BRIDGES
Mathematical Connections in Art, Music, and Science

A Mathematical Analysis of African, Brazilian and Cuban *Clave* Rhythms

Godfried Toussaint*
School of Computer Science
McGill University
Montréal, Québec, Canada

Abstract

Several geometric, graph theoretical and combinatorial techniques useful for the teaching, analysis, generation and automated recognition of rhythms are proposed and investigated. The techniques are illustrated on the six fundamental 4/4 time *clave* and *bell* rhythm timelines most frequently used in African, Brazilian and Cuban music. It was found that Pressing's measure of rhythm complexity agrees well with the difficulty of performing these clave rhythms whereas the Lempel-Ziv measure appears to be useless. An analysis of the rhythms with several similarity measures reveals that the clave *Son* is most like all the other clave rhythms and perhaps provides an explanation for its worldwide popularity. Finally, a combinatorial technique based on permutations of multisets suggests a fruitful approach to automated generation of new rhythms.

1 Introduction

Imagine a clock as illustrated in Figure 1 which strikes a bell at 16 O'clock and at the 3, 6, 10 and 12 positions for a total of five strikes per clock cycle. The resulting pattern rings out a seductive rhythm which in a short span of fifty years during the last half of the 20th century has managed to conquer the planet. It is known around the world mostly as the *Clave Son* from Cuba. However, it is common in many African rhythms and probably travelled from Africa to Cuba with the slaves. In Cuba it is played with two sticks made of hard wood also called claves [21]. In Africa it is traditionally played with an iron bell. In a section below a mathematical argument is offered to explain the world-wide popularity of this rhythm.

The *Clave Son* rhythm is usually notated for musicians using standard music notation which affords many ways of expressing a rhythm. Four examples are given in the top four lines of Figure 2. The fourth line shows it with music notation using the smallest convenient notes and rests. The bottom line shows a popular way of representing rhythms for percussionists that do not read music. It is called the *Box Notation Method* developed by Philip Harland at the University of California in Los Angeles in 1962 and is also known as TUBS (Time Unit Box System). If we connect the tail to the head of this last diagram and draw it in the form of a circle in clockwise direction we obtain the clock representation in Figure 1, where the squares in Figure 2 filled with black dots correspond to the positions of the bells in Figure 1. The box notation method is convenient for simple-to-notate rhythms like bell and clave patterns as well as for experiments in the psychology of rhythm perception, where a common variant of this method is simply to use one symbol for the strike and another for the pause [6]. Thus for the clave son a common way to write it is simply

*This research was supported by NSERC and FCAR. e-mail: godfried@cs.mcgill.ca

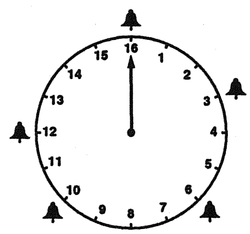

Figure 1: A clock divided into sixteen intervals of time.

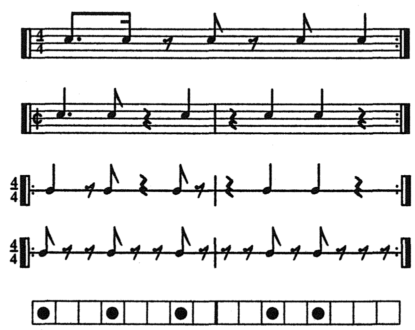

Figure 2: Five ways of representing the *clave Son* rhythm.

as [x . . x . . x . . . x . x . . .]. Finally, in computer science the clave son would be written as the 16-bit binary sequence: [1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0].

Playing these five notes is more difficult than it appears at first hearing. When a Cuban performer passionately invites the audience to participate in clapping this *clave Son*, the non-musicians in the crowd invariably execute their second clap at 4 o'clock instead of 3 o'clock.

There exist literally hundreds of such timeline patterns for bells, claves and woodblocks used in music throughout Africa, Brazil and the Caribbean. This is not surprising when one considers the number of combinations one can create out of five notes played in the sixteen available positions of two bars. Add to that the patterns made with six, seven and up to eleven notes; add to that the patterns that use four bars and in addition the 6/8 time rhythms and we quickly obtain a combinatorial explosion. In this preliminary study however, we are concerned only with the six fundamental five-note 4/4 time clave and bell timelines most frequently used in African, Brazilian and Cuban music. These rhythms are known under many names in different countries but for the purpose of this study I will call them: *Shiko*, *Son*, *Soukous*, *Rumba*, *Bossa-Nova* and *Gahu*. Figure 3 shows all six of them in box notation.

1.1 A geometric representation of rhythms

Consider again the standard musical notation for the clave Son illustrated in the top row in Figure 2. Can the rhythm be played backwards starting at a suitable note so that it sounds exactly the same? Answers to questions such as these are not immediately evident with such a notation. The box notation at the bottom of Figure 2 allows this question to be answered more easily. The answer is yes if we start on the third note. In other words the clave Son is a shifted (or weak) *palindrome*.

An even better representation for such cyclic rhythms is obtained by starting with the clock

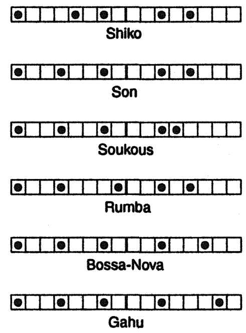

Figure 3: The six fundamental 4/4 time clave and bell patterns in box notation.

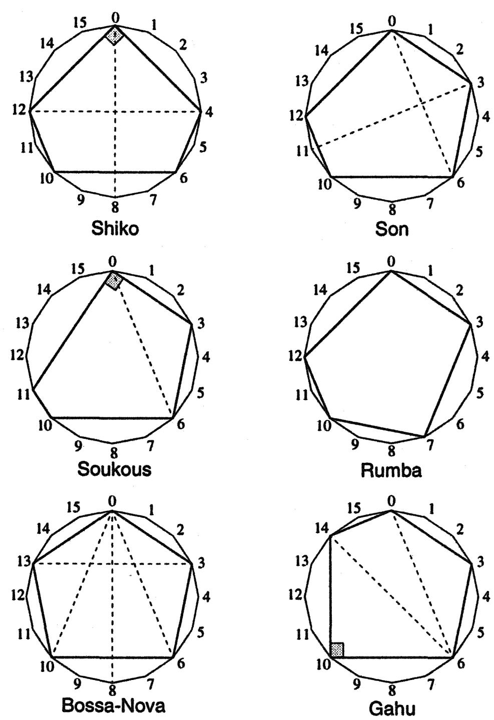

Figure 4: The six fundamental 4/4 time clave and bell patterns represented as convex polygons inscribed in an imaginary circle.

idea of Figure 1 and connecting consecutive note locations to form a convex polygon. Such a representation not only enhances visualization but lends itself more readily to mathematical analysis. It has been used by McLachlan [18] to analyze rhythmic structures from Indonesia and Africa using group theory and Gestalt psychology. The six clave bell patterns are represented as convex polygons in Figure 4 and analysed in more detail in the following section. Note that in Figure 4 the dashed lines indicate either the base of an isoceles triangle or an axis of mirror symmetry.

2 Geometric Analysis of Rhythms

The representation of the six clave and bell patterns described above as convex polygons illustrated in Figure 4 readily uncovers a variety of discriminating geometric properties useful for comparing, analysing and classifying rhythms. Discovering such properties by analyzing classical music notation, or even box-notation, is not self evident. For example, it is imediately obvious from examination of Figure 4 that three timelines, namely Shiko, Soukous and Gahu contain a right interior angle as one of the vertices in their polygons, whereas the other three (Son, Rumba and Bossa-Nova) do not. It is equally irresistible to notice that the former three rhythms evolved in Africa whereas the latter three in America; Bossa-Nova in Brazil and Son and Rumba in Cuba. Does a right angle translate to a stronger beat? Is the presence of the right angle as a discriminating feature between African and American popularity of rhythms more than mere coincidence?

We also see immediately that Shiko and Bossa-Nova are palindromes. They sound the same played forwards or backwards. This can be seen from the mirror symmetry of the polygons about the line through positions (0,8). On the other hand, Son is a *weak palindrome* in that there exists a position other than (0) from which the rhythm sounds the same when played forwards or backwards. In this case the position is (3) since the polygon has mirror symmetry about the line (3,11).

Shiko, Son and Soukous have one isoceles triangle each determined by suitable diagonals. Note that an isoceles triangle indicates two equal consecutive time intervals between notes. Gahu has two isoceles triangles and Bossa-Nova has three. The Bossa-Nova is a *maximally-even* set [3]. A maximally-even set is one in which a subset of the elements has its elements as evenly spaced as possible. The Bossa-Nova has four inter-note intervals of length three and one of length four. In contrast Rumba is the only rhythm with no isoceles triangles, no axis of mirror symmetry and no right angles. Rumba is, at least geometrically speaking, an extraordinary rhythm indeed.

3 Measuring the Complexity of Rhythms

One natural feature useful for a variety of applications including automated recognition of rhythms is *rhythm complexity*. A great deal of attention has been devoted to measuring the objective complexity of sequences in the field of information theory [19], [2]. However, when dealing with rhythm one cannot restrict investigation to consider only objective phenomena. As with visual stimuli, aural stimuli exhibit a variety of perceptual illusions. One of the earliest observations of this kind is concerned with the perception of *beat*. By *beat* is meant one of a series of perceived pulses marking equal units of time. Already in 1894 T. L. Bolton discovered that an isochronous train of identical pulses, such as clock "tics," elicits in the human subject an experience of alternating strong and weak beats, a phenomenon known as *subjective rhythm* [1].

3.1 The Lempel-Ziv complexity

In 1976 Lempel and Ziv [14] proposed an information-theoretic measure of the complexity of a finite sequence in the context of data compression. Their novel approach evaluates the complexity of a finite sequence by scanning the given sequence from left to right looking for the shortest subsequences (words) that have not yet been seen during the scan. Every time such a word is found it forms part of a growing vocabulary. When the scan is finished the size of this vocabulary is the measure of complexity of the sequence. For cyclic sequences such as the rhythms considered here it is sufficient to examine a concatenation of only two instances of the rhythm pattern since no new subsequences will be found by examining longer sequences of the rhythm. As an example consider the clave Son illustrated in Figure 5. The rhythm in binary notation is shown in Figure 5 (a). Repeating it a

Clave Son

(a) 1001001000101000

(b) 10010010001010001001001000101000

(c) 1◆0◆01◆001000◆101◆000100◆1001000101000
1 2 3 4 5 6

Figure 5: Illustrating the computation of the Lempel-Ziv complexity of the clave Son.

Rhythm Complexity Measures

	Lempel-Ziv	Pressing	Metric
Shiko	5	6	2
Son	6	14.5	4
Soukous	6	15	6
Rumba	6	17	5
Bossa-Nova	5	22	6
Gahu	5	19.5	5

Figure 6: A comparison of three measures of rhythm complexity.

second time yields the 32 bit-pattern in Figure 5 (b). Figure 5 (c) shows each new subsequence found by the scan separated by a diamond marker and labelled with an index number underneath. For the clave Son six new subsequences are generated by this process and therefore the Lempel-Ziv complexity is equal to six.

Looking at the scores obtained for the six clave rhythms in Figure 6 (without even comparing to the other measures) shows that this measure is quite bad. There is almost no variance in the scores: all are either 5 or 6. Also the scores do not make sense to anyone experienced in teaching or playing these rhythms. For example Shiko is the simplest of the six rhythms, and Gahu one of the most complex, both to recognize and to play, yet the Lempel-Ziv complexities are 5 for both of these rhythms. It appears that information theoretic measures are not able to capture well the human perceptual, cognitive and performance complexities of rhythms.

3.2 The cognitive complexity of rhythms

Jeff Pressing proposed a measure of the cognitive complexity of a rhythm based on psychological principles and the syncopation present in the rhythm at different levels of pulse [22]. The reader is referred to Pressing [22] for the theory and details on how to measure the cognitive complexity of rhythmic patterns. The cognitive complexities of the ten 4-unit patterns made up of one-note patterns and two-note patterns computed with Pressing's measure are shown in Figure 7. If we take the 16-bit patterns of the clave rhythms, divide them into four units of four bits each, and add the Pressing-complexities of the four corresponding units we obtain values for the Pressing complexities of the six clave rhythms. For example, the Shiko pattern consists of the concatenation of the patterns [1 0 0 0], [1 0 1 0], [0 0 1 0] and [1 0 0 0]. Referring to Figure 7 we obtain the complexities 0, 1, 5 and 0 for a total of 6. On the other hand the Rumba yields a Pressing cognitive

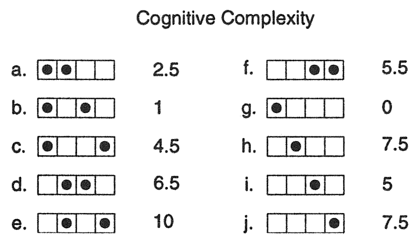

Figure 7: The cognitive complexities of ten basic rhythm patterns according to Pressing [22].

complexity of 4.5 + 7.5 + 5 + 0 = 17. Examining the Pressing cognitive complexities of all six clave rhythms in Figure 6 reveals much more information than the Lempel-Ziv complexities. All the scores are different and the variance is quite large ranging from 6 for the Shiko to 22 for the Bossa-Nova. The scores are also in good agreement with teaching and performing experience. Shiko is easy, Rumba is harder than Son, and Bossa-Nova and Gahu are the hardest of these six rhythms.

3.3 Metricity and metric complexity

Lerdahl and Jackendoff [15] proposed a construct called a *metrical structure* for describing the temporal psychological organization of rhythmic patterns at all metrical levels. Their metrical structure for the 16-time unit interval relevant to the clave and bell patterns considered here is illustrated in Figure 8. The structure defines a function that maps the index of the time unit to the relative strength of its metrical accent. The easiest way to describe the function is by summing levels of beats. At the first level of metrical accent a beat is added at every time-unit starting at unit 0. At the second level a beat is added at every *second* time unit starting at unit 0. At the third level a beat is added at every *fourth* time unit starting at unit 0. We continue in this fashion doubling the time interval between time units at every level. In this way time unit 8 receives 4 beats and finally time unit 0 receives 5 beats. Thus time units 2, 6, 10, and 14 are weak beats, time units 4 and 12 are medium strength beats, unit 8 is stronger and unit 0 is the strongest beat.

A new measure of the complexity of a rhythm can be defined based on the above concept of meter. First define a measure of the total metric strength of a rhythm and call it *metricity*. It is simply the sum of all the metrical accents of the beats present in a rhythm. For example, the clave Son has notes at time units 0, 3, 6, 10 and 12. The metrical accents in Figure 8 corresponding to these time units are 5, 1, 2, 2 and 3, respectively. Therefore the metricity of the clave Son according to the hierarchical arrangement of Lerdahl and Jackendoff [15] is equal to 13. Since this metrical structure has a simple highly structured hierarchy, the metricity function defined here is actually a measure of metric *simplicity*. As such it is inversely proportional to what will be called *metric complexity*. Note that the maximum value of the metricity for five notes cannot exceed 17 and therefore the metric complexity will be defined as 17 minus the metricity. Therefore the metric complexity of the clave Son, for example, is 17 − 13 = 4.

Although the metrical structure defined by Lerdahl and Jackendoff is based on Western or European music it is nevertheless an interesting lens through which to view African and Afro-American rhythms. However, even if this measure has little psychological significance or universal musical merit, at worst it is as valid an objective mathematical measure as the Lempel-Ziv complexity and may find application as a feature extraction method for the automated classification of rhythms. Examination of the metric complexity scores of the six clave rhythms in Figure 6 reveals that this is nevertheless a much better measure than the Lempel-Ziv complexity. The variance is reasonable, Shiko has the lowest score of 2, Rumba (5) has a higher score than Son (4) and Bossa-Nova is

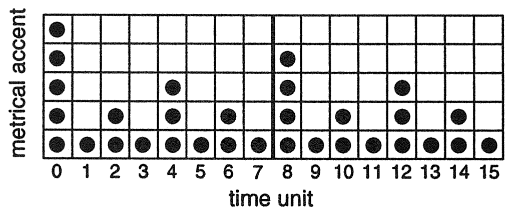

Figure 8: The *metrical structure* of Lerdahl and Jackendoff [15] for a time line pattern of 16 time units.

amongs the most complex with a score of 6. However, the metric complexity measure is not as good as Pressing's cognitive complexity. Bossa-Nova and Gahu are more difficult to play than Soukous, for example, and the metric complexity measure ranks Soukous as harder than Gahu.

4 Measuring the Similarity of Rhythms

At the heart of any algorithm for comparing, recognizing or classifying a rhythm is a measure of the similarity between two rhythms. There exists a wide variety of methods for measuring the similarity of two rhythms represented by a string of symbols. Indeed the resulting approximate pattern matching problem is a classical problem in pattern recognition and computer science in general [5]. When the two strings are binary sequences a natural measure of distance or non-similarity between them is the Hamming distance [11] widely used in coding theory. The Hamming distance is the number of places in the strings that do not match. The Hamming distance is not appropriate for our problem of rhythm similarity because although it measures a missmatch, it does not measure how far the missmatch occurs and if a note is moved a large distance it will sound more different than if it is moved a small distance. Some rhythm detection algorithms [20] and systems for machine recognition of music patterns [4] use inter-onset intervals as a basis for measuring similarity. These are the intervals of time between consecutive note onsets in a rhythm. Coyle and Shmulevich [4] represent a music pattern by what they call a *difference-of-rhythm vector.* If $T = (t_1, t_2, ..., t_n)$ is a vector of inter-onset time intervals for the notes of a rhythm then they define the difference-of-rhythm vector as $X = (x_1, x_2, ..., x_{n-1})$, where $x_i = t_{i+1}/t_i$. This approach is more appropriate than the Hamming distance for the rhythms considered here. In the next subsection the six clave rhythms are compared with respect to a feature vector defined by successive inter-onset intervals in a slightly different way.

4.1 The interval vector distance

Consider the representation of the six clave rhythms as convex polygons in Figure 4. These polygons immediately suggest a variety of possible shape feature vectors for characterizing the rhythms based on measuring angles of vertices or lengths of edges. Alternately one could measure global shape features of the polygon [25].

Here each rhythm is represented by a vector of five numbers that characterize these five intervals. More specifically a rhythm is represented by $X = (x_1, x_2, x_3, x_4, x_5)$, where x_i is the number of vertices skipped by the ith polygon edge starting at the vertex labelled 0. This is essentially the same as the sequence of inter-onset time intervals since the time interval is the number of vertices skipped plus one. The dissimilarity between two rhythms X and Y is measured by the Euclidean distance between the two vectors X and Y in this 5-dimensional vector space. The distance matrix

Interval Vector Distance Matrix

	Shiko	Son	Soukous	Rumba	Bossa	Gahu
Shiko	0	1.41	2	2.45	2	3.16
Son		0	1.41	1.41	1.41	2.83
Soukous			0	2	2.83	4.24
Rumba				0	2	3.16
Bossa-Nova					0	1.41
Gahu						0
Σ	11.02	8.47	12.48	11.02	9.65	14.80

Figure 9: The distance matrix of the interval vectors with the Euclidean metric. The bottom row indicates for each rhythm the sum of the distances it is from the other five.

Minimum Spanning Tree

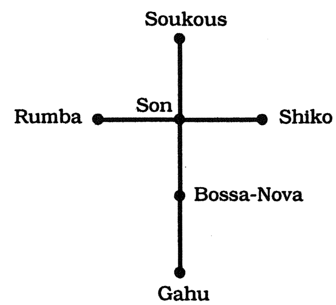

Figure 10: The minimum spanning tree determined by the distance matrix of interval vectors.

based on these vectors is shown in Figure 9. The bottom row in Figure 9 shows for each rhythm the sum of its distances to all the other rhythms. This is a measure of how dissimilar a rhythm is from the rest of the group. Most noteworthy are the highest (14.80) and lowest (8.47) values for Gahu and Son, respectively. David Locke [16] has argued forcefully, from the music theory point of view, the uniqueness of the Gahu bell pattern. The results obtained here provide mathematical confirmation of Locke's musical analysis. They also provide mathematical evidence that the clave Son is most like all the others. This may explain the world-wide popularity of the Son.

4.2 The minimum spanning tree

The minimum spanning tree is a powerful and useful visualization tool for understanding the structure of data in higher dimensional spaces such as that encountered here [26]. For this reason it has been used with great success in cluster analysis [5]. The distance matrix of Figure 9 defines a complete weighted graph G. The six rhythms correspond to the six nodes of the graph. The graph is complete because every pair of nodes is connected by an edge. The weight on each edge connecting two nodes is the distance between the corresponding two rhythms. The minimum spanning tree (MST) of G is the subgraph of G that is a tree (has no cycles), connects (spans) all six nodes, and has minimum weight among all such trees. Here the weight of the tree is just the sum of the weights of all its edges. Note that in general the minimum spanning tree may not be unique. There exist

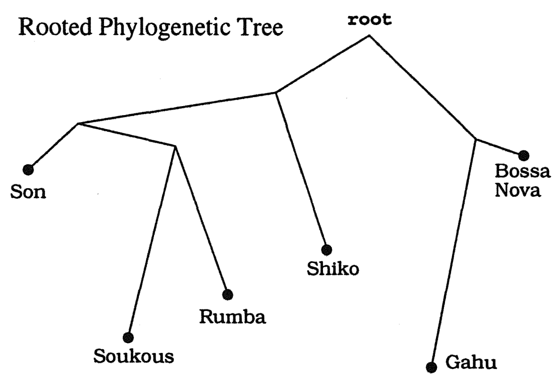

Figure 11: The rooted phylogenetic tree constructed for the distance matrix in Figure 9.

many algorithms for computing the minimum spanning tree [8]. The simplest from a conceptual point of view is Kruskal's algorithm. This algorithm first sorts all pairs of nodes (edges) by increasing weight (distance) and then scans through this sorted list picking edges to add to a growing tree as long as no cycles are created. Proceeding in this way with the distance matrix in Figure 9 quickly results in the minimum spanning tree shown in Figure 10. The topology of the minimum spanning tree in Figure 10 provides at a glance a good idea of how the smallest distances (1.41) in the distance matrix relate to one another qualitatively. Here one can visualize instantly that Gahu is most unlike the remaining rhythms whereas Son is most similar to the other five rhythms. In fact Son is the center of gravity of the tree in the sense that it is the node that minimizes the sum of distances (within the tree) to all the other rhythms.

4.3 Phylogenetic trees

The drawing of the minimum spanning tree in Figure 10 does not attempt to quantitatively visualize *all* the distances in the distance matrix of Figure 9. For this purpose there exist more powerful (and difficult to compute) tools such as the *phylogenetic trees* of Gaston Gonnet [7]. For details on how to construct such trees see [7]. The Computational Biochemistry Research Group of the Swiss Federal Institute of Technology (ETH) offers a web service for computing phylogenetic trees from distance matrices submitted to their server. Although designed for applications to gene sequence analysis in molecular biology, phylogenetic trees can be constructed for any distance matrix such as the ones obtained from sequences of notes in rhythms. Such trees serve to describe the clustering relationships between objects much like the more traditional hierarchical cluster analysis techniques used in data analysis [5], [13].

One type of phylogenetic tree is the "vertically" oriented rooted tree. Such a tree constructed from the distance matrix of Figure 9 is shown in Figure 11. In this tree the minimum vertical distance (vertical component only) travelled in traversing the tree between two nodes (up and down) is proportional to the distance between them in the distance matrix. For example, the distance between Gahu and Soukous is 4.24. In the rooted phylogenetic tree this distance is proportional to the distance travelled upwards from the Gahu node to the root (vertical component only) plus the distance travelled downwards (vertical component only) from the root to the Soukous node. The rooted phylogenetic tree shows the hierarchical clustering that exists between all the rhythms. The first partition is in the two fundamental clusters: Bossa-Nova and Gahu in the right cluster and the other four in the left cluster. The left cluster then breaks up into Shiko and the other three:

Figure 12: The sequence used by Steve Reich in *Clapping Music* [24].

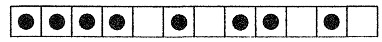

Figure 13: The other sequence found by Joel Haak [9].

Son, Soukous and Rumba. Finally, Son breaks away from the pair Soukous and Rumba. Here the phylogenetic trees have been used for visualization and clustering purposes only. However, in future work such trees will be used to determine "ancestral" rhythms.

5 Combinatorial Analysis of Rhythms

The application of combinatorics to the analysis of music is not new. However, almost all such analyses have been applied to the "vertical" tone scale rather than the "horizontal" time scale [23], [3]. Exceptions to this trend are the two papers by Haak [9], [10] and the one by London [17].

Steve Reich [24] composed an interesting piece called *Clapping Music* for two people clapping hands. Both performers clap exactly the same 12-time-unit rhythm shown in Figure 12. One performer repeats the sequence continually throughout the piece but the second performer, after having repeated the sequence 12 times, shifts the sequence by one beat. The second player continues to shift the pattern by one time unit in the same direction every time the pattern has been played 12 times. The piece finishes when the second performer returns in phase with the first. Thus the last 12 combined patterns sound the same as the first 12 when both performers are perfectly in phase.

There are 8 notes (claps) in Steve reich's *Clapping Music.* The number of ways we can select 8 out of 12 time units in which to clap is $(12!)/(8!)(4!) = 495$. Joel Haak [9] raises the interesting question of how Reich might have come to select the pattern in Figure 12 out of all the possible 495. Did he listen to all 495 and select the pattern he liked best? Haak then suggests a mathematical response to the question. If we take into consideration that a piece should begin with a clap and not a silent pause, and if cyclic permutations of a pattern are considered equivalent, and if during the execution of the entire piece the combined pattern made by both performers clapping does not repeat itself, and finally, if we do not allow consecutive repetitions of the number of claps between two consecutive pauses, then only 2 of the 495 patterns satisfy these constraints. One is the [3,2,1,2] note pattern Reich chose in Figure 12. The other is the [4,1,2,1] note pattern of Figure 13. Haak does not offer a criterion for choosing between the last 2 remaining candidates. But of course if we want to minimize the maximum range of the lengths of consecutive claps, then we obtain Reich's pattern. Alternately, we may invoke the criterion of maximally-even sets [3] to arrive at Reich's pattern.

5.1 Permutations and multisets

In the clave rhythm patterns considered in this paper five notes are played in sixteen available time units. The number of ways to select 5 out of 16 is $(16!)/(5!)(11!) = 4368$. This is a large number of patterns. Furthermore most of these may be useless as good time-line patterns for powerful percussive dance music. How can we reduce this large number to an interesting small subset? Note that in the five clave patterns other than Soukous, the minimum and maximum inter-onset intervals are 2 and 4, respectively. Therefore we can modify the combinatorial question to take such constraints into account. Consider the clave Son pattern in binary sequence representation [1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0] and its interval-vector (2 2 3 1 3) corresponding to the time units skipped (zero's) in between the notes played (one's). One may ask how many permutations exist of the pattern (2 2 3 1 3)? Note that these are *multisets* now since repetitions of the elements are permitted [12]. We have 5 objects of three different classes: 1 of class one, 2 of class two and 2 of class three. Therefore the total number of different permutations of (2 2 3 1 3) is $(5!)/(1!)(2!)(2!) = 30$.

The reader is invited to play these. It turns out all 30 of these permutations sound great. Among these 30 are also found the Rumba, the Gahu as well as the backward versions of the Son, Rumba and Gahu. This also becomes evident from examining Figure 4. If one rhythm may be obtained from another by a permutation of its interval vector the two rhythms will be said to belong to the same *interval combinatorial class.* Thus Son (2 2 3 1 3), Rumba (2 3 2 1 3) and Gahu (2 2 3 3 1) belong to the same interval combinatorial class, whereas Shiko (3 1 3 1 3), Soukous (2 2 3 0 4) and Bossa-Nova (2 2 3 2 2) each belong to their own distinctive classes.

Returning to the 30 rhythms of the Son-Rumba-Gahu interval combinatorial class, and excepting the Son played backwards because it is a weak palindrome, the remaining 26 rhythms have an eerie resemblance to the Son, Rumba and Gahu but sound more modern, more jazzy somehow. Any one of them could be successfully incorporated in new music. This interval combinatorial technique suggests a fruitful approach to automated generation of new rhythms.

References

[1] T. L. Bolton. Rhythm. *American Journal of Psychology*, 6:145–238, 1894.

[2] Gregory J. Chaitin. Information theoretic computational complexity. *IEEE Transactions on Information Theory*, IT-20:10–15, 1974.

[3] J. Clough and J. Douthett. Maximally even sets. *Journal of Music Theory*, 35:93–173, 1991.

[4] E. J. Coyle and I. Shmulevich. A system for machine recognition of music patterns. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, Washington, 1998.

[5] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley, New York, 1973.

[6] Douglas Eck. A positive-evidence model for classifying rhythmical patterns. Technical Report IDSIA-09-00, Instituto Dalle Molle di studi sull'intelligenza artificiale, Manno, Switzerland, 2000.

[7] Gaston H. Gonnet. New algorithms for the computation of evolutionary phylogenetic trees. In S. Suhai, editor, *Computational Methods in Genome Research.* Plenum Press, 1994.

[8] Michael T. Goodrich and Roberto Tamasia. *Algorithm Design.* John Wiley and Sons, Inc., New York, 2002.

[9] Joel K. Haak. Clapping Music - a combinatorial problem. *The College Mathematics Journal*, 22:224–227, May 1991.

[10] Joel K. Haak. The mathematics of Steve Reich's *Clapping Music*. In *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, pages 87–92, Winfield, Kansas, 1998.

[11] R. W. Hamming. *Coding and Information Theory*. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[12] T. C. Hu and B. N. Tien. Generating permutations with non-distinct items. *American Mathematical Monthly*, 83(8):193–196, October 1976.

[13] J. W. Jaromczyk and Godfried T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9):1502–1517, 1992.

[14] A. Lempel and J. Ziv. On the complexity of finite sequences. *IEEE Transactions on Information Theory*, IT-22(1):75–81, 1976.

[15] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, Massachussetts, 1983.

[16] David Locke. *Drum Gahu: An Introduction to African Rhythm*. White Cliffs Media, Gilsum, New Hampshire, 1998.

[17] Justin London. Hierarchical representations of complex meters. In *6th International Conference on Music, Perception and Cognition*, Keele University, United Kingdom, August 5-10 2000.

[18] Neil McLachlan. A spatial theory of rhythmic resolution. *Leonardo*, 10:61–67, 2000.

[19] Abraham Moles. *Information Theory and Esthetic Perception*. The University of Illinois Press, Urbana and London, 1966.

[20] Bernard Mont-Reynaud and Malcolm Goldstein. On finding rhythmic patterns in musical lines. In *Proceedings of the International Computer Music Conference*, pages 391–397, San Francisco, California, 1985.

[21] Fernando Ortiz. *La Clave*. Editorial Letras Cubanas, La Habana, Cuba, 1995.

[22] Jeff Pressing. Cognitive complexity and the structure of musical patterns. In *Proceedings of the 4th Conference of the Australasian Cognitive Science Society*, Newcastle, Australia, 1997. [CD-ROM].

[23] Ronald C. Read. Combinatorial problems in the theory of music. *Discrete Mathematics*, 167-168, April 1997.

[24] Steve Reich. *Writings about Music*. The Press of the Nova Scotia College of Art and Design, Halifax, Canada, 1974.

[25] Godfried T. Toussaint. Generalizations of π: some applications. *The Mathematical Gazette*, pages 291–293, December 1974.

[26] Charles T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, C-20:68–86, 1971.