Genetic Evolution of L and FL–systems for the Production of Rhythmic Sequences

Maximos A. Kaliakatsos–Papakostas
Computational Intelligence Laboratoty (CILab),
Department of Mathematics,
University of Patras,
GR–26110 Patras, Greece
maxk@math.upatras.gr

Andreas Floros
Department of Audio and Visual Arts,
Ionian University,
GR–49100 Corfu, Greece
floros@ionio.gr

Nikolaos Kanellopoulos
Department of Audio and Visual Arts,
Ionian University,
GR–49100 Corfu, Greece
kane@ionio.gr

Michael N. Vrahatis
Computational Intelligence Laboratoty (CILab),
Department of Mathematics,
University of Patras,
GR–26110 Patras, Greece
vrahatis@math.upatras.gr

ABSTRACT

Music composition with algorithms inspired by nature has led to the creation of systems that compose music with rich characteristics. Nevertheless, the complexity imposed by unsupervised algorithms may arguably be considered as undesired, especially when considering the composition of rhythms. This work examines the composition of rhythms through L and Finite L–systems (FL–systems) and presents an interpretation from grammatical to rhythmic entities that expresses the repetitiveness and diversity of the output of these systems. Furthermore, we utilize a supervised training scheme that uses Genetic Algorithms (GA) to evolve the rules of L and FL–systems, so that they may compose rhythms with certain characteristics. Simple rhythmic indicators are introduced that describe the density, pauses, self similarity, symmetry and syncopation of rhythms. With fitness evaluations based on these indicators we assess the performance of L and FL–systems and present results that indicate the superiority of the FL–system in terms of adaptability to certain rhythmic tasks.

Categories and Subject Descriptors

F.4.2 [**Grammars and Other Rewriting Systems**]: Parallel rewriting systems; H.5.5 [**Sound and Music Computing**]: Modeling

General Terms

Grammars, Genetic Evolution

Keywords

L–systems, FL–systems, Genetic Algorithms, rhythm, rhythmic indicators

1. INTRODUCTION

Natural phenomena have inspired the formulation of algorithms that are expressed by simple rules but their output presents rich geometric and dynamical properties. The diversity and complexity presented by the outcome of bio–inspired algorithms has urged many researchers to use these algorithms for music composition. On the other hand, intelligent algorithms allow the effective optimization of results towards a provided target. Two approaches for music composition with the utilization of the aforementioned algorithms are distinctive: unsupervised and supervised. The first category uses an initiated set of rules which unfold into a diverse output that is translated into music [3, 4]. Supervised algorithms compose music which moves towards a target which is either bounded by music theory [6], by similarity to existing pieces [1, 18], or subjectively driven by the user [2, 13].

In algorithmic music composition systems the rhythmic part plays an important role for the contingency of the composition. In the paper at hand, we examine the composition of rhythms by two *unsupervised* intelligent methodologies, the L–systems and the Finite L–systems (FL–systems). We propose and utilize an evolutionary scheme to evolve L and FL–systems towards targeted rhythmical tasks, a procedure that provides a *supervised* "track" for these unsupervised methodologies. In contrast to the majority of previous approaches that examine music composition systems, the results obtained in the present work provide a large scale evaluation procedure with additional pointers to *objective* criteria that concern the complexity of the composed rhythms.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.

The evolution of grammars with genetic algorithms has been used for various tasks like the description of the retina [16], the creation of virtual creatures [12] and the modeling of leaf shapes [22] among others. The utilization of L–systems for music composition has been examined with the production of music score directly from curves [20] and by interpreting the L–systems directly to music [28]. A thorough review and presentation of more approaches can be found in [19]. Furthermore, music composition from L–systems has been embodied into a Genetic Algorithm (GA) context [5, 17].

The FL–systems [14] utilize a recursive scheme and evolve a set of initial axioms, a procedure that generates orbits of string sequences that are directly interpreted to rhythmic patterns. The produced rhythmic sequences were demonstrated to cover a wider range of complexity compared to the ones created by L–systems in an experimental setup that incorporated the use of random rules. The utilization of GA in this work amplifies these findings and expose the lack of flexibility that the L–systems present, on contrast to the FL–systems. We thus present indications that the combination of FL–systems with GA can produce a flexible supervised rhythm composition scheme with the ability to create a wide range of rhythms, from diversified and complicated to harnessed and repetitive.

The paper at hand is organized as follows: In Section 2 we provide a brief description of the Deterministic Context–free L–systems, the FL–systems and the proposed interpretation of strings to rhythmic sequences. Section 3 describes the examined methodology for producing a compound of GA and the aforementioned rules in order to create rhythmic sequences with desired properties. This Section also introduces the rhythm indicators considered as descriptors of the desired properties that lead to the realization of fitness evaluations. In Section 4, the experimental setup is thoroughly described and an overview of the results is presented. Finally, Section 5 provides conclusions and pointers to future work.

2. CONSTRUCTION OF RHYTHMIC SEQUENCES WITH L AND FL–SYSTEMS

This Section first presents the approach that we follow to construct rhythmic sequences from symbolic string sequences and then describes the L and FL–systems. This string–to–rhythm interpretation projects the repetitiveness of characters into the strings produced by the L and FL–systems to musical events. Thus string sequences with more repeating patterns lead to repetitive rhythms and vice versa.

2.1 Modeling rhythmic sequences

In the rest of the paper we will be referring to two different representations of rhythmic sequences: the *binary* and the *interval vector* [26] representation. We may also employ the term *quasi–binary* representation to include more musical information as described later. With the binary representation, a sequence of digits represents the rhythmic actions that occur during equally spaced time intervals. Considering the division of a 4/4 measure of a music piece in sixteenths, for example, we may represent this measure with 16 digits. Digit 1 signals an onset event while digit 0 marks the continuation of the previous event. By utilizing more digits than 1 and 0 we may incorporate further rhythmical information. In the experiments described in Section 4 for example, we have utilized the digit (-1) as a signal for a pause event.

By grouping the cumulative duration of rhythmic events according to the considered time resolution we obtain the interval vector representation. With the interval vector representation, each rhythmic sequence is described as a series of numbers in the form of a vector. Each number denotes the consecutive time subdivisions that correspond to a single rhythmic event. The sum of all the numerical values of the interval vector representation within a measure should add up to the total number of subdivisions of this measure in respect to the considered time resolution. For example, if we consider the division of a 4/4 measure in 32nds, then the sum of all the arithmetic values that describe this measure is 32. An example of a quasi–binary string and its interval vector representation is demonstrated in Table 1.

The L and FL–systems produce a series of symbols, or a word in an alphabet. Suppose that we have an alphabet with n letters, $V = \{X_1, X_2, \ldots, X_n\}$ and a nonempty word in this alphabet $\lambda \in V^+$. We denote a series of k consecutive instances of a letter X_i as X_i^k and rewrite the word λ using this denotation, e.g. if $\lambda = AAABBA$, then we have $\lambda = A^3B^2A^1$. We call this word representation *cumulative representation*. With the cumulative representation we proceed to the rhythmic interval vector representation by simply concatenating the exponents, i.e. the word $\{X_{i_1}^{k_1}, X_{i_2}^{k_2}, \ldots, X_{i_m}^{k_m}\}$ is the rhythm $k_1k_2\ldots k_m$. Table 1 also shows the transformation of a string into rhythm, with each symbol of the string describing different events.

Table 1: Example transformation of a string sequence to its interval vector and quasi–binary representation.

V :	$\{A, B, P\}$
A, B :	onset event
P :	pause
ω :	$AAAABBAAPPBBBBPP$
cumulative:	$A^4B^2A^2P^2B^4P^2$
interval vector:	422242
quasi–binary:	10001010(-1)01000(-1)0

2.2 Deterministic Context–free L-systems

The L–systems are parallel generative grammars [21] with some variations that allow the production of interesting patterns that resemble plant–like forms and fractals. In the paper at hand we use the L–systems belonging to the simplest form which are called *Deterministic Context–free*, with the acronym DOL–systems. In these systems an alphabet denoted by V is defined. This alphabet is a set of symbols $V = \{X_1, X_2, \ldots, X_n\}$ which allows the creation of words. The set of non–empty words of an alphabet V is denoted as V^+. Each symbol X_i is associated with a rewriting rule, denoted as R_{X_i}, with the set of rules being denoted as $P = \{R_{X_i}\}_{i\in[1,2,\ldots,n]}$. The rules P are then serially applied to each letter of a nonempty word in the above mentioned alphabet, $\omega \in V^+$, creating a new word. A DOL–system can thus be described as a triplet $G = \langle V, \omega, P\rangle$. We use the resulting words of the DOL–systems for the construction of rhythmic sequences with the methodology described earlier. Therefore, the rewriting procedure is applied recursively, until the length of the resulting word creates a rhythmic sequence with the desired length. An example of an L–system with the above form is demonstrated in Table 2.

Table 2: Example simulation of a DOL–system for 3 iterations.

V :	$\{A, B\}$
ω :	AB
P :	$R_A = A \rightarrow AB$
	$R_B = B \rightarrow A$
0)	AB
1)	ABA
2)	$ABAAB$
3)	$ABAABABA$

The rich structure that emerges from L–systems, even of the simplest form of the DOL–systems, reveals the composition of music that abounds in interesting tonal variability but this diversity results to unstable rhythm, exposing its non–human origin. For example, the L–system examined in [29] was reported to produce notes that "*do not fit well into 4/4 score notation, because many of the notes are off-beat*". To overcome this problem, some works have examined the utilization a constrained set of rhythmic values [7], but the resulting rhythm was repetitive and uninteresting. Similar effects have been noted for several variations of L–systems, with the utilization of various methods for the transition from symbols to music, providing several rhythmic interpretations [19].

2.3 Finite L–systems

We define an FL–system [14] as a triplet $G = \langle V, \Omega, P \rangle$, where $V = \{X_1, X_2, \ldots, X_n\}$ is an alphabet, Ω is a set of continuously updating axioms ω_i, $i \in \{1, 2, \ldots, m\}$, and $P = \{R_{X_i}\}_{i \in [1,2,\ldots,n]}$ is a set of rules. The strings that constitute the set of rules, P, and the axioms, Ω, are nonempty words of V^+. In contrast to the L–systems, the FL–systems have no fixed axiom, but instead, the axioms are being updated when an orbit is reached. The number of updates (m), depends on the length of the piece and the length of the orbit of each axiom. In each iteration the L-systems produce a word of variable size which depends on the substitution rules. The FL–systems on the other hand, produce sequences of words with fixed length which depends on the time analysis of the applied measure. Specifically, the length of the initial axioms ω_i and the word produced by an FL–system in each iteration is the number of symbols needed to fill the current measure. This length is denoted by α. Each word produced by an FL–system represents the rhythm in a measure and each axiom produces a set of words that we call orbit and describe later.

For the construction of rhythmic sequence with an FL–system we consider an FL–system $G = \langle V, \Omega, P \rangle$. We begin with the application of the set of rules P on the axiom $\omega_1 \in \Omega$, denoting the resulting word with $\lambda_1(\omega_1)$. The length of the word $|\lambda_1(\omega_1)|$ may vary, depending on the length of the strings that constitutes the substitution rules. If this length is different than the length needed for the desired time analysis, α, we perform the *trimming* procedure which comprises of following two actions in the respective cases:

1. if this length is smaller than α, i.e. $|\lambda_1(\omega_1)| < \alpha$, then we substitute $\lambda_1(\omega_1)$ with concatenation of $\lambda_1(\omega_1)$ with itself ($\lambda_1(\omega_1) = [\lambda_1(\omega_1)\lambda_1(\omega_1)]$) until $|\lambda_1(\omega_1)| \geqslant \alpha$ and then apply the next step, or
2. if $|\lambda_1(\omega_1)| > \alpha$, then we substitute $\lambda_1(\omega_1)$ with the string that includes its first α symbols.

This procedure "trims" each word to the required size. More clearly, the first case may happen only if P contains a sufficient number of empty rules, i.e. rules that substitute a symbol with an empty word. We then obtain the next word of the sequence, $\lambda_{n+1}(\omega_1)$ by applying the rules P on $\lambda_n(\omega_1)$ and then *trimming* $\lambda_{n+1}(\omega_1)$.

By recursively following this procedure, we find a pair of integers ρ and τ, so that for each index $i > \rho$ we have $\lambda_i(\omega_1) = \lambda_{i+\tau}(\omega_1)$. The application of rules P on each $\lambda_i(\omega_1)$ for $i > \rho$, provides a sequence of repeating words with period τ. We call the first occurrence of repeating words the *orbit* of axiom ω_1 within the rules P. This orbit of τ words creates the rhythmic sequence obtained by the axiom ω_1. We continue by updating axiom ω_1 and producing the orbits and rhythms of the axiom $\omega_2, \omega_3, \ldots, \omega_m$, until the necessary number of measures is covered.

Table 3: Example simulation of the first axiom ω_1 of an FL–system.

V :	$\{A, B\}$
A, B :	onset event
P:	$R_A = A \rightarrow BAA$
	$R_B = B \rightarrow AAB$
ω_1 :	$AAAABBBABAAABBAA$
iterations: $\lambda_1(\omega_1)$:	$AAAABBBABAAABBAA$
$\lambda_2(\omega_1)$:	$BAABAABAABAAAABA$
$\rho = 3$ $\lambda_3(\omega_1)$:	**AABBAABAAAABBAAB**
$\lambda_4(\omega_1)$:	**BAABAAAABAABBAAB**
$\tau = 2$ $\lambda_5(\omega_1)$:	$AABBAABAAAABBAAB$
$\lambda_6(\omega_1)$:	$BAABAAAABAABBAAB$
orbit:	$AABBAABAAAABBAAB$
	$BAABAAAABAABBAAB$
rhythm:	1010101100010101
	0101100011010101

Table 3 shows the procedure described above to produce the rhythmic sequence from the first axiom ω_1 of an FL–system. It has to be noted that both words in the orbit string, are considered adjusted, i.e. the beginning B symbol of the second word is continuing the concluding B symbol of the first word. The orbit string may also be considered cyclic, i.e. the first symbol of the first word may be considered as the extension of the last symbol of the last word.

3. GENETIC EVOLUTION AND FITNESS EVALUATION

The methodology described above allows the production of rhythmic sequences given an L or an FL–system. In order to constitute the utilization of Genetic Algorithms (GA) applicable on the grammars under discussion, we consider a chromosome representation of the substitution rules that describes these grammars. Furthermore, we discuss a set of rhythmic features that capture several characteristics of a rhythmic sequence and use them as fitness evaluators for the rhythms produced by the systems under examination.

3.1 The chromosome of rules

The output of an L or an FL–system depends mainly on the set of substitution rules that are applied on the axioms. These substitution rules are as much as the symbols of the alphabet, since each symbol on the axiom is substituted by a string provided by the rule. If a symbol appears to have no substitution rule we consider that this rule is the empty string. Given an alphabet $V = \{X_1, X_2, \ldots, X_n\}$ we may formally express the set of rules $P = \{R_{X_1}, R_{X_2}, \ldots, R_{X_n}\}$, where each R_{X_i} is a string that substitutes the letter X_i in the axiom strings. The length of the rule string, denoted as $|R_{X_i}|$, is not fixed and may even be zero if this rule is the empty string (no substitution rule).

As a first step towards constructing a rule chromosome, we consider a maximum rule length that we denote by $\mathcal{M}$, thus $\forall\, i \in 1, 2, \ldots, n$ we have that $|R_{X_i}| \leqslant \mathcal{M}$. Then we consider a *void* character, denoted by `*`, which is not interpreted to any symbol into the final string and is only used for equalizing the length of all rules. This character is added at the end of every rule string as many times need so that the length of each rule becomes $\mathcal{M}$. For example, if we have $\mathcal{M} = 4$ and $R_{X_k} = AB$, with $|R_{X_k}| = 2$, then we transform this rule to $R^*_{X_k} = AB{*}{*}$. The transformed version $R^*_{X_i}$ of a rule R_{X_i} is called the *augmented version*. In this context, the empty string rule can be expressed as $R^*_{X_j} = {*}{*}{*}{*}$. With this process each rule has a predefined fixed length, $\mathcal{M}$, and the genuine length of a rule is computed as the number of the non–`*` characters.

This formulation allows the representation of a chromosome through a single string, denoted with γ, with length $n\mathcal{M}$, where n is the number of rules and $\mathcal{M}$ is the maximum length of the rule strings. The chromosome string γ is constituted of n $\mathcal{M}$–tuples representing the augmented version of the respective rule, i.e. the i–th $\mathcal{M}$–tuple is the augmented version $R^*_{X_i}$ of R_{X_i}. Table 4 demonstrates the transformation procedure from a set of rules to a rule chromosome and Table 5 the reverse procedure, from a rule chromosome to a set of rules.

Genetic operators can be applied to the rule chromosomes of L and FL–system populations and create novel rule sets, enabling the application of the standard GA methodology [11]. To this end, provided a fitness evaluation methodology (discussed later) on the *phenotype* (rhythms) of the aforementioned chromosomes, the genetic procedure may converge to rhythms with certain desired attributes. In this context, a population of L or FL–systems is created with different rules and this population is evolved into a new *generation*. The preservation of the desired characteristics can be accomplished by garnering a portion of the individuals in each generation using a selection scheme and create a new generation by applying to them a set of genetic operators.

3.2 Rhythmic indicators as fitness evaluators

The formulation of efficient rhythmic features has been a subject of intense research effort, yielding results for features that indicate rhythm similarity [27] and complexity [25]. The aim of these approaches is to create a set of rules that is applicable on the comparison and categorization of rhythms. In this work however, we do not need a "global" set of features, but we rather need a set of indicators that describe general characteristics of a rhythmic sequence. These indicators may then be used as fitness indicators for the application of GA. An example rhythm and the values of its indicators, as described below, is demonstrated in Table 6.

1. **Density**: This indicator describes the number of events within a rhythmic sequence. It is calculated from the quasi–binary rhythm representation by dividing the number of the non–zero elements of the rhythm with the total length of the sequence. The density of a rhythmic sequence is a number between 0 and 1.

2. **Pauses**: This indicator gives the percentage of pauses in the rhythmic sequence. It is calculated be dividing the number of the -1 digits of the quasi–binary representation with the length of the sequence. This feature's value is between 0 and 0.5.

3. **Self Similarity**: Self similarity measures the similarity of the rhythmic sequences in different bars. A rhythm with extreme self similarity would be constituted of the same rhythmic sequence in every bar. We measure the similarity of a rhythmic sequence by considering the mean value of the linear correlation among every pair of the rhythms in each bar. This feature may have a value between -1 and 1.

4. **Symmetry**: The symmetry of a rhythm can be described as the repetitiveness of the distances of consecutive onset events. For example, an extremely symmetric rhythm would have equally spaced inner–onset intervals, which means that its interval vector representation would have repetitions of an integer. We measure the symmetry of a rhythmic sequence by dividing the standard deviation with the mean value of the values in its interval vector representation.

5. **Syncopation**: The syncopation of a rhythms has been thoroughly studied with complex theoretic models [9] and cognitive subjective studies [15] among others. A general outcome of all these approaches is that syncopated rhythms are characterized by increased offbeat onsets, i.e. increased ratio of onsets on the even beats to onsets on odd beats. We consider a rather naive measurement of syncopation that is not completely accurate but it is descriptive: the syncopation of a rhythmic sequence is considered as the ratio of even onset events divided with the odd onset events. In the case where we have no odd onset events we consider an arbitrary high penalty value.

All five rhythmic indicators are combined linearly to form a single fitness evaluation. Each indicator value is normalized with the inverse of its range, and the final fitness value is assessed by the sum of all normalized indicator values. This paper addresses a comparison of the adaptability of the L and FL–systems and so the described "coarse" computation of fitness is sufficient. Furthermore, the descriptive nature of the rhythm indicators does not pronounce the necessity of "fine" Pareto optimization. Nevertheless, future research that focuses on the optimization of the rhythmic sequences

Table 4: Composition of the rule chromosome from string rules.

P: R_{X_1} R_{X_2} $\vdots$ R_{X_n}	$\rightarrow$ P^*: $R^*_{X_1}$ $R^*_{X_2}$ $\vdots$ $R^*_{X_n}$	$\rightarrow$ $\gamma = \left[R^*_{X_1} R^*_{X_2} \dots R^*_{X_n}\right]$

Table 5: Decomposition of the rule chromosome to the string of rules.

$\gamma = \Bigg[\underbrace{\overbrace{\overbrace{xx\dots x}^{\mathcal{M}\colon R^*_{X_1}}\overbrace{xx\dots x}^{\mathcal{M}\colon R^*_{X_2}}\dots\overbrace{xx\dots x}^{\mathcal{M}\colon R^*_{X_n}}}^{n}}_{n\mathcal{M}}\Bigg]$	$\rightarrow$ P^*: $R^*_{X_1}$ $R^*_{X_2}$ $\vdots$ $R^*_{X_n}$	$\rightarrow$ P: R_{X_1} R_{X_2} $\vdots$ R_{X_n}

Table 6: An four–bar example rhythm with 16 beat analysis and the values of its indicators.

Rhythm																***Indicator***	
1	−1	1	−1	1	1	−1	1	−1	1	−1	1	1	−1	1	1	Density:	0.5938
1	1	0	−1	0	0	1	1	1	−1	1	−1	1	1	−1	1	Pauses:	0.4474
1	−1	1	1	−1	1	0	0	−1	0	1	1	1	0	1	0	Self Similarity:	0.2520
1	−1	1	1	1	1	1	0	1	−1	1	−1	1	1	−1	1	Symmetry:	0.4460
																Syncopation:	0.7273

per se, should incorporate the principles of multi–objective optimization.

4. RESULTS

We present a comparison of the L and FL–systems in terms of adaptability to certain rhythmical constraints. The rules of the L and FL–systems are adjusted over their chromosomic expression using a GA scheme. The fitness evaluation of an individual's phenotype (the rhythm it produces) is provided by computing the sum of distances of each rhythm indicator from a target value. We have created a set of 100 different random rhythm indicator combinations and used them as fitness values for assessing the results of 100 GA simulations. In these simulations each rhythm indicator is a value in the respective range, i.e. density is in $[0, 1]$, pauses in $[0, 0.5]$, self similarity in $[-1, 1]$, symmetry in $[0, 1]$ and syncopation in $[0, 2]$.

The genetic operators that we included during the evolutionary process, are the *crossover*, the *character mutation* and *substring mutation* with probability 0.80, 0.05 and 0.15 respectively. The crossover operator acts on the chromosome of two individuals by splitting their string representation to a certain point and cross–exchanging the four parts. The character mutation operator acts on a single chromosome string and substitutes a character in a random place within the string with a random character from the alphabet. The substring mutation operator substitutes a substring in a random place within a chromosome string with a new random string of characters in the alphabet with the same length. The selection scheme that we utilized was *roulette* selection, where each individual has a probability of being selected that is proportional to its fitness superiority against the rest individuals in the population.

The results are obtained from evolving L and FL–systems in 100 different simulations with a fitness value provided by the aforementioned 100 target indicator sets. All L and FL–systems for every simulation were evolved for 50 generations with a population size of 20 individuals. The initial population had random rule strings and the axioms in every generation were random, thus an individual was only characterized by its rules. The time analysis was set to 16 and the alphabet constituted of 6 characters, 5 indicating a simple onset event and 1 indicating a pause event. The fitness value for each individual was computed from rhythmic sequences they composed and consisted of 50 measures.

Comparing the adaptability of L and FL–systems.

The FL–systems have been better adapted to the 100 random target tasks. The mean error for the FL–systems was 0.9939 with 0.4069 standard deviation while the respective quantities for the L–systems are 1.5121 and 0.5175. The FL–systems had a better fitness value in the 93% of the target tasks. The box plots of the fitness values of all 100 simulations for the L and FL–systems are illustrated in Figure 1.

Two features have been used to examine the information complexity of the rhythmic sequences with the best fitness value, the Shannon Information Entropy (SIE) [23] and the Compression Rate (CR) using the Ziv–Lembel compression algorithm [30]. The SIE of a rhythmic sequence corresponds to the SIE of the *rhythm Probability Density Function* (rPDF). This is computed from the binary representation form as the probability that a certain beat has an onset event. An example rhythmic sequence and its respec-

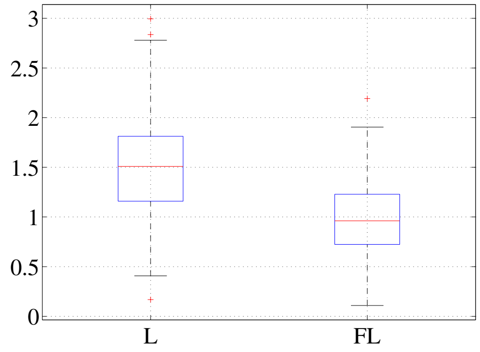

Figure 1: Box plot of errors produced during 100 training simulations with 50 generations of L and FL–systems.

tive rPDF is shown in Table 7. CR represents the ratio of the size of the compressed rhythmic sequence with the Ziv–Lembel algorithm to the size of the uncompressed sequence.

Table 7: Computation of the rPDF of a rhythmic sequence with analysis to fourths.

	1	0	1	1
	1	0	1	0
	1	1	1	1
	1	0	1	0
rPDF	0.36	0.09	0.36	0.18

A greater value for SIE, provided a specific time analysis value, indicates an rPDF closer to the uniform distribution. Greater CR values indicate the lack of repeating rhythmic patterns, since the Ziv–Lembel compression algorithm is based on locating repetitions of a substring within the string under compression. A combination of great SIE and CR values of a rhythmic sequence indicate that this sequence is more random and complex. Figure 2 illustrates the box plot of the SIE values and Figure 3 the CR values of the 100 best fitted rhythms for the L and FL–systems.

The SIE values of the rPDFs of rhythms produced by the FL–systems are systematically lower than the ones produced by the L–systems. This fact indicates that the FL–systems produce rhythms that have a certain rhythmic identity, with characteristic onset events happening on certain beats. The SIE values for the L–systems' rhythms are deteriorated near their mean value which is 2.7667. This value is near the SIE of the uniform distribution with 16 random events (2.7726). Furthermore, the mean CR of the quasi–binary representation of the L–systems' rhythms is 0.0273, a value that approaches the CR of random quasi–binary rhythms (around 0.0300). The FL–systems rhythms are more structured, with a mean CR 0.0146.

Analyzing the impact of each fitness indicator.

The fitness value of the rhythms produced by the L and FL–systems depends on the 5 fitness indicators mentioned earlier. Throughout the evolutionary process the results are optimized simultaneously for all the fitness indicators. Table 8 shows the errors produced for each indicator for the L and FL–systems where it is obvious that the FL–systems outperformed the L–systems in any indicator. The greatest difference is exhibited for the self similarity indicator (in bold) where the FL–systems reached a remarkable performance. It has to be noted that the extreme adaption of the FL–systems to the any random self similarity task does not mean that they only produce rhythms with great self similarity. On the contrary, it means that they may produce a great variety from diverse to repetitive rhythms, since the random fitness tasks for the self similarity had a range from -1 to 1.

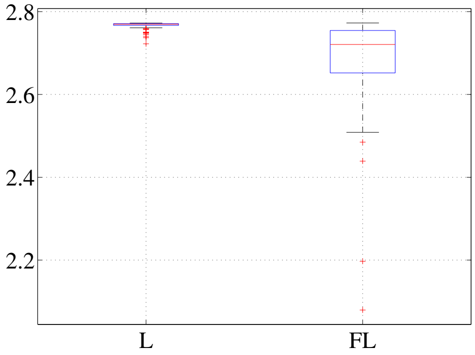

Figure 2: Box plot of the Shannon Information Entropy of the 100 best individuals for each simulation.

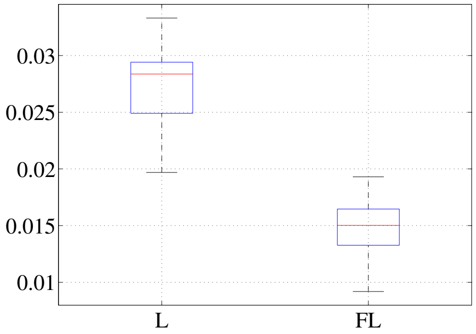

Figure 3: Box plot of the compressor rates of the 100 best individuals for each simulation.

Table 8: Mean errors among 100 simulations for each rhythmic feature separately.

Indicator	*L–systems*	*FL–systems*
Density	0.2724	0.2484
Pauses	0.3305	0.2979
Self Similarity	**0.4123**	**0.0792**
Symmetry	0.1859	0.1699
Syncopation	0.3109	0.1984

5. CONCLUSIONS

This work presents a methodology for applying Genetic Algorithms (GA) for the evolution of the rules that determine the L–systems and their finite variation, the Finite

L–systems (FL–systems), for the composition of rhythmic sequences. The L–systems construct a string by recursively applying their set of rules on an initial axiom, while the FL–system produce packages of string sequences that match the required length of the music measure, provided a time resolution. The strings produced by the L and FL–system are directly interpreted into rhythmic entities. The GA methodology becomes applicable on the rules of the aforementioned grammatical systems through the *augmented* rules representation that we introduce, which leads directly to the formulation of the *chromosome of rules*. The evolutionary process is led towards rhythms with predefined characteristics which are provided by a set of rhythm indicators that describe the density, pauses, self similarity, symmetry and syncopation. Results are presented among 100 random target tasks which indicate that the FL–systems are more flexible and adapt more efficiently to a presented task than the L–systems.

The customizability of the FL–systems not only in terms of rhythmic features, but also in terms of randomness and complexity, may provide a useful tool for automatic music composition that meets specified criteria. The interpretation from string to rhythm may be further developed in order to incorporate more rhythmic or even musical information. For example, the quasi–binary representation could be enriched with more digits that the ones we included in Section 2. These digits could represent rhythmic notions like intensity and polyphony, or even melodic content like notes or pitches. For demonstration of primitive work on these comments, we provide links to two sample compositions with FL–systems, one in which the rhythm, polyphony and intensity is provided by an FL–system [8], and the pitches by the logistic map [24], and another in which an FL–system also determines all rhythmic and tonal factors [10]. Future work would also incorporate the utilization of existing “target” rhythms for fitness evaluation, examining the potentiality of the discussed systems to carbon–copy entire rhythmic sequences.

6. REFERENCES

[1] M. Alfonseca, M. Cebrián, and A. Ortega. Evolving computer-generated music by means of the normalized compression distance. In *Proceedings of the 5th WSEAS international conference on Simulation, modelling and optimization*, SMO’05, pages 343–348, Stevens Point, Wisconsin, USA, 2005. World Scientific and Engineering Academy and Society (WSEAS).

[2] J. A. Biles. Genjam: evolution of a jazz improviser. In P. J. Bentley and D. W. Corne, editors, *Creative evolutionary systems*, pages 165–187. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[3] T. Blackwell. Swarming and music. *Evolutionary Computer Music*, pages 194–217, 2007.

[4] D. Burraston and E. Edmonds. Cellular automata in generative electronic music and sonic art: a historical and technical review. *Digital Creativity*, 16(3):165–185, 2005.

[5] A. O. de la Puente, R. S. Alfonso, and M. A. Moreno. Automatic composition of music by means of grammatical evolution. In *Proceedings of the 2002 conference on APL: array processing languages: lore, problems, and applications*, APL ’02, pages 148–155, 2002.

[6] P. Donnelly and J. Sheppard. Evolving four-part harmony using genetic algorithms. In *Proceedings of the 2011 international conference on Applications of evolutionary computation - Volume Part II*, EvoApplications’11, pages 273–282, Berlin, Heidelberg, 2011. Springer-Verlag.

[7] R. L. DuBois. *Applications of Generative String-Substitution Systems in Computer Music*. PhD thesis, Columbia University, 2003.

[8] finiteL: sample composition with the rhythm, polyphony and intensity controlled by FL–systems. https//sites.google.com/site/maximoskp/finiteL.mp3, 2012.

[9] W. T. Fitch and A. J. Rosenfeld. Perception and Production of Syncopated Rhythms. *Music Perception*, 25(1):43–58, Sept. 2007.

[10] hideBehind: sample composition with the rhythm, polyphony, intensity and pitches controlled by FL–systems. https//sites.google.com/site/maximoskp/finiteL.mp3, 2012.

[11] J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.

[12] G. S. Hornby and J. B. Pollack. Evolving l-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048, 2001.

[13] M. A. Kaliakatsos-Papakostas, M. G. Epitropakis, A. Floros, and M. N. Vrahatis. Interactive evolution of 8-bit melodies with genetic programming towards finding aesthetic measures for sound. In *Proceedings of the 1st International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design, EvoMUSART 2012*, volume 7247 of *LNCS*, pages 140–151, Malaga, Spain, apr 2012. Springer Verlag.

[14] M. A. Kaliakatsos-Papakostas, A. Floros, and M. N. Vrahatis. Intelligent generation of rhythmic sequences using Finite L-systems. In *Proceedings of the Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP 2012)*, page (To appear), Piraeus, Athens, Greece, 18-20 July 2012.

[15] P. E. Keller and E. Schubert. Cognitive and affective judgements of syncopated musical themes. *Advances in Cognitive Psychology*, 7:142–156, Dec. 2011.

[16] G. Kokai, R. Vanyi, and Z. Toth. Parametric L-system description of the retina with combined evolutionary operators. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1588–1595, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.

[17] B. F. Lourenc and C. P. Brand. L–systems, scores, and evolutionary techniques. In *Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference*, pages 113–118, 2009.

[18] B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro, and J. Romero. A corpus-based hybrid approach to music analysis and composition. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, pages 839–845. AAAI Press, 2007.

[19] S. Manousakis. Musical L-systems. Master’s thesis,

The Royal Conservatory, the Hague, the Netherlands, 2006.

[20] P. Prusinkiewicz. Score generation with L–systems. *Computer*, pages 455–457, 1986.

[21] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[22] Y. Rodkaew, C. Lursinsap, T. Fujimoto, and S. Siripant. Modeling leaf shapes using l-systems and genetic algorithms. In *In International Conference NICOGRAPH 2002*, pages 73–78, 2002.

[23] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5:3–55, January 2001.

[24] C. H. Skiadas, I. Dimotikalis, and C. Skiadas, editors. *Chaos Theory: Modeling, Simulation and Applications*. World Scientific Publishing Company, May 2011.

[25] E. Thul and G. T. Toussaint. Rhythm complexity measures: A comparison of mathematical models of human perception and performance. In *Proceedings of the 9th International Society for Music Information Retrieval Conference, (ISMIR 2008)*, pages 663–668, Philadelphia, USA, 2008.

[26] G. Toussaint. The geometry of musical rhythm. In *In Proc. Japan Conference on Discrete and Computational Geometry, LNCS 3742*, pages 198–212. Springer-Verlag, 2004.

[27] G. T. Toussaint. A comparison of rhythmic similarity measures. In *Proceedings of the 5th International Society for Music Information Retrieval Conference, (ISMIR 2004)*, pages 242–245, Barcelona, Spain, 2004.

[28] P. Worth and S. Stepney. Growing music: Musical interpretations of l-systems. In *EvoWorkshops*, Lecture Notes in Computer Science, pages 545–550. Springer, 2005.

[29] P. Worth and S. Stepney. Growing music: Musical interpretations of l-systems. In *EvoWorkshops*, volume 3449 of *Lecture Notes in Computer Science*, pages 545–550. Springer, 2005.

[30] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337– 343, May 1977.