# A Computational Framework for Aesthetical Navigation in Musical Search Space

**Sahar Arshi[1], Darryl N. Davis[1]**

**Abstract.** This article addresses aspects of an ongoing project in the generation of artificial Persian (-like) music. Liquid Persian Music software (LPM) is a cellular automata based audio generator. In this paper LPM is discussed from the view point of future potentials of algorithmic composition and creativity. Liquid Persian Music is a creative tool, enabling exploration of emergent audio through new dimensions of music composition. Various configurations of the system produce different voices which resemble musical motives in many respects. Aesthetical measurements are determined by Zipf's law in an evolutionary environment. Arranging these voices together for producing a musical corpus can be considered as a search problem in the LPM outputs space of musical possibilities. On this account, the issues toward defining the search space for LPM is studied throughout this paper.

## 1 INTRODUCTION

The *Radif* is the repertory of Persian traditional music which consists of different *Dastgāhs* [1]. Dastgāhs are distinguished from each other by their musical modal systems and the movement of melodies[2], [3], [4]. Dastgāhs have been unevenly mapped to modes in Western musical terminology [1]. The Dastgāh concept determines both the title for a group of individual pieces with their characteristic modal identity and the primary mode in each group [1]. There are twelve principle groups of modes in Persian music, namely, Shur, Abou'atā, Bayāt-e-Tork, Afshāri, Dashti, Homāyoun, Bayāt-e-Esfehān, Segāh, Chāhārgāh, Māhour, Rāstpanjgāh, and Navā [1]. Each Dastgāh consists of individual melodies called Gushé, which vary in length and importance[1]. Performing in a Dastgāh begins with Darāmad which are the most representative pieces of a Dastgāh. Darāmads have the prominent mode and melodic patterns of the Dastgāh itself giving the Dastgāh its identity [1], [5]. The modulation occurs with the move from one Gushé to another or a change in the central tone, or Shāhed note [6].

The current musical warehouse is the result of centuries of evolution of Persian music conjoint with historical and cultural transmutations. However, there are still varieties of other melodies waiting to emerge. Once modulated with Western music it can be considered as a potential bed for cross cultural interactions. Although Persian music has vast musical systems in comparison to its Western contemporary music counterpart, one of the problems encountered is the entrapment in the structures. This makes the composition more reliant on the emergence of great masters whom with their novel creativity and familiarity of the complexities of Persian music were able to put a step forward in this field and add pieces to different Dastgāh. Therefore the variety of melodies and Gushé in a Dastgāh is limited to what was produced in the past.

The use of algorithmic composition has been under investigation for many years with different motivations: Mechanization of music production; exploration of the behaviour of the algorithms; mathematical models in generating the patterns; studying the cognitive behaviour of creation in human being [7]; and modelling biological patterns in nature in respect to music.

Mechanisation of music generation has been done for producing melody, rhythm, harmonization, and counterpoint or imitating a specific genre of music or composition style [8]. The level of automation varies from generating motifs for inspiration to more complex corpus composition. Computer aided algorithmic composition is the term applied for assisting musicians in the composition process and providing them with new materials; (some available frameworks or languages for making musical software include Csound [9], MAX/MSP [10], while some musical software include EMI [11], [12], GenJam [13], and LBM [14], [15]). Deeper levels of composition automation target minimal or no interactions with human (Melomics corpus generation [16])

Methodologies in algorithmic composition can be categorized, based on the survey from [8], in four groups: , knowledge based systems; machine learning; evolutionary algorithms; and computational intelligence (e.g. cellular automata). All of the aforementioned categories except the last one apply human knowledge in their application. They have been widely used both for style imitation and creating novel music. However, cellular automata are able to generate novel material without utilising existing human domain knowledge. This potential of creativity makes them well-suited for exploring new dimensions of music composition.

There have been good progress with the research into genre imitation; successful applications include Strasheela [17]. Most research efforts are now focused on algorithmic creativity applications. The future directions for algorithmic composition includes hybrid methods [8] that use cellular automata (CA) as their music generator.

Liquid Persian Music is a CA based toolkit for exploring various musical possibilities. Pattern matching rules classify output from the cellular automata and update the parameters of a synthesiser to yield audio output. Controlling synthesizer parameters by means of the emergent nature of CA is an important characteristic of LPM. In this work each parameterisation of LPM, across both CA and pattern matching rules and the synthesiser, is considered to be an audio voice. Sequencing LPM produced

[1] Dept. of Computer Science, Univ. of Hull, HU6 7RX, UK.
Email: { S.Arshi@2014., D.N.Davis@ }hull.ac.uk

voices in a musical manner requires investigating a huge search space. The dimensions of this space are defined by the number of CA rules, pattern matching rules, the elements of sound synthesizer and melodic structure. One important question that this research addresses is how to evaluate the musical productions of such system in terms of aesthetics? Furthermore, is there a measurement for the creativity of the system itself?

Creativity is a diverse concept with multiple definitions. We need to be specific with its definition in order to evaluate it in the current project. In the next section, more clarifications addressing the concept of creativity are presented.

Previous researches show the application of musical Turing tests [18] and surveys [19] for evaluating musical productions; however, giving equal measurements for creativity for human and machine is challenging [8]. These evaluations are done according to pre-existing musical knowledge and cultural backgrounds, while the real creativity goes beyond pre-existing musical styles [20]. Despite the widespread research in the area of algorithmic composition itself, less attention has been given to assessing the outputs from creativity viewpoints [20]. Evaluating the computational creativity can be traced in [20] , [21], [22]. Nevertheless, the creativity of an artefact can be perceived by their aesthetical values [20]. Various scientific studies have been conducted on the matter of universals for recognizing natural or human-like phenomena, as well as frequency distributions, and power law. Among these is the use of Zipf's law as a basis for aesthetical measurement [7]. Zipf's law has had successful applications in measuring the aesthetical aspects of music [23] and we have been looking at it as the start point for advancing the current research from an aesthetical point of view.

In a previous experiment [24] the LPM output voices were analyzed in the search for finding proper tools for enhancing them in a musical way. A pool of voices have been produced and the pleasantness of each of those elements have been evaluated against aesthetic measurements using Zipf's law [23]. In a later experiment in the same paper, random sequences of voices were produced with nearly acceptable Zipf's slopes. The next level of investigations consists of designing a computational framework for sequencing LPM outputs in an evolutionary environment. The idea is that Genetic Algorithms are suitable candidates for delving in the problem of sequencing LPM musical elements. However, the huge search space makes it an impractical one, unless suitable constraints are taken into account.

In the second section, different types of creativity from viewpoint of computation are described. The third, fourth, and fifth sections revolve around background studies relevant to the current research. These include cellular automata and optimization methods and their applications in algorithmic composition and Zipf's law. In the sixth section the features of LPM software in employing cellular automata is briefly overviewed and the basis for measuring the creativity of the sequencer in the evolutionary domain is presented. The seventh part is devoted to the design of computational framework that allow the sequencing of LPM productions to be viewed as a search problem. The issues raised and potential solutions are discussed in detail. The paper is concluded with future direction of the research.

## 2 ON COMPUTATIONAL CREATIVITY

"What is creativity?" –This can be considered as an open-ended philosophical question. There are no boundaries for creativity, yet binding creativity in a framework for a definition is a necessary but difficult task. However, an artefact has some representative features which describe its qualities to some extent. These qualitative descriptions clarify the attributes an artefact should have to be considered as a piece of art work. Amongst all descriptions what is clear is that art and novelty have been two inseparable concepts. Sometimes a black dot on a white canvas is defined as a masterpiece and is exhibited in art galleries. The work of John Cage in his composition "four minutes and 33 seconds of silence" unbounds framed viewpoints towards art and creativity with avant-garde music. In a silent musical performance he lets the energy from audience noise vibrate the strings of a grand piano. The interaction of audience noises and musical instruments is popular as aesthetics of art performance. There are other criteria for defining creativity other than novelty, for example quality [21]. This discusses how the creation is to be considered to be a high-quality instance of its genre. Jon McCormack defines this attribute of creativity as being exhibitable [7].

Two different viewpoints exist about man-machine creativity. The machines that create art-like productions, and the machines which are autonomous in creating art [7]. The aim of creating could be to satisfy an audience or could involve the exploration of general meaning of creativity, without contributing to human comprehension or appreciation.

Boden [22] defines three types of creativity: combinational; exploratory; and transformational. She states all can be modelled by artificial intelligence. Combinational creativity consists of populating pre-existing materials and linking them in an artistic manner for generating new ideas. Exploratory creativity includes navigating in a *conceptual space* with implicit constraining rules. This exploration can result in discovering new transformed styles which would not have existed before an alteration happening on one or more of their defining dimensions (transformational creativity).

## 3 THE NATURE OF CELLULAR AUTOMATA & ITS APPLICATION IN ALGORITHMIC COMPOSITION

The advent of cellular automata originally dates back to 1940s, when Von Neumann was looking forward to develop a system capable of reproduction, comparable in certain respects with biological breeding [25], [26]. Cellular automata was studied as a dynamical system in 1960s [27]. Cellular Automata are discrete dynamical systems whose global intricate behaviour is determined by the reciprocal influence of identical elementary individuals.

Cellular automata exhibiting myriad genres of behaviour have been targeted as a creative tool for artists. By increasing the number of states and neighbourhood size, the state space expands exponentially, in a way that the normal life expectancy of a human is not adequate for navigating through all these patterns. Amongst all the various applications such emergent machine can have are, namely, the extraction of overall conformation for composition, MIDI sequencing, and sound synthesis [28].

Two of the early models of musical cellular automata include Beyls cellular automata explorer, and CAM developed by Millen. Having the aim of achieving complex musical patterns in the output [8], Beyls investigated broad criteria of configurations for

CA rules, and cell neighbourhood [29]. Some of these include the application of time dependent rules, and involving the neighbour states from previous and future generations in the computations. Dale Millen employed two and three dimensional game of life cellular automata and mapped the results to pitch and duration. He later explored the formation of musical organization from CAM [29].

Other popular cellular automata musical systems are CAMUS and Chaosynth [30], [31]. CAMUS exploits Game of Life and Demon Cyclic Space, and uses a Cartesian space mapping to MIDI for achieving musical triplets. The main idea in CAMUS is to model the dissemination of musical patterns in time by simulating the same effect in cellular automata [31]. Chaosynth is based on the model of chemical reactions of a catalyst. It is a cellular automata sound generator based on the production of sound granules which are the results of underlying additive synthesis processes. However the produced tones do not often resemble the acoustic sounds found in the real world; they are sometimes reminiscent of the natural sounds flow as well as the sound of waterfalls, or insects [30]. The interested reader is referenced to [8], [29] for a thorough review on previous research on the application of CA in generating electronic music.

Cellular Automata are usually used as a hybrid tool beside other artificial intelligence tools in music composition algorithms, since, in isolation, they do not presently produce melodic sounds. However, they can be a source of raw material and structures for inspiration for musicians [8]. In the end the generated sounds may need heavy editing by the composer and so be conformed to musical playing as stated by Xenakis; one of the pioneers who used CA for achieving the general structure of his compositions [32], [19]. Similar issue have been stated by Miranda, the creator of CAMUS, who considers the results as not being very musical [33].

## 4 GENETIC ALGORITHMS & THEIR APPLICATION IN ALGORITHMIC COMPOSITION

Genetic Algorithms (GA) are a class of Evolutionary Algorithms inspired by natural selection [34]. They are employed in areas of search and optimization. Previous applications of GAs imply their success in problem solving for domains with widespread solution spaces [35]. Therefore, they can be considered a well suited candidate in music composition, with its almost infinite possible combinations of musical elements. However, in order to guide the search and constrain the musical search space one can tailor fitness functions which fulfil musical aesthetical aspects or adhere to certain musical tastes [36].

A population of individuals are randomly initialized in a mating pool. Candidate solutions are coded as genotypes and are continually evolved in each nascent generation. The solutions contribute to crossover and mutation operations according to their fitness function. This assessment guarantees the survival of the most competent genes and raises the expectancy of convergence to optimal solutions. The reproduction operation consists of the selection of parents as the fittest individuals for breeding, which then undergo the crossover and mutation operations. In crossover, individual parents are selected and their genes are transmitted to each other by swapping, mostly in a meaningful manner. The mutation is a low-probability operation and involves changing a gene in the genotype [37]. It can help the search by avoiding being entrapped in local solution spaces. The algorithm stops when a pre-specified goal has been satisfied or some sort of limitations such as time or number of generations has been reached [36].

In previous applications evolutionary algorithms have been widely used for composing melodies, and harmonizing pre-specified melodies. The fitness function can be interactive or autonomous. In interactive fitness functions a human user evaluates the candidate individuals in the population. These fitness functions usually contribute to user fatigue and should be used in domains where other fitness functions are unable to gain the desired results. The other types of fitness assessment usually contribute to the application of machine learning methods. In the following some examples of both types of fitness functions are described.

Horner and Goldberg [38] are one of the first to present the application of genetic algorithms in algorithmic composition. Thematic bridging is a composition method; starting from an initial pattern, the system goes into a series of evolutionary process to transform to the final pattern. The GA individuals are the transformation operators and the fitness function is evaluated as the distance to the target pattern. The sequences of the generated patterns are the output of the system. Jacob [39] applied three phase modules in the design of his composition system; the *Ear*, *Composer* and the *Arranger*. The human user trains the *Ear* which acts as an evaluator in the process of creating musical motifs according to authorized intervallic combinations. The *Arranger* is determined by the user as well, to reorder and assemble the output in the form of musical phrases. In GenJam [40], Biles devised an evolutionary algorithm for generating Jazz melodies. Later, he used an artificial neural network (ANN) to automate the task of evaluation to overcome the interactive fitness function bottleneck. However, the ANN failed to extend the evaluations to cases other than what was in its training set [8].

Genetic algorithms have been applied independently or as hybrid models accompanying various self–governing artificial intelligence and computational methods as well as knowledge-based models, Markov chains, artificial neural networks, and complex systems in producing artificial music.

Fitness functions can be defined simply as a weighted sum of distances to a target melody, however, if the musical statistical are selected poorly, reaching satisfactory results are unlikely to happen [41], [42].

In a series of applications neural networks have been used as fitness functions. Neurogen applies two neural networks, one for assessing the intervals between pitches, the other one for evaluating the overall structure. One of the successful neural networks and genetic algorithms hybrid approaches in computer music is the work of Manaris and his colleagues. Manaris et al. trained neural networks as a fitness function with statistical metrics to identify individual compositions with Zipf's distribution property [23].

Markov Chains have been applied as fitness functions for evolving musical sequences in a number of applications [43]. In [44] variations between two musical pieces have been modelled using random jumps between two Markov chains trained with two different pieces. Hidden Markov models trained with proper counterpoint training set have been able to produce Palestrina style first species counterpoint [45]. HMM trained with chorale harmonization add extra voice elements to a pre-processed melody in [46].


Figure 1. LPM user interface.

In [19], n-gram models, Zipf's law, and information entropy are applied as trainable fitness functions in a series of experiments. Musical samples are used to train N-gram classifier which is later applied as the fitness function in a random mutation hill climber. These fitness functions evaluate sequences of pitches, and the genetic operators are employed as tools for search space navigation. Later in the same work evolutionary algorithms are applied to evolve cellular automata as a music generator.

## 5 ZIPF'S LAW IN MUSIC

Zipf's law characterizes the scaling attributes of many natural effects including physics, social sciences, and language processing. Events in a dataset are ranked (descending order) according to their prevalence or importance [23]. The rank and frequency of occurrence of the elements are mapped to a logarithmic scale, where linear regression is applied to the events graph. The slope and $R^2$ measurements demonstrate to what extent the elements conform to Zipf's law. A linear regression slope of -1 indicates Zipf's ideal. Zipf's law can be formulated as $F{\sim}r^{-a}$, in which $r$ is the statistical rank of the phenomena, $F$ is the frequency of occurrence of the event, and $a$ is close to one in an ideal Zipfian distribution. The frequency of occurrence of an event is inversely proportional to its rank. $P(f) = 1/f^n$ is another way to express the Zipf's law. $P(f)$ is the probability of occurrence of an event with rank f. In case of $n$=1(Zipf's ideal), the phenomenon is known as pink noise. The cases of $n$=0 and $n$=2 are called white and brown noises, respectively.

Voss and Clarke [47] have observed that the spectral density of audio is 1/f like and is inversely proportional to its frequency. They devised an algorithm which used white, pink, and brown noise sources for composing music. The results show that pink noise is more musically pleasing due to its self-similarity characteristics, the white noises are too random, and the brown noises are too correlated producing a monotonous sound.

In the musical domain, Zipf's metrics are obtained by enumerating the different musical events' frequency of occurrence and plotting them in a log-log scale versus their rankings. The slope of Zipf's distribution ranges from $-\infty$ to 0. The decreasing of the slope to minus infinity reflects an increase in the level of monotonicity. The r-squared value is between 0 and 1. Various publications explore the utilization of Zipf's law in musical data analysis and composition. Previous experiments show its successful application in capturing significant essence from musical contents. In [23] the Zipf's metrics consist of simple and fractal metrics. The simple metrics include seventeen features of the music as well as the ranked frequency distributions of pitch, and chromatic tone. Fractal metrics gives a measurement of the self-similarity of the distribution. These metrics were later used to train neural networks to classify musical styles and composers, with an average success rate of over ninety percent; demonstrating that Zipf's metrics extract useful information from music in addition to determining the aesthetical characteristics of music pieces.

## 6 LPM OVERVIEW

Liquid Persian Music (LPM) is an auditory software tool developed at the University of Hull [15][48]. LPM explores the idea of artificial life systems in producing voices which can be considered as new types of electronic music. The software takes advantage of the Synthesis Toolkit (STK) [49] for implementing the physical model of a stringed musical instrument. A model of its parameters are controlled by defined pattern matching rules. An OpenAL library is responsible for propagating the producing voices. Figure 1 illustrates the LPM user interface.

The elementary CA used in LPM consists of an assembly of cells arranged in a one dimensional array that produces a two dimensional matrix over time. Each cell is in one of k finite states at time t, and all the cells evolve simultaneously. The state of a cell at time *t* depends on its state and its neighbours' states at time *t-1*. In the one dimensional elementary CA (which is the subject of this study), the permutations of each cell with its two adjacent neighbours specifies eight situations. Once allocated to binary states, the selection of one of the 256 local transition rules specify the CA evolution [27]. Wolfram studies on CA recognize four classes of behavior, namely, fixed, cyclic, chaotic, and complex. Li and Packard [50] subdivided the second class to three further subgroups, namely heterogeneous, periodic with intervals greater than one, and locally chaotic.

In every time step of the CA, the pattern matcher extracts the difference between consecutive generations. Twenty different pattern matching rules have been defined in this software as well as metrics using Dice's coefficient, and Jaccard similarity. The obtained values from the pattern matchers are then fed as parameters into the STK synthesizer for producing sounds. Some of the synthesizer parameters include ADSR envelope, loop gain, and the musical instrument string length for defining frequency. Further information about the software can be found in [51].

An important point is that the aggregation of a CA rule and a pattern matching rule on each of the synthesizer elements does not produce a single note but a collection of notes; these are referred to as *voices* throughout this paper.

Studying the musical behaviour derived from one-dimensional (1D) CA does not require the investigation of the 256 rules' behaviours. The rule space can be reduced to 88 fundamental behaviours [52] by applying conjugate, reflection, and both transformations together [27], since they lead to rule sets with inherently equivalent behaviour. (The interested reader is referred to [27] for formulation of conjugate and reflection transformations and how they are applied to find equivalent CA rules). The 88 1D CA rule behaviours, 7 defined synthesizer parameters, together with 20 pattern matching rules, expand the number of voices to $88 * 20^7$. If the pattern matchers are chosen from separate cellular automata rules, then the voices number would become $88^7 * 20^7$. Considering the temporal and intervallic patterns and the CA number of iterations the search space would expand to $88^9 * 20^9 * t$.This defines the base auditory search space for the computational framework being developed.

In [24] the outputs of LPM have been explored through graphs and auditory tests. The behaviour of each of the pattern matching rules over one-dimensional cellular automata rule space have been explored and categorized in an initial step. The consequent experiments in [24] focus on the study of Zipf's law on LPM individual voices and sequence of voices.

In a first experiment, the output distributions have been investigated regarding their compliance with Zipf's law. Figure 2 presents examples on LPM output with their corresponding Zipf's distribution. In a second test, the results from the first experiment were categorized according to the expectations from studying their behaviours. This part of the experiment was conducted by a confusion matrix to measure the convenience of using Zipf's law for recognizing musical from unmusical voices. In a third experiment, collections of voices were sequenced; some with pleasing Zipfian slopes results that were expected to have musical voices. Figure 3 depicts two samples of Zipfian distribution for random sequences of voices. The random sequences of voices are selected from pattern matching rules applied to CA with iterations up to 10000 and 20. Figure 3 (a) shows a more monotonous output than figure 3 (b). The sequence of longer motives seem to have a more tedious structure. However some of the CA and pattern rules did not contribute to musical outputs by themselves. However, experiments with crafted pieces have shown that the proper combination of the voices can produce acceptable musical results. The measured Zipfian slopes characterize the global features [23] of the produced music. The attention was kept on one dimension of synthesizer (the frequency) and on global measurement of aesthetics throughout the study, for simplicity. In the next section we shall reveal some of the challenges in designing a computational framework that will allow candidate LPM voices to be sequenced into musical composition system. The experiments conducted in the previous paper [24] have been targeted as a base for designing the fitness function for the problem of sequencing LPM voices in an evolutionary space.

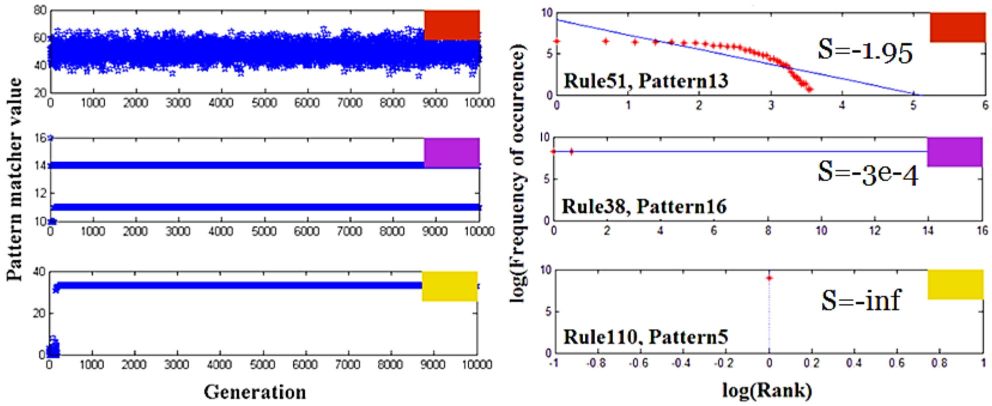
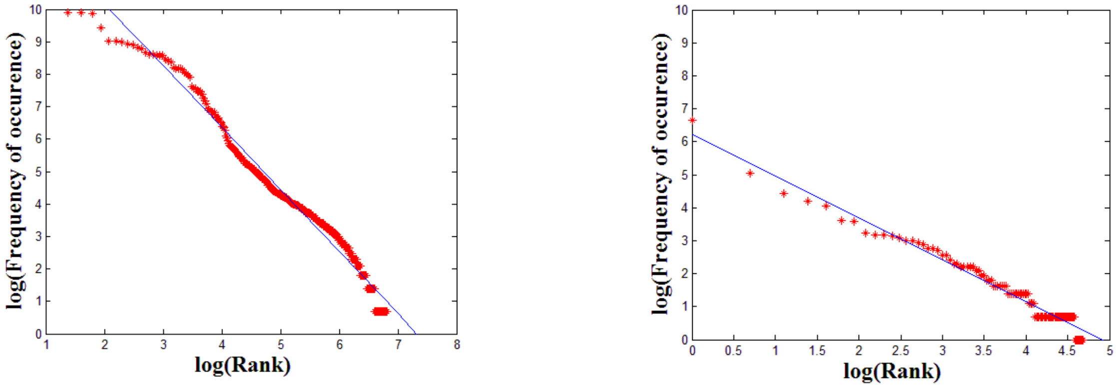


Figure 2. Some Examples for LPM Outputs and Zipf's distribution.



(a) S = -1.91 , $R^2$ = 0.96 (b) S= -1.36 , $R^2$ = 0.97

Figure 3. Zipfian distributions of random sequenced voices with lengths up to (a) 10000 and (b) 20.

## 7 THE DESIGN OF LPM SEQUENCER

In this section, sequencing LPM voices is taken as a search problem for producing the required melodic structure. Designing such a system gives raise to the following questions:

- How to design an efficient search space traversal which resolves the sequencing problem within the constraints of given hardware resources.
- How to sequence voices in a musical manner? What are the defining musical critiques?
- What are the possible genotypes and phenotypes of a musical sequencer based on LPM?
- Is there a measurement for the creativity of the system itself?

Applying Genetic Algorithms for search and optimization of musical sequences has special requirements. For example, defining the search space; specifying the musical knowledge and rule representation; and the choice of an appropriate fitness function [36]. The search for finding optimal solutions is guided by assigning higher fitness to competent individuals. Since there are, in effect, infinite possibilities for producing music; it is necessary to define suitable constraints for limiting the search space.

As stated in the previous section LPM outputs are a set of voices instead of notes. The voices resemble musical motives of varying lengths depending on the number of cellular automata iterations involved in their production. The design of competent genotypes and phenotypes are requirements for an efficient search. The genotypes are codes which manifest a higher level of behaviour known in the phenotype. For example the eye colour is coded in genes. However, what is seen as blue, green, and etc., are the phenotypes. It should be noted that in the LPM system, the phenotypes are the voices which are heard as the behaviours of the individuals and the genotypes are the set of genes coded whether as binary or integer representations.

A first naïve design for the search space would be to define the individuals as the elements of voices set. Regarding the huge search space and our current facilities, software implementation is nearly impossible unless the search space is reduced by a notable amount. Perhaps selecting a limited number of voices and evolving them would be a more feasible solution. During the evolution of such a design, all the contributing parameters change dynamically to a point that fulfils predefined musical expectations or at least tries to do so. This stabilization includes a gradual justification of musical parameters and general improvement in each generation. There are no unique solutions to musical problems, In fact starting from the same initial conditions, the search may result to different sets of solutions in every execution.

Further improvement in the design is to divide the search problem into several multi-optimization ones, relating to the constituent elements of the produced melody based on the LPM output. The first search determines the structure of the melody, including the pitch frequency, the intervals and the note durations. The second search problem involves the optimization of the remaining synthesizer elements. This separation provides two categories of different natures for exploration. The search pool sizes of which becomes $88^2 * 20^2 * t$ and $88^7 * 20^7$ respectively. Evaluators for the individuals of each of these search spaces vary. This paper focuses on the first optimization problem though. The related fitness function scores every one of the individuals based on their statistical aesthetical competence, coded in their individual genes.

For crossover and mutation after selection operator, various methodologies can be thought of. The crossover operator can be defined as swapping the codes of the related voice producer parameters. By this methodology it is guaranteed that the newly born individuals are those previously existing in the grand pools which are given the chance of being investigated musically towards the aims of the genetic algorithm.

## 8 CONCLUSIONS & FUTURE WORK

CA evolution have been employed as a controller for the parameters of a synthesizer. Computational intelligence models as well as cellular automata are sources of creativity which can produce musical material without contributing to human knowledge. This research requires working with exploratory, and transformational types of creativity. Evolutionary algorithms have been found to be well-suited for this kind of navigations. Genetic algorithms have been chosen as a creativity exploratory tool for evolving sequence of voices.

LPM software, equipped with cellular automata and synthesis tool kit, has been introduced as an assisting tool for producing music. This paper provides a conceptual approach towards the design of a computational framework for sequencing LPM voices. We have described the problem of sequencing voices from a creativity point of view. Some existing visions towards computational creativity have been discussed. The dimension of the search space have been determined regarding the number of elements involved in voice generation and the components related to producing the melody. The search space is then divided to different categories regarding their nature as two different optimization problems. These include the psychoacoustic and melodic structure of LPM output. We are developing an evolutionary environment to enable this. Aesthetical measurements based on Zipf's law have been propounded as a base for designing fitness function for the optimization problem. Although, Zipf's law can be considered as a good approach for investigating the pleasantness of the output melody, there are other approaches which can be taken into account. Experiments of this kind (measuring the frequency distribution) are to be necessary but not sufficient conditions for investigating the aesthetical aspects of the phenomena (music in our case). However, they have been taken as an integral part in the design of the fitness function in the first stage. The next level of evaluation could contribute to human auditory tests in the form of survey.

The future research direction includes the design of fitness function for the multi optimization problem of sequencing LPM outputs.

## REFERENCES


[1] H. Farhat, *The Dastgah Concept in Persian Music*. Cambridge University Press, 1990.

[2] E. Zonis, "Contemporary Art Music in Persia," *Music. Q.*, vol. 51, no. 4, pp. 636–648, 1965.

[3] J. During, "Dastgah, Encyclopedia Iranica," *Iranica*. 2011.

[4] "Radif (music)," *Wikipedia, the Free Encyclopedia*. .

[5] L. Englander, C. Lecocq, and G. Kerker, "The Radif of Persian Music : Studies of Structure and Cultural Context by Bruno Nettl ; Carol M . Babiracki ; Béla Foltin; Daryoosh Shenassa ; Amnon Shiloah ; Accompanying Cassette : The Persian Radif in Comparative Perspective by Bruno Nettl ; La musiq," 2015.

[6] P. Heydarian and J. D. Reiss, "The Persian Music and the Santur Instrument," *Proc. 6th Int. Soc. Music Inf. Retr. Conf.*, pp. 524–527, 2005.

[7] J. McCormack, "Facing the Future: Evolutionary Possibilities for Human-Machine Creativity," *Art Artif. Evol. A Handb. Evol. Art Music*, pp. 417–451, 2007.

[8] J. D. Fernández and F. Vico, "AI methods in algorithmic composition: A comprehensive survey," *J. Artif. Intell. Res.*, vol. 48, pp. 513–582, 2013.

[9] Boulanger, *The Csound Book: Perspectives in Software Synthesis, sound design, signal processing, and programming*. MIT Press, 2000.

[10] M. Puckette, "Max at seventeen," *Comput. Music J.*, vol. 26, no. 4, pp. 31–43, 2002.

[11] D. Cope, "An expert system for computer assisted music composition," *Comput. Music J.*, vol. 11, no. 4, pp. 30–46, 1987.

[12] D. Cope, *EMI: Experiments in musical intelligence*. A-R Editions, 1996.

[13] J. a. Biles, "GenJam: A genetic algorithm for generating jazz solos," *Proc. Int. Comput. Music Conf.*, pp. 131–137, 1994.

[14] N. V. Woods and N. V. Woods, "No Title," no. May, pp. 1–58, 2009.

[15] C. Turner, "Liquid Brain Music," University of Hull, 2008.

[16] G. Diaz-Jerez, "Composing with Melomics: Delving into the computational world for musical inspiration," *Leonardo Music J.*, vol. 21, pp. 13–14, 2011.

[17] T. Anders, "Composing Music by Composing Rules: Design and Usage of a Generic Music Constraint System," *Soc. Sci.*, vol. 10, no. February, p. 412, 2007.

[18] C. Ariza, "The interrogator as critic: The Turing test and the evaluation of generative music systems.," *Comput. Music J.*, vol. 33, no. 2, pp. 48–70, 2009.

[19] M. Y. Lo, "Evolving Cellular Automata for Music Composition with Trainable Fitness Functions," *Electron. Eng.*, no. March, 2012.

[20] M. T. Pearce and G. a Wiggins, "Towards A Framework for the Evaluation of Machine Compositions," *AISB Symp. AI Creat. Arts Sci.*, pp. 22–32, 2001.

[21] G. Ritchie, "Some Empirical Criteria for Attributing Creativity to a Computer Program," *Minds Mach.*, vol. 17, no. 1, pp. 67–99, 2007.

[22] M. a. Boden, "Computer models of creativity,"

*Psychologist*, vol. 13, no. 2, pp. 72–76, 2000.
[23] B. Manaris, J. Romero, P. Machado, D. Krehbiel, T. Hirzel, W. Pharr, and R. B. Davis, "Zipf's Law, Music Classification, and Aesthetics," *Comput. Music J.*, vol. 29, no. 1, pp. 55–69, 2005.
[24] S. Arshi and D. N. Davis, "Towards a Fitness Function for Musicality using LPM," *6th York Dr. Symp.*, 2015.
[25] A. W. Burks, "Von Neumann's Self-Reproducing Automata," *Essays on Cellular Automata*. pp. 3–64, 1970.
[26] A. Burks, *Essays on Cellular Automata*. University of Illinois Press, 1970.
[27] S. Wolfram, *A New Kind of Science*. 2002.
[28] D. Burraston and E. Edmonds, "Cellular automata in generative electronic music and sonic art: a historical and technical review," *Digit. Creat.*, vol. 16, no. 3, pp. 165–185, 2005.
[29] D. Burraston, E. Edmonds, D. Livingstone, and E. R. Miranda, "Cellular Automata in MIDI based Computer Music," *Proc. Int. Comput. Music Conf.*, vol. 4, pp. 71–78, 2004.
[30] E. R. Miranda, "Evolving cellular automata music: From sound synthesis to composition," *Proc. Work. Artif. Life Model. Music. Appl.*, p. 12, 2001.
[31] E. R. Miranda, "Sounds of artificial life," in *Proceedings of the 4th Conference on Creativity & and Cognition, ACM*, 2002, pp. 173–177.
[32] P. Hoffmann, "Towards an Automated Art: Algorithmic process in Xenakis' Composition," *Contemp. Music Rev.*, vol. 21, no. 2–3, pp. 121–131, 2002.
[33] E. R. Miranda, "Cellular automata music: From sound synthesis to musical forms," *Evol. Comput. Music*, pp. 170–193, 2007.
[34] C. Darwin, "The Origins of Species," in *Chapter 4, Natural selection*, London, England: OUP Oxford; Revised edition edition (13 Nov. 2008), 1906.
[35] B. P. Buckles and F. Petry, *Genetic Algorithms*. 1992.
[36] a R. Burton and T. Vladimirova, "Generation of musical sequences with genetic techniques," *Comput. Music J.*, vol. 23, no. 4, pp. 59–73, 1999.
[37] D. E. Goldberg, *Genetic Algorithm in search, optimization, and machine learning.* 1989.
[38] A. Horner and D. E. Goldberg, "Genetic algorithms and computer-assisted music composition," in *In Proceedings of the International Conference on Genetic Algorithms*, 1991.
[39] B. L. Jacob, "Composing with genetic algorithms," *Proc. 1995 Int. Comput. Music Conf.*, no. September, pp. 452–455, 1995.
[40] Biles, "GenJam," 2014. [Online]. Available: http://igm.rit.edu/~jabics/GenJam.html. [Accessed: 01-Oct-2014].
[41] P. Laine and M. Kuuskankare, "Genetic algorithms in musical style oriented generation," *First IEEE Conf. Evol. Comput.*, pp. 858–861, 1994.
[42] P. Dahlstedt, "Autonomous evolution of complete piano pieces and performances," *Proc. 9th Eur. Conf. Artif. Life*, no. 1999, 2007.
[43] M. Lo and S. M. Lucas, "Evolving Musical Sequences with N-Gram Based Trainable Fitness Functions," *2006 IEEE Int. Conf. Evol. Comput.*, no. 1, pp. pp. 601–608, 2006.
[44] R. Wooller and A. R. Brown, "Investigating morphing algorithms for generative music," *Procedings Third Iteration Third Int. Conf. Gener. Syst. Electron. Arts*, 2005.
[45] M. Farbood and B. Schoner, "Analysis and synthesis of Palestrina-style counterpoint using Markov Chains," in *In Proceedings of the International Computer Music Conference*, 2001.
[46] K. M. Biyikoglu, "A Markov Model for Chorale Harmonization," in *In Proceedings of the ESCOM Conference*, 2003.
[47] R. F. Voss and J. Clarke, "'1/f noise' in music: Music from 1/f noise," *Journal of the Acoustical Society of America*, vol. 63, no. 1. pp. 258–263, 1978.
[48] N. V. Woods, "Liquid Brain Music: Phase II," no. May, 2009.
[49] P. R. Cook and G. P. Scavone, "The Synthesis ToolKit in C++ (STK)," no. 09/10/2008. 2008.
[50] W. Li and P. Norman, "The Structure of the Elementary Cellular Automata Rule Space," *Complex Syst.*, vol. 4, pp. 281–297, 1990.
[51] D. N. Davis, "Computer and Artificial Music: Liquid Persian Music." [Online]. Available: http://www2.dcs.hull.ac.uk/NEAT/dnd/music/lpm.htm l. [Accessed: 01-Sep-2015].
[52] E. Jen, "Global properties of cellular automata," *J. Stat. Phys.*, vol. 43, no. October, pp. 219–242, 1986.