in the last section of the system, in which they will be performed in sync with the rhythm.

3.3. Learning of Dancing Styles

In this section is described in detail the process that leads to learning dancing styles under the guide of a human teacher. The robot can collect and store various emission matrices corresponding to different dance behaviours, and create a choreography in a relationship at the perceived rhythms. After learning phase, live performances are judged by real audience determining the success or the failure of dance styles. the evaluation influences the robot behaviour in future executions. Thanks to cognitive architecture, it is possible to create interesting performances mixing interactions with humans, environment perceptions, and creative composition of dances [4].

To teach the robot to dance, we have used an approach inspired to the basic Simple Genetic Algorithm (SGA) Vose (1999) and the Interactive Evolutionary Computation (IEC), Takagi (2001). The population of the genetic algorithm

---

[4] Videos of live events are available at http://www.pa.icar.cnr.it/scarlab/robodance/

is made up of emission matrices EMs, which constitute the associations of M movements (rows) and K music classes (columns). Each emission matrix is a real-valued chromosome and it is composed of genes, which are the matrix rows. Each gene is constrained by the sum of all elements that must be equal to 1 since each element represents the probability of movement execution in according to a perceived music class.

The population is composed of N individuals (chromosomes) , i.e. vectors *em* of size 1 by M*K obtained by resizing EMs. At each evolution step, the 10% of chromosomes is selected to create the next generation. The crossover operation preserves the EM constraints, cutting the chromosome only at positions corresponding to the end of the rows. One of the main advantages is that this approach allows to work directly with the phenotype and no additional coding is required. Furthermore, by choosing the crossover point at the end of a gene we preserve the constraint that all the elements of a gene sum up to 1. This avoids a subsequent normalization, which could distort the solution. After crossover, the parents used to create the new population are deleted. The first population is randomly generated and ordered with respect an internal fitness function ρ based on a suitable distance function aiming to *reproduce* a reference sequence of movements $\mathbf{m}^*$, or to *follow* the music model $\mathbf{s}$.

Given a vector em_{ij}, where i=1,. . . ,N, and j indicates the evolution step, the corresponding matrix EM_{ij} determines a sequence of movements $\mathbf{m}_{ij}$ for $\mathbf{s}$. Then, we compute the internal fitness function ρ as dist($\mathbf{m}_{ij}$, $\mathbf{m}^*$). In our experiments, we tried also to avoid the use of an explicit example, computing dist($\mathbf{m}_{ij}$, $\mathbf{s}$). In the first case, we tested two variants of cosine distance and Jaccard distance. In the second case we tested Spearman distance. At given evolution step, the population is ordered by means of internal fitness function, and best 100- l individuals are selected as parents for the next generation. The l l cardinality of the set L is equal to 0 when GA starts, and increases for each evolution step thank to the external evaluation. In fact, randomly five em_{ij} are selected and their derived dances are proposed to the teacher. If the judgement is positive the selected individuals are saved in a list of “privileged population”

L that will survive to the next generations. The dances are simulated by a 3D viewer, allowing the teacher to fastly evaluate performances without using the real robot, and to easily change the point of observation.

The choice of selecting and judging few random individuals from 1000 individuals of the population without any influence of the internal evaluation is justified by the fact that a sequence of movements can be judged creative also it does not fit with the internal evaluation, leading to the fact that the robot has truly created a new style that is pleasant to the evaluator.

When $l = 100$, the robot has a complete repertoire stored in LM, and it is ready to execute performances to submit for general audience evaluations. Trough its cognitive architecture, the external evaluations strongly influence the robot behaviour, and if necessary, robot could try to make improvements continuing evolution, eventually changing the internal fitness functions or introducing mutation operators.

4. Experiments

We tested the proposed approach embedding the cognitive architecture on the humanoid robot NAO, Corporate (2015). The set of M=20 basic movements has been created in cooperation with professional dancers, selecting the more relevant ones with respect simplicity, aesthetic, duration, and robot stability. The music used in the learning phase is a composition of few seconds ($11 \div 16s$) extracted from six musical pieces belonging to different musical genres. In particular, the processed music produces a music model $\mathbf{s}$ of size W=73, i.e. a sequence of 73 values of music classes in the range from 1 to 9 (=K) . Also, $\mathbf{r}$ has 73 movements given by a choreographer as a relevant example for the robot. In this work we use $K = 9$ levels to classify music, and a set of $M = 20$ basic movements created in collaboration with professional dancers. The population is composed of N=1000 individuals (chromosomes) , i.e. vectors em of size 1 by M*K obtained by resizing EMs. At each evolution step, 100 chromosomes are selected to create next generations.

We have done various trials pursuing different goals, exploring four different fitness functions:

1. Cosine similarity between **m** and **r**;
2. Jaccard similarity between **m** and **r**;
3. Cosine similarity between histograms of occurences of **m** and **r**;
4. Spearman similarity between **m** and music model **s**.

where **m** is the sequence of movements given music model **s**, EM and TM matrices. The reference sequence **r** is the sequence of movements indicated by a professional dancer as valuable example or model to follow. For the diagrams in the following, we consider also distance as 1-similarity. To calculate similarity or distance, each movement is associated to an integer in the range from 1 to M=20, and similar or equivalent movements in the opinion of the choreographer have associated consecutive integers.

$$Cosine\ similarity = \frac{\sum\limits_{i=1}^{W} r_i m_i}{\sqrt{\sum\limits_{i=1}^{W} r_i^2} \cdot \sqrt{\sum\limits_{i=1}^{W} m_i^2}} \tag{1}$$

$$Jaccard\ similarity\ coefficient = \frac{Different\ Movements}{W} \tag{2}$$

$$Cosine\ similarity\ on\ occurrences = \frac{\sum\limits_{i=1}^{W} p_i q_i}{\sqrt{\sum\limits_{i=1}^{W} p_i^2} \cdot \sqrt{\sum\limits_{i=1}^{W} q_i^2}} \tag{3}$$

where p is the occurrence vector of r, and q is the occurrence vector of m (each element has a value between 0 and W=73, the sum of all elements is equal to W).

Spearman distance

$$Spearman\ distance = \rho = 1 - \frac{6 - \sum d_i^2}{W(W^2-1)} \tag{4}$$

Where $d_i = s_i - m_i$, and i=1,...,W.

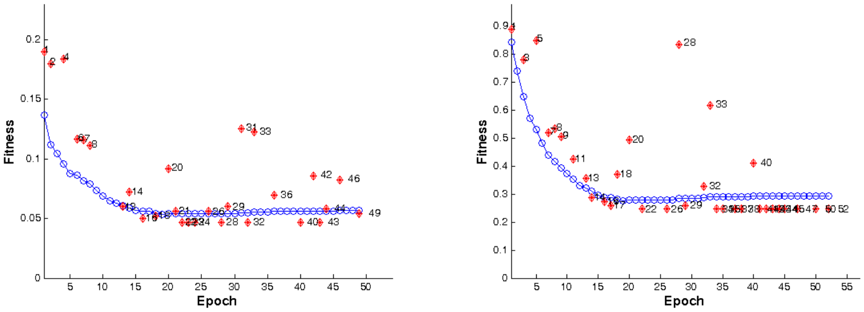
By means of these similarity functions, we could obtain different creative behaviors from learning phase. Figure 3 shows how fitness function (reported as distance = 1.0-*similarity*) converges to its minimum value after 50 epochs in the four considered situations. For each epoch, a circle indicates the best fitness value of the 1000 individuals. Diamonds indicate the fitness of selected chromosomes after the teacher has expressed a positive evaluation on derived dances. The number beside the diamond indicates the epoch in wich the selection has been done. After 15 epochs, all distances are minimized. Naturally, individuals selected in first epochs generally have low fitness, and after 50 epochs all individuals have the maximum fitness value. But it is interesting that some selected individuals are very different from reference one, causing an injection of variability on evolution relevant for computational creativity. In this sense, Jaccard approach seems more subject to consider the significative different sequence of movements, breaking the monotony of a simple imitation. At the opposite, the Spearman approach is uniform and evolves following the teacher evaluation, even if it does not consider an explicit example.

In the first case of cosine similarity, the aim of the learning phase is to follow the sequence of reference given by choreographer taking into account the order of movements, and considering acceptable to substitute similar movements. Jaccard similarity causes an imitative behavior, minimizing the differences between movements sequence with respect the reference one. The cosine of movements occurrences avoids to consider the specific order of movements of the example given by choreographer but takes into account the frequency of utilization of the movements. Spearman similarity allows the GA to evolve without an example to imitate, searching an intrinsic correlation between movements and music models. The approach simulates a human dancer that learns to dance independently, by self-taught, solely with music stimuli. For this scope, we have used the *Spearman distance* that looks for a relationship between two different sets. This distance is used to train a robot knowledge base of dance in an autonomous manner, that only through listening music and opinion of the audience, the robot enhances his knowledge base if the judge is negative discard the last information and continues to train until it reaches a good level . During evolution, the system use a Spearman distance as fitness function between the Music Model of input music and the sequence of movement generated at the given epoch, in order to reproduce motion sequences that are in keeping with the model of the music and find autonomously a good correlation between the model of the music and the movements, suiting accordingly the values of the emission matrix.

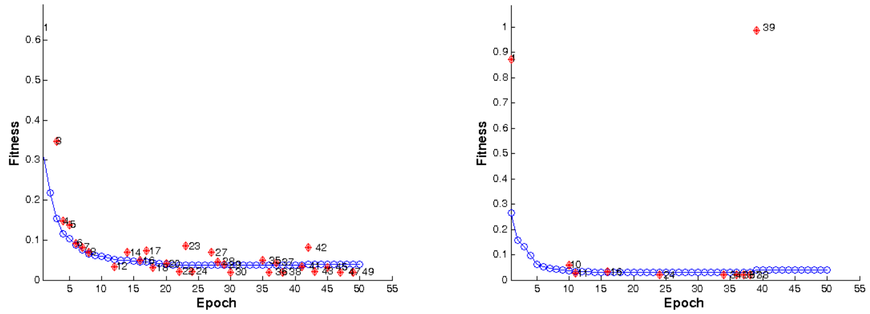
The second series of graphs in figure 4 show the mean frequency of movements at epoch 1, 10, 20, 30, 40 (blue lines), and 50 (red line). The green line shows the occurrences of the reference sequence. As stated before, Jaccard approach refers to an imitation behavior, while the others are quite different from the reference sequence. Maybe, in this sense, Spearman approach could be considered more creative, being independent of a given example, and creating an own dance style even if derived from teacher evaluations.

5. Conclusions

The paper has discussed the effects of different similarity functions used as internal fitness functions of an interactive genetic algorithm. GA is part of learning process aiming to obtain a computational creative behavior in a cognitive architecture deployed in a humanoid dancer. In the learning phase, a choreographer has the possibility to follow different objectives by changing the fitness function. In the experiments, we use a fixed set of movements and a given medley of music to evaluate only the influence of GA in artistic performance. But the system is flexible to any music genre, in fact, it is possible to change the number of intervals to be considered for the duration of each single movement. In this way it is also possible to specialize the robot to dance a specific genre of music, adding specific movements in order to be able to better interpret the song. In this way it is also possible to improve the robot to dance a specific genre of music, increasing the set with characteristic movements in order to

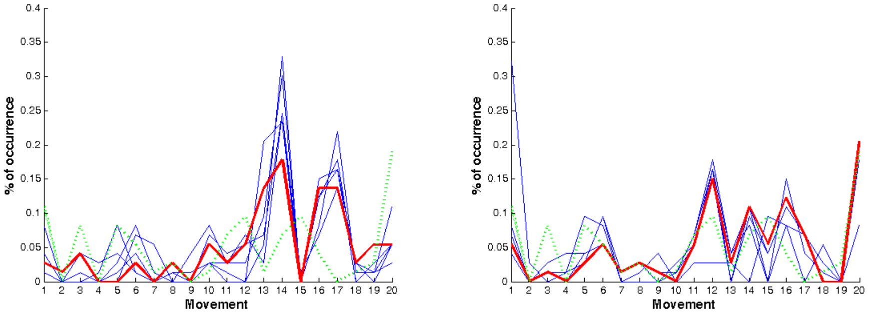


(a) Cosine (b) Jaccard



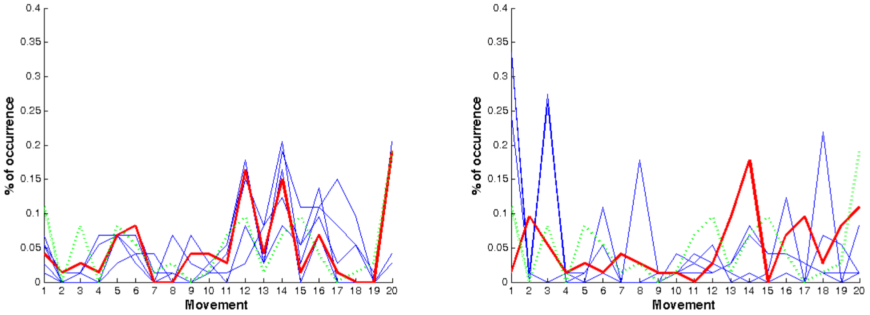
(c) Cosine of occurences (d) Spearman

Figure 3: Fitness function (reported as distance = 1.0-*similarity*) vs GA epochs using different similarity functions. For each epoch, a circle indicates the best fitness value of the 1000 individuals. Diamonds indicate fitness of selected chromosomes after the teacher has expressed a positive evaluation on derived dances. The number beside the diamond indicates the epoch in wich the selection has been done.



(a) Cosine

(b) Jaccard



(c) Cosine of occurences

(d) Spearman

Figure 4: Blue lines are the mean frequencies of movements at epoch 1, 10, 20, 30, 40. Final epoch 50 is drawn with red line. The green line shows the occurences of the reference sequence

interpret better the song. In future, we planning to define a *creative* strategy to modify transition matrix, including new movements, or modifying its probabilities according to some given constraints such as symmetricity, fluidity, esthetic impact, emotional impact, and so on.

References

Augello, A., Infantino, I., Lieto, A., Pilato, G., Rizzo, R., & Vella, F. (2015a). Artwork creation by a cognitive architecture integrating computational creativity and dual process approaches. *Biologically Inspired Cognitive Architectures*, .

Augello, A., Infantino, I., Pilato, G., Rizzo, R., & Vella, F. (2013). Introducing a creative process on a cognitive architecture. *Biologically Inspired Cognitive Architectures*, *6*, 131–139.

Augello, A., Infantino, I., Pilato, G., Rizzo, R., & Vella, F. (2014). Combining representational domains for computational creativity. In *Proceedings of 5th international conference on computational creativity, ICCC 2014, June*.

Augello, A., Infantino, I., Pilato, G., Rizzo, R., & Vella, F. (2015b). Creativity evaluation in a cognitive architecture. *Biologically Inspired Cognitive Architectures*, *11*, 29–37.

Bach, J. (2009). *Principles of synthetic intelligence PSI: an architecture of motivated cognition* volume 4. Oxford University Press.

Bartl, C., & Dörner, D. (1998). Psi: A theory of the integration of cognition, emotion and motivation. In *Proceedings of the 2nd European Conference on Cognitive Modelling* (pp. 66–73). DTIC Document.

Biles, J. (1994). Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference* (pp. 131–131). INTERNATIONAL COMPUTER MUSIC ACCOCIATION.

Boden, M. (2009). Computer Models of Creativity. *AI Magazine*, (pp. 23–34). URL: `http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/2254`.

Boden, M. A. (1998). Creativity and artificial intelligence. *Artificial Intelligence*, *103*, 347–356.

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., & Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. In *ISMIR* (pp. 493–498).

Cai, Z., Goertzel, B., & Geisweiller, N. (2011). Openpsi: Realizing dörners psi cognitive model in the opencog integrative agi architecture. In *Artificial General Intelligence* (pp. 212–221). Springer.

Colton, S., Pease, A., & Charnley, J. (2011). Computational creativity theory: The face and idea descriptive models. In *Proceedings of the Second International Conference on Computational Creativity*.

Colton, S., Wiggins, G. A. et al. (2012). Computational creativity: The final frontier? In *ECAI* (pp. 21–26). volume 12.

Corporate, A. (2015). Who is nao? In *https://www.aldebaran.com/en/humanoid-robot/nao-robot*.

Dostál, M. (2012). Genetic algorithms as a model of musical creativity–on generating of a human-like rhythmic accompaniment. *Computing and Informatics*, *24*, 321–340.

Galanter, P. (2012). Computational aesthetic evaluation: Past and future. In *Computers and Creativity* (pp. 255–293). Springer.

Goertzel, B., Lian, R., Arel, I., de Garis, H., & Chen, S. (2010). A world survey of artificial brain projects, part ii: Biologically inspired cognitive architectures. *Neurocomputing*, *74*, 30–49.

Horowitz, D. (1994). Generating rhythms with genetic algorithms. In *AAAI* (p. 1459). volume 94.

Jordanous, A. (2012). A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, *4*, 246–279. URL: `http://dx.doi.org/10.1007/s12559-012-9156-1`. doi:`10.1007/s12559-012-9156-1`.

Koenemann, J., & Bennewitz, M. (2012). Whole-body imitation of human motions with a nao humanoid. In *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on* (pp. 425–425). IEEE.

Kwon, J., & Park, F. C. (2008). Natural movement generation using hidden markov models and principal components. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, *38*, 1184–1194.

Manfrè, A., Infantino, I., Vella, F., & Gaglio, S. (2015). An automatic system for humanoid dance creation. *Biologically Inspired Cognitive Architectures*, .

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*, 257–286.

Romero, J., Machado, P., Carballal, A., & Correia, J. (2012). Computing aesthetics with image judgement systems. In J. McCormack, & M. dInverno (Eds.), *Computers and Creativity* (pp. 295–322). Springer Berlin Heidelberg. URL: `http://dx.doi.org/10.1007/978-3-642-31727-9_11`. doi:`10.1007/978-3-642-31727-9_11`.

Seo, J.-H., Yang, J.-Y., Kim, J., & Kwon, D.-S. (2013). Autonomous humanoid robot dance generation system based on real-time music input. In *RO-MAN, 2013 IEEE* (pp. 204–209). IEEE.

Shinozaki, K., Iwatani, A., & Nakatsu, R. (2007). Concept and construction of a robot dance system. In *International Workshop and Conference on Photonics and Nanotechnology 2007* (pp. 67942J–67942J). International Society for Optics and Photonics.

Shiratori, T., Nakazawa, A., & Ikeuchi, K. (2006). Dancing-to-music character animation. In *Computer Graphics Forum* (pp. 449–458). Wiley Online Library volume 25:3.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, *89*, 1275–1296.

Tanco, L. M., & Hilton, A. (2000). Realistic synthesis of novel human movements from a database of motion capture examples. In *Human Motion, 2000. Proceedings. Workshop on* (pp. 137–142). IEEE.

Todd, M., Lee, C., & O'Boyle, D. (2002). A sensorimotor theory of temporal tracking and beat induction. *Psychological Research*, *66*, 26–39. URL: `http://dx.doi.org/10.1007/s004260100071`. doi:`10.1007/s004260100071`.

Toiviainen, P., Luck, G., & Thompson, M. R. (2010). Embodied meter: hierarchical eigenmodes in music-induced movement. *Music Perception: An Interdisciplinary Journal*, *28*, 59–70.

Tubb, R., & Dixon, S. (2014). A four strategy model of creative parameter space interaction. In *Proc. of fifth international conference on computational creativity, Ljubljana, Slovenia, 9th–13th June 2014*.

Vose, M. D. (1999). *The simple genetic algorithm: foundations and theory* volume 12. MIT press.

Xia, G., Tay, J., Dannenberg, R., & Veloso, M. (2012). Autonomous robot dancing driven by beats and emotions of music. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1* (pp. 205–212). International Foundation for Autonomous Agents and Multiagent Systems.