An Intervallic Approach to Tone Rows: *SerialGen*, a Twelve-Tone Row Generator

Marco Feitosa
marco.feitosa@yahoo.com

Raphael Sousa Santos
contact@raphaelss.com

Luan Simões Cardoso
luansimoes0@gmail.com

Abstract: In this paper, we propose an intervallic approach to tone rows, making use of mathematical concepts from the theory of partitions and original musical ones, also presenting a new software called *SerialGen*, which generates tone rows and row classes based on interval compositions. At the end, we present preliminary results in dialogue with important works in the specific area, which address group theory and twelve-tone music.
Keywords: Post-Tonal Theory. Serialism. Twelve-Tone Music. Theory of Partitions. SerialGen.

1. Introduction

Serial music or, more specifically, dodecaphonic serialism plays an important role in twentieth-century music. Developed by Arnold Schoenberg (1874-1951) from early 1920s onwards, the method of composing with twelve tones is a compositional procedure based on his theory of "the emancipation of the dissonance" (SCHOENEBRG, 1950 and 1969), where dissonances are considered merely more remote consonances in the series of overtones, so that it does not aim at the establishment of a tonality (yet does not exclude it entirely), deriving a total musical structure from a complex of pitch classes not functionally differentiated.

The basis of the twelve-tone technique is what Schoenberg used to refer as the *basic set* (BS) and today is often called a *tone row* or *series*,[1] which consists of an ordered arrangement or succession, i.e., a permutation without repetition of the twelve pitch classes from the equal-tempered chromatic scale to be used as basic (structural) material in a musical composition.

As a general rule, the pitch classes within a tone row must be used according to their fixed order. Roughly speaking, one cannot be repeated until all others are played. In turn, a tone row may be combined with its various transformations which include the transpositions of its *original* or *prime form* (denoted O or P)[2] and its derivatives — *inversion* (I), *retrograde* (R) and *retrograde inversion* (RI). As Schoenberg points out, all of these possibilities may appear in a wide variety of ways within a composition:

[1] Other names are *pitch row*, *note row*, or simply *row*, also *(ordered) set*, *sequence*, and so forth.
[2] We will adopt O.

FEITOSA, Marco; SANTOS, Raphael; CARDOSO, Luan. An Intervallic Approach to Tone Rows: SerialGen, a Twelve-Tone Row Generator. Congresso Internacional de Música e Matemática, 6. Rio de Janeiro, 2021. *Anais*... Rio de Janeiro: UFRJ, 2022, p. 132-143.

> For the sake of a more profound logic, the Method of Composing with Twelve Tones derives all configurations from a basic set (Grundgestalt) [tone row]. The order in this basic set and its three derivatives — contrary motion [inversion], retrograde, and retrograde inversion respectively — is, like the motive, obligatory for a whole piece. Deviation from this order of tones should normally not occur, in contrast to the treatment of the motive, where variation is indispensable. Nevertheless, variety is not precluded. The tones in the right order may appear either successively in a melody, theme or independent voice, or as an accompaniment consisting of simultaneous sounds (like harmonies). (SCHOENBERG, 1969, p. 193-194).

For Schoenberg (1950, p. 103-108), "composition with twelve tones has no other aim than comprehensibility." To ensure that, the tone row "functions in the manner of a motive," and this explains why it has to be "invented anew for every piece." For him, it has to be "the first creative thought" of a twelve-tone composition. In this sense, a tone row is not merely another type of pre-compositional material, but a compositional (musical) idea in itself, and its construction, more than just a previous stage of the compositional process, is instead already a fundamental part of it.

Building a tone row is much more than randomly picking a sequence of pitch classes (although it may also be just that). It is actually a creative decision-making process with countless musical implications, and almost all of these basically stem from the choice of the musical intervals within the row. In a nutshell, the "sonic identity" or "profile" of a tone row depends directly on its interval structure:

> Each row's "sound" is determined not so much by its sequence of pcs [pitch classes] (unless one has absolute pitch) as by the ordered intervals between its successive pcs. […] Rows can be created to maximize certain intervals, to omit others or to provide as much diversity as possible — as in all-interval rows. (MORRIS, 2015, p. 182).

As we can notice, intervals — even more than the pitch classes themselves — are essential for the construction of a tone row and its sonic identity. But how can one use intervals to build a row? Rather, how can one build a row exclusively from intervals? At first glance, that seems like an easy question. However, as we will show, the answer is not as simple and trivial as one might think.

Thus, bearing that question in mind and endeavouring to answer it, we propose in this paper an intervallic approach to tone rows, making use of mathematical concepts from the theory of (integer) partitions and original musical ones, also presenting a new software tool in Python language called *SerialGen* (still in development), which generates tone rows and row classes based solely on intervals and interval structures, or more precisely, on *interval compositions*, as will be defined.[3] At

[3] The theoretical bases of the concepts and definitions presented here are found in FEITOSA (2020).

the end, we present preliminary results in dialogue with important works in the specific area, which address group theory and twelve-tone music.

2. Concepts and Definitions

Before introducing new musical concepts and definitions, let us take a look at some basic mathematical concepts. According to Andrews (1994, p. 149), "the theory of partitions is an area of additive number theory, a subject concerning the representation of integers as sums of other integers". Briefly, a *partition* is "a way of splitting a number into integer parts" (ANDREWS; ERIKSSON, 2004, p. 3). Let us consider the following:

$$4 = 3 + 1 = 2 + 2 = 2 + 1 + 1 = 1 + 1 + 1 + 1.$$

As we see, there are 5 partitions of the number 4, since there are 5 ways of splitting it into integer parts. Thus, a *partition* of a positive integer s is a representation of s as a sum of positive integers, called *summands* or *parts* of the partition, the order of which is irrelevant. Since order is irrelevant, we shall henceforth write partitions with non-increasing order of parts. In this sense (ANDREWS, 1998, p. 1), a partition of a positive integer s is a finite non-increasing sequence of positive integers $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$, such that $\sum_{i=1}^{k} \lambda_i = s$. Each λ_i corresponds to a part of the partition and if λ is a partition of s, we write $\lambda \vdash s$. The number of partitions of s is denoted by $p(s)$, therefore $p(4) = 5$.

By definition, the order of the parts of a partition is irrelevant, which technically characterizes an *unordered partition*. However, when the order of the parts is considered, then we have what is called an *ordered partition* or, simply, a *composition*. Hence, a composition of s can be thought of as an expression of s as an ordered sum of integers. Formally, a composition of s is a finite (ordered) sequence of positive integers $\alpha = \langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle$,[4] such that $\sum_{i=1}^{k} \alpha_i = s$. For example, for $s = 4$, besides the 5 partitions mentioned before: (4), $(3,1)$, $(2,2)$, $(2,1,1)$, $(1,1,1,1)$; there are 8 distinct compositions: $\langle 4 \rangle$, $\langle 3,1 \rangle$, $\langle 1,3 \rangle$, $\langle 2,2 \rangle$, $\langle 2,1,1 \rangle$, $\langle 1,2,1 \rangle$, $\langle 1,1,2 \rangle$, $\langle 1,1,1,1 \rangle$. Again, each α_i corresponds to a part of the composition and the number of compositions of s is denoted by $c(s)$, in the present case, $c(4) = 8$.

Now that we are already familiar with the elementary concepts of partition and composition, let us take as a starting point the tone row B♭, A, C♯, B, F, G, E, C, A♭, E♭, F♯, D, from Schoenberg's *Fantasy for Violin and Piano*, Op. 47 (1949),[5] presented in both musical and integer notation[6] (Figure 1).

[4] We will differentiate partitions from compositions by using round and angle brackets, respectively.

[5] For an analysis of that piece, see TIPTON (2017).

[6] For details on integer notation, see STRAUS (2016, p. 5-6).

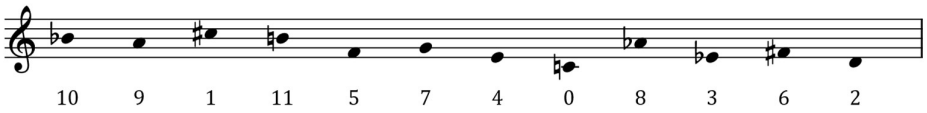


Figure 1: Tone row from Schoenberg's *Fantasy for Violin and Piano*, Op. 47 (1949).

By replacing each pitch class with its corresponding numerical label, the row may be represented as the 12-tuple $(10, 9, 1, 11, 5, 7, 4, 0, 8, 3, 6, 2)$. In addition, since we are dealing with pitch classes within a modular space, an alternative representation of that row is as a *clock diagram*, i.e., a directed graph in the form of a regular dodecagon whose vertices correspond to the respective pitch classes being successively connected by directed edges (Figure 2).

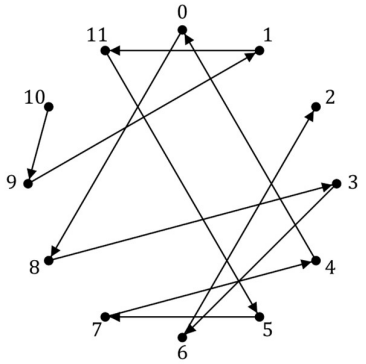
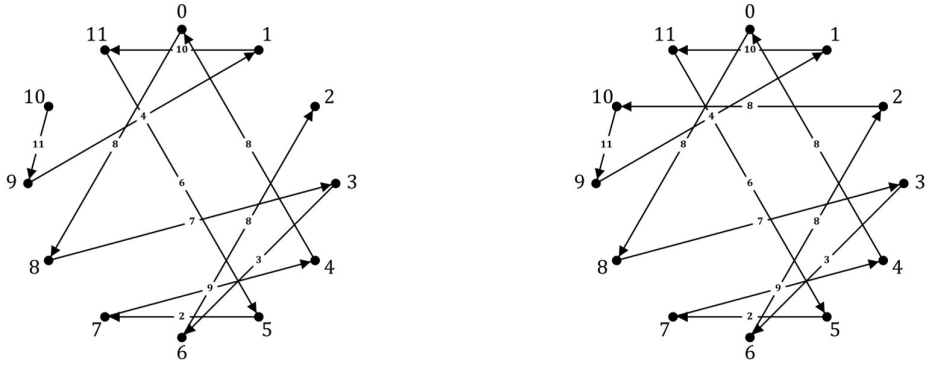


Figure 2: Clock diagram of the tone row $\mathbf{(10, 9, 1, 11, 5, 7, 4, 0, 8, 3, 6, 2)}$.

Insofar as the directed edges may be associated with the directed intervals between the successive pitch classes (ordered pitch-class intervals[7]), we can also represent the row in terms of those intervals only. For that, we need to compute the number of clockwise steps necessary to get from one pitch class to the next, starting from the first to the second, then from the second to the third, and so on, obtaining the interval succession $11, 4, 10, 6, 2, 9, 8, 8, 7, 3, 8$ (Figure 3a). However, since the row is within a modular space, we should think of it cyclically and then, in order to derive its complete interval structure, we should also include the interval between the last and the first pitch class, getting the sequence $11, 4, 10, 6, 2, 9, 8, 8, 7, 3, 8, 8$ (Figure 3b).

[7] For details on ordered pitch-class intervals, see STRAUS (2016, p. 9-11).



a) Intervals: $11, 4, 10, 6, 2, 9, 8, 8, 7, 3, 8$ **b)** Intervals: $11, 4, 10, 6, 2, 9, 8, 8, 7, 3, 8, 8$

Figure 3: Clock diagrams with directed intervals.

Mathematically, the directed interval from pitch class x to pitch class y is given by $(y - x)$ mod 12. The modular arithmetic gives us a more practical method for calculating directed intervals between pitch classes than counting clockwise steps. Hence, when we go from pitch class 10 to pitch class 9, we have the directed interval $(9 - 10)$ mod $12 = -1$ mod $12 = 11$. By repeating this process for all successive pitch classes within the row (last and first inclusive), we will obtain the same sequence given in Figure 3b. If we finally associate those directed intervals with the parts of a partition, or rather, with the parts of a composition, and sum them up, we will get:

$$11 + 4 + 10 + 6 + 2 + 9 + 8 + 8 + 7 + 3 + 8 + 8 = 84.$$

From this we may verify that the present sequence of directed intervals is actually a composition of 84. Thus, by generalizing all the reasoning done so far, we finally arrive at the following definition.

Definition 2.1 (Interval Composition). Let $\mathbb{Z}_n = \{0, 1, 2, \dots, n-1\}$, $n \in \mathbb{Z}^+$, be the set of n pitch classes labeled successively from 0 to $n - 1$, an *interval composition* α is generically defined by:

$$\alpha = \langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle, \quad \text{such that} \sum_{i=1}^{k} \alpha_i = n \cdot d, \tag{1}$$

where $\alpha_i \in \mathbb{Z}^+$ is called a *part* of α; $k \in \mathbb{Z}^+$, $k \leq n$, is the *length* of α; $d \in \mathbb{Z}^+$ is the *dimensional factor* of α; and $s = n \cdot d$ is the *span* of α. Then, given a tone row $(p_1, p_2, \dots, p_k)$, $p_i \in \mathbb{Z}_n$, its corresponding interval composition α is defined by:

$$\alpha = \left\langle \left(p_{(i+1) \bmod k} - p_{i \bmod k}\right) \bmod n \right\rangle_{i=1}^{k}, \quad \text{where } p_0 = p_k. \tag{2}$$

For our present purposes, since we are dealing with pitch classes and twelve-tone rows within a modular space derived from twelve-tone equal temperament, it is clear that $n = 12$ and $k = 12$ always. Consequently, we may infer that $\alpha_i \leq 11$, which implies that $\sum_{i=1}^{12} \alpha_i \leq 12 \cdot 11 = 132$ (i.e., $s = n \cdot d \leq 132$) and, therefore, $d \leq 11$.

Now, returning to our particular example, the corresponding interval composition of the row $(10, 9, 1, 11, 5, 7, 4, 0, 8, 3, 6, 2)$ is $\langle 11, 4, 10, 6, 2, 9, 8, 8, 7, 3, 8, 8\rangle$. In this case, as we mentioned before, the span $s = 84 = 12 \cdot 7$, so the dimensional factor $d = 7$. We may alternatively say that such row has *dimension* 7. It is worth mentioning that the dimensional factor corresponds to the number of overlapping clockwise turns around the clock diagram, which musically represents the "number of octaves" that the row theoretically comprises (thinking of its pitch realization only in ascending direction). Thus, in an abstract sense, the dimensional factor is a measure of *compression* or *dispersion* of a row in terms of its directed intervals.

As we can observe so far, an interval composition indicates only the interval structure of the row, not providing any information about its pitch classes. For that, it is necessary that we associate the *pitch* operator P_x, $x \in \mathbb{Z}_n$, with the interval composition, obtaining a *pitched interval composition* $P_x\alpha$, where the *pitch index* x is the first pitch class of the corresponding row. For the mentioned example, we have: $(10, 9, 1, 11, 5, 7, 4, 0, 8, 3, 6, 2) = P_{10}\langle 11, 4, 10, 6, 2, 9, 8, 8, 7, 3, 8, 8\rangle$. Conversely, given a pitched interval composition $P_x\langle \alpha_1, \alpha_2, \dots, \alpha_k\rangle$, $\alpha_i \in \mathbb{Z}^+$, we can obtain its corresponding tone row $(p_1, p_2, \dots, p_k)$, $p_i \in \mathbb{Z}_n$, as follows:

$$P_x\langle \alpha_1, \alpha_2, \dots, \alpha_k\rangle = \left(\left(x + \sum_{i=1}^{j} \alpha_{i-1} \right) \bmod n \right)_{j=1}^{k}, \tag{3}$$

where $\alpha = \langle \alpha_1, \alpha_2, \dots, \alpha_k\rangle$ is a *restricted interval composition* whose parts satisfy

$$\left(\sum_{i}^{j} \alpha_i \right) \bmod n \neq 0, \quad \text{for every } i, j \in \mathbb{Z}^+, \text{such that } i < j \leq k - 1. \tag{4}$$

Simply put, this last condition means that the sum of the successive (contiguous) parts of α, taken two by two, three by three, and so on, from the first up to the penultimate part, should not be a multiple of n, otherwise we would get repeated pitch classes. And again, since we are dealing with twelve-tone rows, $n = 12$ and $k = 12$.

At this point, we can already answer our motivation question — how can one build a row exclusively from intervals? And the answer is basically — through a restricted interval composition whose parts satisfy Equation 4. However, that is not an easy task to perform manually[8] and here computer assistance is welcome. But before we present our software, let us delve a little deeper into some properties of interval compositions and introduce other useful concepts and definitions.

[8] It would be necessary to check $\binom{k-1}{2}$ successive sums manually. For $k = 12$, then the total is 55.

First, let us consider the *twelve-tone matrix*[9] for our sample row (Figure 4). We notice that there are 48 forms of the row, i.e., 12 transpositions for each of its 4 basic forms — original (O), inversion (I), retrograde (R), and retrograde inversion (RI).

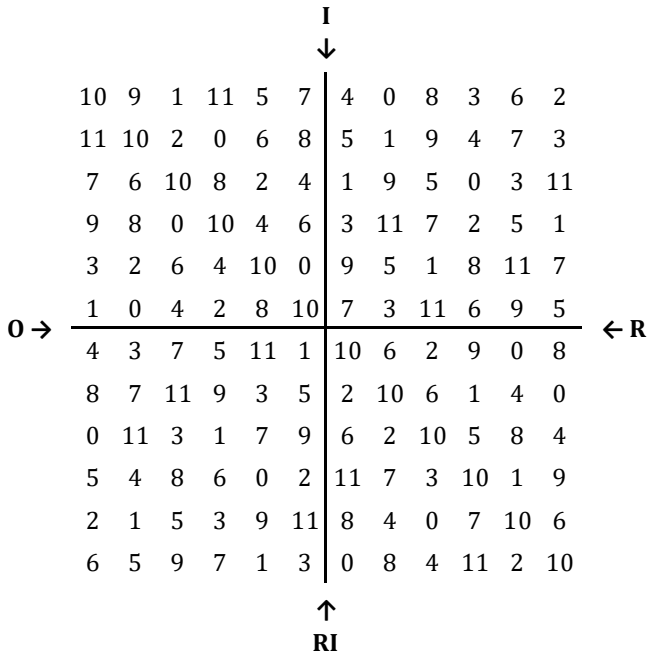


Figure 4: Twelve-tone matrix for $\mathbf{(10, 9, 1, 11, 5, 7, 4, 0, 8, 3, 6, 2)}$.

Since all transpositions of those basic forms have the same interval composition, we may conceive them as equivalence classes of rows related by transposition and represent them by their corresponding interval compositions, then we have: $\langle 11, 4, 10, 6, 2, 9, 8, 8, 7, 3, 8, 8\rangle$ for O, $\langle 1, 8, 2, 6, 10, 3, 4, 4, 5, 9, 4, 4\rangle$ for I, $\langle 4, 9, 5, 4, 4, 3, 10, 6, 2, 8, 1, 4\rangle$ for R, and $\langle 8, 3, 7, 8, 8, 9, 2, 6, 10, 4, 11, 8\rangle$ for RI. According to Schoenberg's method, tone rows are equivalent if they can be determined by transposition, inversion, and/or retrograde from a single tone row. In this sense, those four interval compositions satisfy the Schoenbergian notion of equivalence insofar as they comprise all the rows related by such operations. Finally, we may define those basic operations and more elaborate ones, such as *cyclic shift* (S), *multiplication* (M), and *five-step permutation* (F), in terms of interval compositions only, in order to obtain different equivalence classes, or rather, *row classes* from their various combinations, as follows.

DEFINITION 2.2 (OPERATIONS). Let $\alpha = \langle \alpha_1, \alpha_2, \dots, \alpha_k\rangle$, $\alpha_i \in \mathbb{Z}^+$, be an interval composition, the operations of *inversion* (I), *retrograde* (R), *retrograde inversion* (RI), *cyclic shift* (S), *multiplication* (M), and *five-step permutation* (F) are respectively defined by:

[9] For details on twelve-tone matrices, see STRAUS (2016, p. 301-302).

$$I\langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle = \langle (-\alpha_i) \bmod n \rangle_{i=1}^{k}. \tag{5}$$

$$R\langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle = \left\langle \left(-\alpha_{(k-i) \bmod n}\right) \bmod n \right\rangle_{i=1}^{k}, \quad \text{where } \alpha_0 = \alpha_k. \tag{6}$$

$$RI\langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle = \left\langle \alpha_{(k-i) \bmod n} \right\rangle_{i=1}^{k}, \quad \text{where } \alpha_0 = \alpha_k. \tag{7}$$

$$S\langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle = \left\langle \alpha_{(i+1) \bmod n} \right\rangle_{i=1}^{k}, \quad \text{where } \alpha_0 = \alpha_k. \tag{8}$$

$$M\langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle = \langle (5 \cdot \alpha_i) \bmod n \rangle_{i=1}^{k}. \tag{9}$$

$$F\langle \alpha_1, \alpha_2, \dots, \alpha_k \rangle = \left\langle \sum_{i=1}^{5} \alpha_{(i+5 \cdot j) \bmod n} \right\rangle_{j=0}^{k-1}, \quad \text{where } \alpha_0 = \alpha_k. \tag{10}$$

Therefore, taking our example:

$\alpha = \langle 11, 4, 10, 6, 2, 9, 8, 8, 7, 3, 8, 8 \rangle$, $s = 84$ and $d = 7$;
$I\alpha = \langle 1, 8, 2, 6, 10, 3, 4, 4, 5, 9, 4, 4 \rangle$, $s = 60$ and $d = 5$;
$R\alpha = \langle 4, 9, 5, 4, 4, 3, 10, 6, 2, 8, 1, 4 \rangle$, $s = 60$ and $d = 5$;
$RI\alpha = \langle 8, 3, 7, 8, 8, 9, 2, 6, 10, 4, 11, 8 \rangle$, $s = 84$ and $d = 7$;
$S\alpha = \langle 4, 10, 6, 2, 9, 8, 8, 7, 3, 8, 8, 11 \rangle$, $s = 84$ and $d = 7$;
$M\alpha = \langle 7, 8, 2, 6, 10, 9, 4, 4, 11, 3, 4, 4 \rangle$, $s = 72$ and $d = 6$;
$F\alpha = \langle 9, 11, 5, 9, 1, 7, 10, 3, 10, 10, 11, 10 \rangle$, $s = 96$ and $d = 8$.

Here we verify that the dimensional factor is the same for the original interval composition α and $RI\alpha$ (also for $S\alpha$), as well as for $I\alpha$ and $R\alpha$, being different for $M\alpha$ and $F\alpha$. This means that $I\alpha$ and $R\alpha$ are the "most compact" in terms of directed intervals ($d = 5$) and, between them, $I\alpha$ would be the class representative (*normal form*[10]) of the row classes comprising such operations, since it is even more compact.

Considering all the theoretical scope provided so far, we can then describe how these concepts and definitions were used in our software.

3. *SerialGen*

The *SerialGen* software[11] was built in two steps. The first one generated the restricted interval compositions whose parts satisfy Equation 4 in the form of a database. The second one aimed to partition that database into equivalence classes determined by the combination of operations (Definition 2.2) and filter those interval compositions with user-defined parameters, such as intervallic constraints.

First, we must explain the *restricted_compositions* function, that returns a Python *Generator* object listing all interval compositions restricted by Equation 4,

[10] The *normal form* is the interval composition with the smallest d and greatest α_k, α_{k-1}, and so on.
[11] The source code of the software is available at: https://github.com/marco-feitosa/serialgen.

given the positive integers s, k and n as parameters. This algorithm was used to populate the database with the restricted interval compositions for every dimensional factor $d \in \mathbb{Z}^+$, such that $d \leq 11$.

The main idea of the algorithm comes from the perception that it is possible to reach every valid interval composition by recursively assigning each eligible value to each part, checking that the current composition satisfies the constraints at each recursive call. The recursion itself is implemented in fact by the *compute* function, which is defined within the scope of *restricted_compositions* and is where the compositions are actually generated, while the only role of the outer function is to call *compute* with the initial values. In addition, the *restrict* function calculates the summation constraints defined by Equation 4, ensuring the integrity of the current composition.

The *compute* function takes as parameters the values s and k, which correspond to the remaining sum and number of parts of the composition, respectively. The other parameters are the variable *composition*, a tuple representing the current interval composition, and the variable r, which is an auxiliary array used to prevent redundancy in the summation calculations made by the *restrict* function. In the first call to the *compute* function, s and k assume the values that were also passed to *restricted_compositions* as inputs, while *composition* and r start as empty structures. During the execution of the algorithm, the parts of *composition* are filled with all possible values, while s and k decrease at each choice. Before calling the recursion, r is updated with the summation values by *restrict*, which acts as an auxiliary function. It should be noted that a branch is created on every call to the *compute* function, as it is called for each possible value to be inserted in *composition*. An interval composition is yielded when there is no part left to choose and the desired sum is reached, i.e., if $k = 0$ and $s = 0$. Finally, the restricted interval compositions are returned and written to the database.

The classification algorithm starts from the database containing all the interval compositions explained above. The goal of this procedure is to group all interval compositions in their respective equivalence classes. In our classification, we identify each class by the *normal form* of its members based on the fact that every member of a class has the same normal form. This gives us a straightforward way of grouping all interval compositions in class. We scan the whole database of interval compositions computing the normal form of each one and group them as we go along. After having completed this, we should have a mapping between normal forms and the members of their corresponding class.

At the core of the algorithm, lies the normalization procedure, i.e., how we compute the normal form of a given interval composition. It consists of two steps:

1. Given an interval composition and the argument to the normalization function, list all members of its class.

2. Given the list of interval compositions generated in the previous step, find the most compact one.

For the first step, in order to list all members of a class given one of its members, we need to first define the functions on interval compositions that we wish to take as defining our classes. For example, if we want our classes to group the interval compositions related by retrogradation, the only function needed is the retrograde function. However, if we want our classes to group the interval compositions related by retrogradation and inversion, we need to use the retrograde and inversion functions. Having that, we can apply them repeatedly to the input interval composition to compute all the members of its class. In order to achieve this, the normalization procedure is parameterized by the set of functions that determines the equivalence class. This allows the user to choose the operations that are relevant to their work.

The second step is the simplest one. Having the list of all interval compositions, we sum all elements of each one of them and pick the ones with the lowest resulting number. If there is only one, that is the normal form. If there are more than one, this process is repeated with the following two changes on each iteration until we have a single interval composition, the normal form:

- We only consider the interval compositions that are tied with the lowest sum in the previous iteration.
- Starting from the last element, we disregard one more element from the right end of the interval compositions while summing them.

Besides classifying the entire database of interval compositions, *SerialGen* is also able to filter them to include in the output only those that match certain user provided criteria. These can be the number of distinct or specific intervals and the presence of a given sequence of intervals.

4. Preliminary Results and Concluding Remarks

With the software we were able to generate all the restricted interval compositions and produce the following tables listing the number of tone rows and row classes, according to their respective combinations of operations (types) [12] and dimensional factors.

The data presented here is not completely new, since there are other works which achieve similar results through group theory. [13] However, the mathematical means to obtain it and its segmentation by dimension provide a new perspective of understanding tone rows and row classes, opening the way for a broader and exhaustive taxonomy.

[12] Some of these row classes (types) are isomorphic to finite groups (cyclic, symmetric, dihedral, etc.).
[13] See FRIPERTINGER and LACKNER (2015), HUNTER and HIPPEL (2003), and REINER (1985).

At last, both the research and the software are still under development, with room for new operations, functions, and applications exploring tone rows and row classes that are symmetrical or combinatorial, among other possibilities, which will be done in due course.

Table 1: Tone rows and row classes of each type by dimensional factors.

d	*s*	Tone Rows	Type 1 T	Type 2 T, RI	Type 3 T, R	Type 4 T, I	Type 5 T, I, R, RI
1	12	12	1	1	1	1	1
2	24	24,432	2,036	1,049	2,036	2,036	1,049
3	36	1,831,644	152,637	76,617	152,637	152,637	76,617
4	48	26,441,856	2,203,488	1,102,940	2,203,488	2,203,488	1,102,940
5	60	116,857,368	9,738,114	4,871,638	9,738,114	9,738,114	4,871,638
6	72	188,690,976	15,724,248	7,865,430	7,864,044	7,862,124	3,933,675
7	84	116,857,368	9,738,114	4,871,638			
8	96	26,441,856	2,203,488	1,102,940			
9	108	1,831,644	152,637	76,617			
10	120	24,432	2,036	1,049			
11	132	12	1	1			
Total		479,001,600	39,916,800	19,969,920	19,960,320	19,958,400	9,985,920

Table 2: Row classes of each type (including ***S***) by dimensional factors.

d	*s*	Type 6 T, S	Type 7 T, RI, S	Type 8 T, R, S	Type 9 T, I, S	Type 10 T, I, R, RI, S
1	12	1	1	1	1	1
2	24	171	101	171	171	101
3	36	12,741	6,579	12,741	12,741	6,579
4	48	183,634	92,415	183,634	183,634	92,415
5	60	811,670	407,546	811,670	811,670	407,546
6	72	1,310,354	656,830	656,137	656,137	329,375
7	84	811,670	407,546			
8	96	183,634	92,415			
9	108	12,741	6,579			
10	120	171	101			
11	132	1	1			
Total		3,326,788	1,670,114	1,664,354	1,664,354	836,017

Table 3: Row classes of each type (including ***S***, ***M***, and ***F***) by dimensional factors.

d	*s*	Type 11 T, I, R, RI, S, M	Type 12 T, I, R, RI, S, F	Type 13 T, I, R, RI, S, M, F
1	12	1	1	1
2	24	101	100	100
3	36	6,563	6,412	6,371
4	48	87,170	83,834	74,898
5	60	260,327	258,977	123,257
6	72	65,251	70,089	6,385
Total		419,413	419,413	211,012

References

ANDREWS, George. *Number Theory*. New York: Dover, 1994.

ANDREWS, George. *The Theory of Partitions*. Cambridge: Cambridge University Press, 1998.

ANDREWS, George; ERIKSSON, Kimmo. *Integer Partitions*. Cambridge: Cambridge University Press, 2004.

FEITOSA, Marco. Partitional Harmony: The Partitioning of Pitch Spaces. *MusMat: Brazilian Journal of Music and Mathematics*, v. 4, n. 2, 2020, pp. 1-27.

FRIPERTINGER, Harald.; LACKNER, Peter. Tone Rows and Tropes. *Journal of Mathematics and Music*, v. 9, n. 2, 2015, pp. 111-172.

HUNTER, David; HIPPEL, Paul. How Rare Is Symmetry in Musical 12-Tone Rows? *The American Mathematical Monthly*, v. 110, n. 2, 2003, pp. 124-132.

MORRIS, Robert. Review of "Tone Rows and Tropes" by Harald Fripertinger and Peter Lackner. *Journal of Mathematics and Music*, Vol. 9, No. 2, 2015, pp. 179-195.

REINER, D. L. Enumeration in Music Theory. *The American Mathematical Monthly*, Vol. 92, No. 1, 1985, pp. 51-54.

SCHOENBERG, Arnold. *Structural Functions of Harmony*. 2nd ed. New York: W. W. Norton & Company, 1969.

SCHOENBERG, Arnold. *Style and Idea*. New York: Philosophical Library, 1950.

STRAUS, Joseph. *Introduction to Post-Tonal Theory*. 4th ed. New York: W. W. Norton & Company, 2016.

TIPTON, Lisa. *Schoenberg's "Phantasy" Form*. New York, 2017. 182f. Dissertation (Doctor of Musical Arts). Graduate Faculty in Music, City University of New York, New York, 2017.