In this paper, we propose an integer linear programming model whose solutions are the aperiodic rhythms tiling with a given rhythm $A$. We show how it can be used to define an iterative algorithm that, given a period $n$, finds all the rhythms which tile with a given rhythm $A$ and also to efficiently check the necessity of the Coven-Meyerowitz condition (T2). To conclude, we run several experiments to validate the time efficiency of the model.

**Keywords:** Integer linear programming; mathematics and music; tiling Problems; Vuza canons; (T2) conjecture

1. Introduction

In this paper, we deal with the mathematical and computational aspects of a musical problem that draws the interest of mathematicians, computer scientists, music theorists, and composers (Amiot 2004; Andreatta 2004): the construction of Vuza canons, i.e. tiling rhythmic canons without repetitions within the individual voices.

A *tiling rhythmic canon* is a purely rhythmic canon through which the composer tries to fill the rhythmic space, with no superimposition of the different voices (Vuza 1991–1993). From a mathematical point of view, it can be described as a factorization of an Abelian group with two subsets, $A$ and $B$, called *rhythms*: $A \oplus B = \mathbb{Z}_n$. In this case, we say that $A$ is a *tiling complement* of $B$, or, equivalently, that $A$ *tiles with* $B$ (and vice versa) the set $\mathbb{Z}_n$. Equivalently, a tiling rhythmic canon can also be described in terms of polynomials with coefficients in $\{0, 1\}$. Indeed, the two interesting conditions (T1) and (T2) are expressed in terms of these polynomials: together they are sufficient for the existence of tiling rhythmic canons, and (T1) is also necessary (Coven and Meyerowitz 1999). The necessity of condition (T2), on the other hand, remains an open question, and it is in this context that Vuza canons play a fundamental role.

**Theorem 1.1 (Amiot 2005a)** If a rhythmic canon does not satisfy condition (T2), it is possible to collapse it to (that is, it is obtainable by concatenation and/or duality from) a Vuza canon that does not satisfy (T2).

---

*Corresponding author. Email: g.lanzarotto@campus.unimib.it

1 Concatenation and duality are two transformations applicable to rhythmic canons: the first consists in replacing one of the two rhythms with different copies of the same, the second in the exchange of the two rhythms.
This means that the problem of determining whether condition (T2) is necessary or not can be reduced to the investigation of Vuza canons.

Finding a general procedure to construct Vuza canons has been the subject of many studies, see for example Hajós (1950), de Bruijn (1953a), Vuza (1991–1993), Jedrzejewski (2006), Fidanza (2007), and Lanzarotto and Pernazza (2022), but an exhaustive construction method is not known to date.

In this paper, we introduce a linear problem whose set of solutions is composed of all the aperiodic tiling complements of a given rhythm. In particular, we impose the aperiodicity of the solution through a family of linear constraints.

The purpose of our model is twofold. First, we want to determine, for a given rhythm \( A \), all the tiling complements \( B \) in \( \mathbb{Z}_n \). In this case, we are interested in testing the tiling property and finding all the complements of \( A \). Given a rhythm \( A \) and a period \( n \), Kolountzakis and Matolcsi’s (2009) fill-out procedure (FP) provides a complete enumeration of the complements of \( A \) in \( \mathbb{Z}_n \). The main idea behind their algorithm is to use packing complements and add one by one the new elements discovered by an iterative search. To the best of our knowledge, it is the only algorithm able to provide the complete list of complements of a given rhythm for \( n \leq 200 \). For larger \( n \), the problem has been considered by Jedrzejewski (2013) and Lanzarotto and Pernazza (2022), but they were only able to give a lower bound on the number of tiling complements. Therefore, we choose to compare the performances of our model with the ones of the fill-out procedure. Secondly, we aim to determine if a given aperiodic rhythm \( A \), which does not satisfy the (T2) property, tiles with an aperiodic rhythm \( B \). Then we could efficiently test possible counterexamples to the necessity of condition (T2) (Amiot 2011).

The tiling problem is very similar to the decision problem \( \text{DIFF}^2 \) considered by Kolountzakis and Matolcsi (2006), which has been shown to be NP-complete. This fact suggests a lower bound on the computational complexity of the tiling decision problem, which is consistent with our formulation, since it requires solving an integer linear system of \( 3n - 1 \) unknowns and \( 3n + 3(M_n(p) - 1) \) constraints, where \( M_n(p) \) denotes the number of all distinct primes in the factorization of \( n \). Of course, by solving our system once, we find only one of all the possible solutions. However, we can update the problem by removing the found solution from the feasible set. If we solve the updated problem, we can find a new solution. We will find all the tiling complements of the given rhythm \( A \) by iterating this process until the problem cannot be solved. Since we are not interested in looking for all the possible solutions but rather for all the classes of equivalent rhythms modulo translations or affine transformations, we customize the constraints added at each step. We notice that, if we are interested in finding all the solutions modulo affine transformations, the number of constraints to add at each iteration is equal to the cardinality of \( \mathcal{P} = \{ a \in \mathbb{N} : \gcd(a, n) = 1 \} \) times the cardinality of the set of all translations fixing the first entry of the solution equal to 1. Therefore, we add \( O(|\mathcal{P}|n/n_A) \) new constraints at every iteration, where \( n_A \) is the cardinality of the rhythm \( A \). As a result, finding new tiling rhythms gets more challenging at each iteration.

The outline of the paper is the following.

In Section 2, we recall the main notions and results about tiling rhythmic canons and formulate the tiling problem.

In Section 3, we reformulate the tiling problem as an integer linear programming problem. We endow the obtained system with additional constraints to impose the aperiodicity of the solution. We then define an iterative algorithm to find all the tiling complements of a given rhythm.

In Section 4, we report the results of our tests. We compare the time required by our method with the one required by the fill-out procedure.

---

2 Given an integer \( n \) and two sets \( E, D \subseteq \mathbb{Z}_n \), the \( \text{DIFF}^2 \) problem consists in determining the maximum value \( k \) such that \( |A| = k, A \subseteq E \), and \( (A - A) \subseteq D \).
To conclude, in Section 5, we outline the possible future directions of the research.

2. Tiling in music

This section fixes our notation and recalls the main notions about rhythmic canons in mathematics. For a complete and exhaustive discussion of this topic, we refer to Amiot (2011).

Definition 2.1 A tiling rhythmic canon \((A, B)\) with period \(n\) is a factorization of the cyclic group \(\mathbb{Z}_n\) with subsets \(A\) and \(B: A \oplus B = \mathbb{Z}_n\).

Let us now consider for a moment that a (finite) rhythm \(A'\) tiles the whole set \(\mathbb{Z}\), i.e. there exists \(C' \subset \mathbb{Z}\) such that \(A' \oplus C' = \mathbb{Z}\). Then, by Tijdeman’s (1995) theorem,

\[
A' \oplus C' = A' \oplus (B' \oplus n\mathbb{Z}) = (A' \oplus B') \oplus n\mathbb{Z} = \mathbb{Z} \implies A \oplus B = \mathbb{Z}_n,
\]

that is, \(A \oplus B\) gives a complete set of residues modulo \(n\), where \(A\) denotes the elements of \(A'\) taken modulo \(n\). In other words, any canon repeats itself for some period \(n\). Moreover, \(A \oplus B = \mathbb{Z}_n\) implies that \(n_A n_B = n\), i.e. the period \(n\) of the canon must be a multiple of the cardinality of rhythm \(A\).

It is possible to characterize tiling rhythmic canons through polynomials:

Definition 2.2 Let \(A \subset \mathbb{N}\) be finite. The characteristic polynomial of \(A\) is defined as

\[
A(x) = \sum_{k \in A} x^k.
\]

Lemma 2.3 Let \(A(x), B(x) \in \mathbb{N}[x]\) and \(n\) be a positive integer. Then

\[
A(x) B(x) \equiv \sum_{i=0}^{n-1} x^i \mod (x^n - 1)
\]

if and only if

1. \(A(x), B(x) \in \{0, 1\}[x]\), and
2. \(A \oplus B = \{r_1, \ldots, r_n\} \subset \mathbb{Z}\) with \(r_i \neq r_j \mod n\) for each \(i, j \in \{1, \ldots, n\}, i \neq j\).

Remark 2.1 Note that

\[
\sum_{i=0}^{n-1} x^i = \frac{x^n - 1}{x - 1} = \prod_{d|n, d \neq 1} \Phi_d(x)
\]

where \(\Phi_d(x)\) is the \(d\)th cyclotomic polynomial, that is, the minimal polynomial of any primitive \(d\)th root of unity over the field of rational numbers.

An important property exploited in our algorithm is the invariance of the set of solutions under affine transformations.
Theorem 2.4 (Vuza 1991–1993) Let $A \oplus B = \mathbb{Z}_n$ be a canon and $f : \mathbb{Z}_n \to \mathbb{Z}_n$ be an affine transformation of $\mathbb{Z}_n$ that is

$$f : x \mapsto ax + b \mod n,$$

where $a$ is coprime with $n$ and $b \in \mathbb{Z}_n$. The affine transform of $A$ by $f$ still tiles with $B$; i.e., $(aA + b) \oplus B = \mathbb{Z}_n$.

Definition 2.5 Let $k|n$ be a non-zero element of $\mathbb{Z}_n$. A rhythm $A \subset \mathbb{Z}_n$ is periodic modulo $k$ if and only if $k + A = A$. A rhythm $A \subset \mathbb{Z}_n$ is aperiodic if and only if it is not periodic modulo any $k \in \mathbb{Z}$ such that $k|n$.

Remark 2.2 Note that a set $A$ is periodic modulo $k|n$ if and only if

$$\chi^k - 1 \bigg| \chi^k - 1 \bigg| A(x).$$

Whenever a rhythm $A$ is periodic modulo $k|n$, with $k \neq n$, it is periodic modulo all multiples of $k$ dividing $n$. As a result, to check whether $A$ is periodic or not, it suffices to check if it is periodic modulo $m_1 = p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_N^{\alpha_N}$, $m_2 = p_1^{\alpha_1}p_2^{\alpha_2-1}\cdots p_N^{\alpha_N}$, ..., $m_N = p_1^{\alpha_1}p_2^{\alpha_2-1}\cdots p_N^{\alpha_N-1}$, where $n = p_1^{\alpha_1}p_2^{\alpha_2}\cdots p_N^{\alpha_N}$ is the prime power factorization of $n$.

Definition 2.6 A tiling rhythmic canon $(A, B)$ in $\mathbb{Z}_n = A \oplus B$ is a Vuza canon if both $A$ and $B$ are aperiodic.

The existence of Vuza canons depends on the order of the cyclic group $\mathbb{Z}_n$ to be factorized. Hajós (1950) proposed the following definition.

Definition 2.7 A finite Abelian group $G$ is a good group if in any tiling $G = A \oplus B$ one of the two subsets $A$ and $B$ has to be periodic. $G$ is a bad group if there exists a tiling $G = A \oplus B$ where $A$ and $B$ are aperiodic.

Hajós (1950), Rédei (1950), de Bruijn (1953b), and Sands (1974) completely characterized all good and bad groups. Moreover, they partitioned the set of finite cyclic groups into two disjoint classes. In particular,

- the good groups, for which there are no Vuza canons, have orders in

$$\left\{ p_1^{\alpha_1}p_2^{\alpha_2}p_3^{\alpha_3}p_4^{\alpha_4} : \alpha_1 \in \mathbb{N} \right\},$$

where $p_1, p_2, p_3, p_4$ are distinct primes, and

- the bad groups have orders $n = P_1N_1P_2N_2N_3$, with

  - $\gcd(P_1N_1, P_2N_2) = 1$, and
  - $P_1, N_1, P_2, N_2, N_3 \geq 2$.

Therefore, the analysis on Vuza canons exclusively concerns these last cyclic groups, whose orders are explicitly identified.

Coven and Meyerowitz (1999) found two sufficient conditions for a (periodic or aperiodic) rhythmic pattern to tile. They proved that those conditions are necessary under specific hypotheses. However, it is not clear whether, in the general case, the second condition is necessary as well. The polynomial representation of tiling rhythmic canons seems quite suitable for presenting these results. To state the conditions introduced by Coven and Meyerowitz, we need to define two sets based on the cyclotomic polynomials that divide the characteristic polynomial of the rhythm we are considering, as explained in Remark 2.1.
Definition 2.8 Let \( A \subset \mathbb{N} \) be finite and \( n \in \mathbb{N} \) such that \( n_A | n \). We define:

- \( R_A \doteq \{ d \in \mathbb{N} : d | n, \ \Phi_d(x) | A(x) \} \),
- \( S_A \doteq \{ d \in R_A : d = p_1^{\alpha_1} \) with \( p_1 \) prime and \( \alpha_1 \in \mathbb{N}^* \} \),

where \( \mathbb{N}^* \doteq \mathbb{N} \setminus \{0\} \).

We can now state the following.

**Theorem 2.9 (Coven and Meyerowitz 1999)** Let us consider the conditions:

(T1) \( A(1) = \prod_{p_i^{\alpha_i} \in S_A} p_i \);

(T2) if \( p_1^{\alpha_1}, \ldots, p_M^{\alpha_M} \in S_A \), where \( p_1^{\alpha_1}, \ldots, p_M^{\alpha_M} \) are powers of distinct primes, then \( p_1^{\alpha_1} \ldots p_M^{\alpha_M} \in R_A \).

Then

1. if \( A \) satisfies (T1) and (T2), then it tiles;
2. if \( A \) tiles, then it satisfies (T1);
3. if \( A \) tiles and \( n_A \) has at most two prime factors, then \( A \) satisfies (T2).

Determining whether condition (T2) is necessary for a rhythm \( A \) to tile is still an open question. Łaba and Londner proved that condition (T2) holds for all integer tilings with period \( n = p_1^2 p_2^2 p_3^2 \), where \( p_1, p_2, p_3 \) are distinct odd primes.

**Theorem 2.10 (Łaba and Londner 2022)** Let \( n = p_1^2 p_2^2 p_3^2 \), where \( p_1, p_2, p_3 \) are distinct odd primes. Assume that \( A \oplus B = \mathbb{Z}_n \), with \( n_A = n_B = p_1 p_2 p_3 \). Then both \( A \) and \( B \) satisfy condition (T2).

Another step in this direction has been made by Łaba who proved, furthering Amiot’s (2005b) reduction idea combined with an equiepartition lemma, that condition (T2) is necessary when the period \( n \) is square-free.\(^3\)

**The fill-out procedure**

Kolountzakis and Matolcsi (2009) introduced an algorithm for tiling problems articulated into five phases: (i) recognition of the prime powers that divide \( n \) and of all their partitions; (ii) elimination of the partitions that do not lead to Vuza canons due to condition (T2); (iii) listing the rhythms achievable through the Coven-Meyerowitz theory and sorting them into equivalence classes based on the zero-set of their Fourier transforms; (iv) discarding the classes that make the tiling complements periodic; (v) application of the fill-out procedure to the remaining classes. In this paragraph, we briefly describe the last part, i.e. the fill-out procedure, that can be used to find all the tiling complements of a given rhythm \( A \). The routine of the procedure is the following: given a rhythm \( A \subset \mathbb{Z}_n \) such that \( 0 \in A \), the algorithm sets \( P = \{0\} \) and starts the search for possible expansions of the set \( P \). The expansion is done by adding one element \( \alpha \in \mathbb{Z}_n \) to \( P \) at the time. To select which element \( \alpha \) to add, they introduce a function \( r(x, P) \) that ranks the elements in \( x \in \mathbb{Z}_n \setminus P \) by counting all the possible ways in which \( x \) can be covered through a translation of \( A \). After every element of \( \mathbb{Z}_n \setminus (A \oplus P) \) has been ranked, the algorithm adds the element with the lowest rank. This expansion defines a new set, namely \( \hat{P} \supset P \), which is again expanded until either it can no longer be expanded or the set becomes a tiling complement. The search ends when

---

\(^3\)https://terrytao.wordpress.com/2011/11/19/some-notes-on-the-coven-meyerowitz-conjecture/
all the possibilities have been explored. The algorithm also finds periodic solutions, therefore, after the algorithm finishes its routine, an additional post-processing step removes the periodic solutions along with the multiple translations of the same aperiodic rhythm.

3. A linear model for tilings in \( \mathbb{Z}_n \)

In this section, we introduce our model and our algorithm. First, we briefly overview the integer linear programming problems (ILPs). Second, we define the linear equations that describe the tiling property. Afterward, we impose the aperiodicity constraints. Our main result is Theorem 3.2, where we state that imposing the aperiodicity of the solution can be done through linear constraints. Finally, we show how solving a sequence of increasingly harder ILPs leads to a complete enumeration of all the tiling complements of a given rhythm \( A \).

**Integer linear programming problems**

In this subsection, we briefly recall what integer linear programming problems (ILPs) are and what are the techniques used to solve them. For a complete discussion on the ILPs, we refer the reader to Conforti, Cornuéjols, and Zambelli (2014).

Integer linear programming problems are linear optimization problems in which the solution is constrained to have integer values. In its most general form, an integer linear programming problem can be described as it follows:

\[
\max c^\top x \\
Ax \leq b, \quad x \in \mathbb{Z}^n
\]

where \( A = (a_{ij}) \) is a rational \( m \times n \) matrix, \( c = (c_1, c_2, \ldots, c_n) \) is a cost vector, \( b = (b_1, b_2, \ldots, b_m) \) is the right-hand-side of the constraints, and \( x \) is an integer decision variable whose entries are, indeed, integer values. ILPs are known to model real-world applications such as scheduling, production planning, and telecommunications problems (Wolsey 1998; Conforti, Cornuéjols, and Zambelli 2014). However, they are way more time-demanding than a regular linear programming (LP) problem, in which the solution can assume any real value and not only integer ones. Indeed, ILP problems are NP-hard, while LP problems are solvable in polynomial time. To make ILPs more reasonable from a computational point of view, several solving methods have been proposed, relying on different iteration of the cutting plane (CP) method (Gilmore and Gomory 1961; Marchand et al. 2002) or the branch-and-bound (B&B) algorithm (Land and Doig 1960). Both these techniques are known to be reliable tools to approach ILPs and they are used by state-of-the-art solvers such as Gurobi. It is worthy of mention that modern solvers are able to tackle problems of any size practically, since they have no limits on the number of variables or constraints. However, despite the efficiency of the recent solving methods, it seems that the hardness of a problem is related to the sparseness of its parameters. For example, dense models with thousands of constraints and thousands of variables are known to be difficult to be solved with Gurobi.

**Feasibility condition**

Let us consider an inner rhythm \( A \) and a possible outer rhythm \( B \). Since the degrees of their characteristic polynomials, \( A(x) \) and \( B(x) \), are both less than or equal to \( n - 1 \), the degree of the product \( R(x) \) is less than or equal to \( 2n - 2 \). We denote by \( r \) the vector with \( 2n - 1 \) entries...
containing the coefficients of the polynomial $R(x) \equiv A(x)B(x)$. From Lemma 2.3, we know that $B$ tiles with $A$ if and only if

$$R(x) \equiv 1 + x + x^2 + \cdots + x^{n-1} \mod x^n - 1. \quad (1)$$

We can express condition (1) through $n$ linear equations

$$r_i + r_{i+n} = 1 \quad \forall i = 0, \ldots, n - 1.$$ 

Therefore, we can express the constraint

$$R(x) = A(x)B(x) \equiv \sum_{i=0}^{n-1} x^i \mod x^n - 1,$$

through the linear system

$$F_i(B) - r_i = 0 \quad \forall i \in \{0, \ldots, 2n - 2\},$$

$$r_j + r_{j+n} = 1 \quad \forall j \in \{0, \ldots, n - 1\},$$

where $F_i(B)$ is the function that associates to a rhythm $B$ the $i$th coefficient of $A(x)B(x)$, that is

$$F_0(B) \equiv a_0b_0,$$

$$F_1(B) \equiv a_1b_0 + a_0b_1,$$

$$F_2(B) \equiv a_2b_0 + a_1b_1 + a_0b_2,$$

$$\vdots \quad \vdots$$

$$F_{2n-2}(B) \equiv a_{n-1}b_{n-1},$$

where $b = (b_0, b_1, \ldots, b_{n-1})$ are the coefficients of $B(x)$. Notice that, since $A$ is given, all the equations presented above are linear with respect to the variables $b_i$ and $r_i$. We then can express them through a linear system

$$A \cdot \mathcal{X} = \mathcal{Y}, \quad (2)$$

where

- $A$ is a $(3n - 1) \times (3n - 1)$ matrix which depends only on the given rhythm $A$;
- $\mathcal{X} = (b, r)^T$ is the vector composed by the coefficients of $B(x)$ and the coefficients of $R(x)$;
- $\mathcal{Y}$ is the $(3n - 1)$-dimensional vector defined as

$$\mathcal{Y}_i = \begin{cases} 0 & \text{if } i \in \{0, \ldots, 2n - 2\} \\ 1 & \text{otherwise.} \end{cases}$$

Finally, in order to ensure that $B(x)$ and $R(x)$ are polynomials with coefficients in $\{0, 1\}$, we will require $b_i$ and $r_i$ to be binary variables, i.e. they can only assume value 0 or 1.
**Aperiodicity constraints**

Let us assume \( n = p_1^{a_1}p_2^{a_2} \cdots p_N^{a_N} \). Without loss of generality, we can assume

\[
p_1 < p_2 < \cdots < p_N.
\]

Hence, the set of maximal divisors \( \mathcal{M}_n \doteq \{ m_k = \frac{n}{p_k} \}_{k=1}^{N} \), is such that

\[
m_N < m_{N-1} < \cdots < m_1.
\]

According to Remark 2.2, to verify whether rhythm \( B \) is periodic or not, it is sufficient to check its periodicity for periods in \( \mathcal{M}_n \). We can characterize the periodicity modulo a given period \( m_j \) as follows.

**Proposition 3.1** Let \( B \) be a rhythm in \( \mathbb{Z}_n \), let \( b \) be the binary vector containing the coefficients of \( B(x) \), and let \( m_j \in \mathcal{M}_n \). Then, \( B \) is periodic modulo \( m_j \) if and only if

\[
\sum_{r=0}^{p_j-1} b_{i+rm_j} = \begin{cases} p_j & \text{if } i \in B \\ 0 & \text{otherwise}, \end{cases}
\]

for each \( i = 0, \ldots, m_j - 1 \).

**Proof** Let us assume \( B \subset \mathbb{Z}_n \) is periodic modulo \( m_j \). We prove that (3) holds. By Definition 2.5, we have that

\[
i \in B \iff i + rm_j \in B
\]

for each \( r = 0, \ldots, p_j - 1 \). Let \( b \) be the vector of the coefficients of \( B(x) \). By Equation (4), we get

\[
b_i = 0 \iff b_{(i+rm_j)} \mod n = 0 \quad \text{for } r = 0, \ldots, p_j - 1, \tag{5}
\]

\[
b_i = 1 \iff b_{(i+rm_j)} \mod n = 1 \quad \text{for } r = 0, \ldots, p_j - 1, \tag{6}
\]

therefore, for any given \( i = 0, \ldots, m_j - 1 \), we have

\[
\sum_{r=0}^{p_j-1} b_{i+rm_j} = \begin{cases} p_j & \text{if } i \in B \\ 0 & \text{otherwise}, \end{cases}
\]

which concludes the first half of the proof.

Let us now assume that (3) holds and fix \( i \in \{0, \ldots, m_j - 1\} \). If

\[
\sum_{r=0}^{p_j-1} b_{i+rm_j} = 0,
\]

we have \( b_{i+rm_j} = 0 \) for each \( r = 0, \ldots, p_j - 1 \), since each \( b_k \) is either equal to 0 or 1, which is equivalent to (5). Similarly, if

\[
\sum_{r=0}^{p_j-1} b_{i+rm_j} = p_j,
\]

we have \( b_{i+rm_j} = 1 \) for each \( r = 0, \ldots, p_j - 1 \), which is equivalent to (6). Since (5) and (6) are equivalent to the periodicity modulo \( m_j \) of \( B \), the proposition follows.
Let us take $m_j \in M_n$. To impose that the rhythm $B$ is not periodic modulo $m_j$, we introduce the family of auxiliary variables

$$\mathcal{U}^{(j)} \doteq \{ U_i^{(j)} \}_{i=1,\ldots,m_j-1}. $$

Each family $\mathcal{U}^{(j)}$ is composed of binary variables subject to the following constraints.

$$\sum_{k=0}^{p_j-1} b_{i+km_j} - p_j U_i^{(j)} \leq p_j - 1, \quad (7)$$

$$\sum_{k=0}^{p_j-1} b_{i+km_j} - p_j U_i^{(j)} \geq 0, \quad (8)$$

$$\sum_{i=0}^{m_j-1} U_i^{(j)} \leq \frac{n_B}{p_j} - 1, \quad (9)$$

for each $j$ such that $p_j | n_B$ and for each $i = 0, \ldots, m_j - 1$, where $n_B = \frac{n}{n_B}$ is the cardinality of $B$. Since $\sum_{k=0}^{p_j-1} b_{i+km_j} \leq p_j$, condition (7) assures us that we have $U_i^{(j)} = 1$ whenever

$$\sum_{k=0}^{p_j-1} b_{i+km_j} = p_j. $$

Condition (8) assures us that $U_i^{(j)} = 1$ only if

$$\sum_{k=0}^{p_j-1} b_{i+km_j} = p_j. $$

Therefore, conditions (7) and (8) combined, assure us that

$$U_i^{(j)} = 1 \iff \sum_{k=0}^{p_j-1} b_{i+km_j} = p_j. $$

Since $\sum_{i=0}^{n-1} b_i = n_B$, if $\sum_{i=0}^{m_j-1} U_i^{(j)} = \frac{n_B}{p_j}$, it follows that

$$\sum_{k=0}^{p_j-1} b_{i+km_j} = \begin{cases} p_j & \text{if } U_i^{(j)} = 1 \\ 0 & \text{otherwise,} \end{cases}$$

and, hence, according to Proposition 3.1, $B$ is periodic modulo $m_j$. By adding constraints (7), (8), and (9) to the linear system, we, therefore, remove all the periodic solutions from the feasible set.

**Remark 3.1** To improve efficiency, we remove a family of auxiliary variables $\mathcal{U}^{(j)} \doteq \{ U_i^{(j)} \}$ imposing

$$\sum_{i=0}^{m_j-1} b_i \leq \frac{n_B}{p_j} - 1. \quad (10)$$

Indeed, if $B$ is not periodic modulo $m_j$, there must exist a translation of $B$ such that (10) holds. Since $U^{(1)}$ is the family containing the highest number of variables, and therefore the one more memory demanding, we choose to remove it.
Conditions (7)–(9) and (10) are linear for any $j$, therefore, we can add them to the system described in (2) and obtain the following ILP.

\[
\min \quad \mathcal{O} \left( \{b_i\}, \{r_i\}, \left\{ U_i^{(j)} \right\} \right)
\]

\[
\text{s.t.} \quad \sum_{j=0}^{i} a_{i-j} b_j - r_i = 0 \quad \forall i \in \{0, \ldots, n-1\},
\]

\[
\sum_{j=0}^{(i+1)} a_{n-(i-j)} b_j - r_{i+n} = 0 \quad \forall i \in \{0, \ldots, n-2\},
\]

\[
r_j + r_{j+n} = 1 \quad \forall j \in \{0, \ldots, n-1\},
\]

\[
\sum_{j=0}^{m_0-1} b_j \leq n_B \frac{m_0}{n} - 1,
\]

\[
\sum_{j=0}^{p_j-1} b_{i+km_j} - p_j U_i^{(j)} \leq p_j - 1, \quad \forall j \in \{1, \ldots, N\},
\]

\[
\forall i \in \{0, \ldots, m_j - 1\},
\]

\[
\sum_{j=0}^{p_j-1} b_{i+km_j} - p_j U_i^{(j)} \geq 0, \quad \forall j \in \{1, \ldots, N\},
\]

\[
\forall i \in \{0, \ldots, m_j - 1\},
\]

\[
\sum_{i=0}^{m_j-1} U_i^{(j)} \leq \frac{n_B}{p_j} - 1, \quad \forall j \in \{1, \ldots, N\},
\]

\[
\forall i \in \{0, \ldots, m_j - 1\},
\]

\[
b_0 = 1,
\]

\[
b_k \in \{0, 1\} \quad \forall k \in \{1, \ldots, n-1\},
\]

\[
r_k \in \{0, 1\} \quad \forall k \in \{0, \ldots, 2n-2\},
\]

\[
U_i^{(j)} \in \{0, 1\} \quad \forall j \in \{1, \ldots, N\},
\]

\[
\forall i \in \{0, \ldots, m_j - 1\},
\]

where $\mathcal{O}$ is a suitable linear function to minimize. Since the tiling property is invariant under translations, we can add without loss of generality the constraint (19), which allows us to reduce the size of the feasible set by removing a degree of freedom from the possible solutions. We denote the model just introduced the master problem (MP).

**Theorem 3.2** Given an inner rhythm $A$ in $\mathbb{Z}_n$, let $\hat{Y} = (b, r)$ be a solution of MP. Then, the rhythm associated with the characteristic polynomial

\[
B(x) \doteq \sum_{i=0}^{n-1} b_i x^i,
\]

is aperiodic and tiles with $A$. 

---

*Note: The document includes mathematical expressions and logical flow that are typical in a technical or scientific context, focusing on optimization and algebraic constraints.*
Remark 3.2  The set of constraints of the MP fully characterizes the possible aperiodic rhythms tiling with a given rhythm \( A \).

The functional \( \mathcal{O} \) does not play any role; however, one can use it to induce an order or selection criteria on the space of solutions. For example, let us consider the functional

\[
\mathcal{O} (b, r) \equiv \sum_{i=0}^{2n-2} i^2 b_i,
\]

which prefers the tiling complements whose first components are as full as possible of 1’s. Choosing the right functional \( \mathcal{O} \) can help in discerning, among all the possible tiling complements of the given rhythm \( A \), the ones we want to find. However, since the aim of our tests is to find all possible tilings, we will not need to impose any selection criteria and, therefore, we set

\[
\mathcal{O} (b, r) \equiv 0.
\]

**Cutting sequential algorithm**

Once we find an aperiodic rhythm \( B^{(1)} \) tiling with a given rhythm \( A \), we can remove \( B^{(1)} \) from the set of all possible solutions \( D_A \) and obtain a new set of feasible solutions \( D^{(1)}_A \). Let us denote with \( \text{MP}^{(1)} \) the restriction on \( D^{(1)}_A \) of MP and call \( B^{(2)} \) the solution of \( \text{MP}^{(1)} \); we can then remove \( B^{(2)} \) from \( D^{(1)}_A \), define the set \( D^{(2)}_A \), and define \( \text{MP}^{(2)} \), starting the whole process again. By repeating the process until we find an unsolvable problem, we retrieve all the possible solutions of the original master problem and, therefore, we generate all the aperiodic rhythms tiling with the rhythm \( A \).

In this subsection, we detail how to cut out from the feasible set the solution found at each iteration.

Let \( B^{(1)} \) be a rhythm tiling with \( A \) and let \( b^{(1)} = (b_0, \ldots, b_{n-1}) \) be the coefficients of its characteristic polynomial. We denote with \( I^{(1)} \) the set of non-zero coordinate indexes of the vector \( b^{(1)} \), that is

\[
I^{(1)} \equiv \{ i \in \{0, \ldots, n-1\} : b_i = 1 \}.
\]

We then define a new linear system by adding the constraint

\[
\sum_{i \in I^{(1)}} b_i^{(1)} \neq \frac{n}{n_A},
\]

or equivalently

\[
\sum_{i \in I^{(1)}} b_i^{(1)} \leq \frac{n}{n_A} - 1,
\]

to the MP. By solving the new problem, we find a new solution \( b^{(2)} \neq b^{(1)} \) of the initial tiling problem. We iterate the procedure until we find an unsolvable problem. All the solutions found during the process are stored in memory and given as the final output of the algorithm.

In Algorithm 1, we sketch the pseudocode.

Remark 3.3  Adding the constraints one by one is highly inefficient. Therefore, once we find a solution, we compute all its affine transformations, which, according to Theorem 2.4, are possible solutions and remove them as well. Since we impose \( b_0 = 1 \), we consider only the affine transformations that preserve this constraint.

The procedure, however, is customizable: if we remove only the translations of the found solution, the algorithm will return all the solutions modulo translations. Given a solution \( b^{(1)} \), we
Algorithm 1: The Cutting Sequential Algorithm.

Input: rhythm $A$
Output: $S$, list of aperiodic rhythms $B$, such that $A \oplus B = \mathbb{Z}_n$

1. $z^* = \text{OPT}(\text{MP})$
2. add $z^*$ to $S$
3. while $P \neq \emptyset$ do
   4. add $\sum_{i \in I_z} b_i \leq \beta$ to $\text{MP}^{(i)}$
   5. Solve $\text{MP}^{(i)}$
   6. $z_{\text{new}} = \text{OPT}(\text{MP}^{(i)})$
   7. set $I_z \triangleq I_{z_{\text{new}}}$
   8. add $z_{\text{new}}$ to $S$
4. end
10. return $S$

can remove the affine transformations of a given solution through a linear constraint. According to (21), we impose

$$\sum_{i \in I_z} b_{a(i+k)} \leq n_B - 1 \quad (22)$$

where $k$ runs over all the translations which fix the first position and $a$ runs over the set of numbers coprime with $n$.

**Complexity of the method**

To conclude, we analyze the complexity of the system (2). The unknowns to determine are the $3n - 1$ coordinates of the vector $(b, r)$ plus the variables needed to impose the aperiodicity constraints, $U_i^{(i)}$, which are

$$\sigma_n \triangleq \sum_{p \in \mathbb{P}_n \backslash \{p_0\}} \frac{n}{p},$$

where $\mathbb{P}_n$ is the set of primes that divide $n_B$. Therefore, we have $3n - 1$ constraints for the feasibility, the $3\sigma_n$ given by conditions (16), (17), and (18) plus the one given by condition (15).

If we want a complete enumeration of all the tiling complements of the given rhythm, the complexity increases since we add constraints at each iteration. The number of conditions to add depends on the equivalence relation we are considering. If we look for all the solutions modulo translation, we add $n_B$ constraints at each iteration, since precisely $n_B$ feasible translations preserve the condition $b_0 = 1$. If we search for all the solutions up to affine transformations, the number of constraints added is $n_B$ times the quantity of numbers coprime with $n$.

4. **Numerical results**

In this section, we report the results of our tests.

We run our tests in two frameworks. In the first one, we aim to find *all* the complements of a given rhythm. We compare the CSA with the FP on rhythms in $\mathbb{Z}_n$, for $n \in \{72, 108, 120, 144, 168, 180\}$. We also calculate all the Vuza canons for particular choices of periods $200 \leq n \leq 450$ and parameters $P_1, N_1, P_2, N_2, N_3$ examined by Jedrzejewski (2013). In the
second one, we want to determine if a given aperiodic rhythm tiles, i.e. if there exists at least one aperiodic complement of that rhythm. This simplification allows us to test our methods on larger values of $n$.

We run all our experiments on an ASUS VivoBook15 with Intel Core i7. The algorithm is implemented in Python using Gurobi v9.1.1 (Gurobi Optimization LLC, 2021).

**Remark 4.1** We choose to run our experiments using Gurobi since it is one of the state-of-the-art solvers for ILP problems. As we noticed in Section 3, Gurobi uses its own B&B method to find a solution, although it might be possible to define an ad hoc B&B method that runs better than the standard one used by Gurobi, defining such method is beyond the aims of the present paper.

**Runtimes**

The experiment we run is the following. Given an aperiodic rhythm $B$ built according to the recipe by Coven and Meyerowitz (1999) (and therefore characterized by a set $R_B$ not including $n$), we list every complement $A$. We sort them by different sets $R_A$, keeping a representative $A$ for each possibility. Afterward, we reverse the problem: for every remaining rhythm $A$, we look for all the complements $B$ using the CSA. Thus, given a period $n$ and the not necessarily disjoint sets $R_A$ and $R_B$, we are able to retrieve all the rhythms that tile together.

The rhythms used for our experiments along with the respective number of tilings up to translation and, in brackets, up to affine transformation, are reported in Table 1 while, in Table 2, we compare the runtimes of CSA with the runtimes of the FP. The sets in gray express a choice of $R_A$ (or $R_B$) already considered in a previous row: indeed, the couple $(R_A, R_B)$ can have non-empty intersection, giving rise to different scenarios, as it happens for the case of $n = 144$.

We notice that the illustrated algorithm performs better than the FP on all instances. This strongly suggests that adding the aperiodicity conditions from the beginning rather than removing the periodic solutions as a post-processing step reduces the overall runtime of the implementation. Indeed, for a given rhythm, there might be an overwhelming number of periodic complements that our algorithm simply cuts out. This also explains why some runtimes from Table 2 are so diverse. For example, to find all the aperiodic tiling rhythms related to $n = 2, 3, 6, 8, 12, 24, 36, 48, 72$.

Table 1. List of rhythms tested and number of tiling complements.

| $n$ | $R_A$ | $R_B$ | No. of $A$'s | No. of $B$'s |
|-----|-------|-------|--------------|--------------|
| 72  | (2, 8, 9, 18, 72) | (3, 4, 6, 12, 24, 36) | 6(2) | 3(1) |
| 108 | (3, 4, 12, 27, 108) | (2, 6, 9, 18, 36, 54) | 252(30) | 3(1) |
| 120 | (2, 5, 8, 10, 15, 30, 40, 120) | (3, 4, 6, 12, 20, 24, 60) | 18(4) | 8(2) |
| 120 | (2, 3, 6, 8, 15, 24, 30, 120) | (4, 5, 10, 12, 20, 40, 60) | 20(3) | 16(5) |
| 144 | (2, 8, 9, 16, 18, 72, 144) | (3, 4, 6, 12, 24, 36, 48) | 36(10) | 6(1) |
| 144 | (4, 9, 16, 18, 36, 144) | (2, 3, 6, 8, 12, 18, 24, 48, 72) | 6(2) | 12(9) |
| 144 | (4, 9, 16, 18, 36, 144) | (3, 4, 6, 8, 12, 24, 36, 48, 72) | 48(7) | 6(1) |
| 144 | (2, 9, 16, 18, 36, 144) | (3, 4, 6, 8, 12, 24, 36, 48, 72) | 12(2) | 156(9) |
| 168 | (2, 7, 8, 14, 21, 42, 56, 168) | (3, 4, 6, 12, 24, 28, 84) | 54(8) | 16(3) |
| 168 | (2, 3, 6, 8, 12, 24, 42, 168) | (4, 7, 12, 14, 28, 56.) | 42(4) | 104(15) |
| 180 | (3, 4, 5, 12, 15, 20, 45, 60, 180) | (2, 6, 9, 10, 18, 30, 36, 90) | 2052(136) | 8(2) |
| 180 | (2, 5, 9, 10, 18, 20, 45, 90, 180) | (3, 4, 6, 12, 15, 30, 36, 60) | 96(12) | 6(1) |
| 180 | (3, 4, 9, 12, 36, 45, 180) | (2, 5, 6, 10, 15, 18, 20, 30, 60, 90) | 1800(171) | 16(5) |
| 180 | (2, 4, 9, 18, 20, 36, 180) | (3, 5, 6, 10, 12, 15, 30, 45, 60, 90) | 120(18) | 9(2) |
Table 2. Comparison of runtimes (in seconds) of the cutting sequential algorithm (CSA) and the fill-out procedure (FP) needed to find all rhythms tiling with a given aperiodic A or B.

| n   | RA            | RB            | CSA A | FP A | CSA B | FP B |
|-----|---------------|---------------|-------|------|-------|------|
| 72  | {2, 8, 9, 18, 72} | {3, 4, 6, 12, 24, 36} | 0.10  | 1.59 | 0.02  | 0.33 |
| 108 | {3, 4, 12, 27, 108} | {2, 6, 9, 18, 36, 54} | 7.84  | 896.06 | 0.03  | 0.72 |
| 120 | {2, 5, 8, 10, 15, 30, 40, 120} | {3, 4, 6, 12, 20, 24, 60} | 0.27  | 24.16 | 0.07  | 2.13 |
| 120 | {2, 3, 6, 8, 15, 24, 30, 120} | {4, 5, 10, 12, 20, 40, 60} | 0.14  | 19.92 | 0.15  | 3.30 |
| 144 | {2, 8, 9, 16, 18, 72, 144} | {3, 4, 6, 12, 24, 36, 48} | 2.93  | 83.53 | 0.06  | 3.77 |
| 144 | {4, 9, 16, 18, 36, 144} | {2, 3, 6, 8, 12, 18, 24, 48, 72} | 0.10  | 7.13  | 1.71  | 66.27 |
| 168 | {3, 4, 6, 12, 18, 24, 36, 48, 72} | {2, 3, 6, 8, 12, 24, 36, 48, 72} | 0.11  | 12.13 | 1.71  | 74.78 |
| 168 | {2, 7, 8, 14, 21, 42, 56, 168} | {3, 4, 6, 12, 24, 28, 84} | 17.61 | 461.53 | 0.13  | 7.91 |
| 168 | {2, 3, 6, 8, 21, 24, 42, 168} | {4, 7, 12, 14, 28, 56, 84} | 0.91  | 46.11 | 1.94  | 35.36 |
| 180 | {3, 4, 5, 12, 15, 20, 45, 60, 180} | {2, 6, 9, 10, 18, 30, 36, 90} | 1422.09 | > 3600.00 | 0.25  | 1243.06 |
| 180 | {2, 5, 9, 10, 18, 20, 45, 90, 180} | {3, 4, 6, 12, 15, 30, 36, 60} | 48.04  | 900.75 | 0.11  | 8.22 |
| 180 | {3, 4, 9, 12, 36, 45, 180} | {2, 5, 6, 10, 12, 30, 36, 60, 90} | 492.18 | > 3600.00 | 0.18  | 7.51 |
| 180 | {2, 4, 9, 18, 20, 36, 180} | {3, 5, 10, 12, 15, 30, 45, 60, 90} | 8.82  | 28.072 | 0.29  | 14.34 |

Table 3. Number of some Vuza canons with periods analyzed by Jedrzejewski (2013).

| n   | P_1 | N_1 | P_2 | N_2 | N_3 | No. of A’s | No. of B’s | CSA A’s | CSA B’s |
|-----|-----|-----|-----|-----|-----|------------|------------|---------|---------|
| 200 | 2   | 2   | 5   | 5   | 2   | 60         | 125        | 2.74    | 3.72    |
| 216 | 2   | 2   | 3   | 9   | 2   | 72         | 729        | 2.85    | 224.30  |
| 240 | 2   | 4   | 3   | 5   | 2   | 200        | 32         | 19.89   | 1.50    |
| 252 | 7   | 2   | 3   | 3   | 2   | 624        | 9          | 1787.98 | 0.29    |
| 264 | 2   | 2   | 11  | 3   | 2   | 558        | 40         | 1289.36 | 1.30    |
| 280 | 2   | 2   | 7   | 5   | 2   | 180        | 480        | 35.24   | 35.31   |
| 300 | 3   | 2   | 5   | 5   | 2   | 240        | 125        | 79.07   | 4.61    |
| 300 | 2   | 3   | 5   | 5   | 2   | 480        | 125        | 161.90  | 6.96    |
| 400 | 2   | 4   | 5   | 5   | 2   | 2040       | 250        | 5318.88 | 37.42   |
| 450 | 3   | 3   | 5   | 5   | 2   | 1920       | 375        | 9769.25 | 44.40   |

\[ R_A = \{3, 4, 12, 27, 108\} \] in \( \mathbb{Z}_{108} \) our algorithm takes less 8 seconds, while the FP takes almost 900 seconds.

Finally, we run our algorithm on the instances considered by Jedrzejewski (2013). In Table 3, we report the number of tiling complements along with the runtimes.

The tail effect

Every time we find a solution, we have to add new constraints to the master problem and solve it again. As a result, the problem gets computationally harder at each iteration.

In Figure 1, we report the time required to find the next tiling solution for two rhythms in \( \mathbb{Z}_{180} \). As expected, the time needed for each iteration grows.

Verifying the tiling property

We are now interested in determining whether a given rhythm A admits an aperiodic tiling complement B. By pairing our model with a function that builds a non-(T2) rhythm A, we could find a counterexample to the necessity of condition (T2), if it exists. Consequently, being able to verify the tiling property of a rhythm A in a reasonable amount of time is essential.
In Table 4, we report the rhythms tested with our method. Since our algorithm is not able to find a complement for the particular rhythms we chose, we are still unable to determine whether (T2) is necessary or not. However, we notice that the runtimes required to determine the non-existence of an aperiodic complement vary in a range from 1 minute (for the rhythms in $\mathbb{Z}_{1050}$, $\mathbb{Z}_{2310}$, and $\mathbb{Z}_{6300}$) up to 10 minutes (for the rhythm in $\mathbb{Z}_{27225}$). The short runtimes of the algorithm make it conceivable to use our algorithm to check whether a rhythm could be a possible counterexample to the (T2) conjecture.
Table 4. Non-(T2) candidate rhythms checked.

| n    | Rhythm tested                                                                 |
|------|-------------------------------------------------------------------------------|
| 1050 | {0, 15, 30, 35, 45, 60, 75, 90, 105}                                          |
| 2310 | {0, 5, 6, 10, 12, 18, 24, 26, 30, 31, 36}                                     |
| 6300 | {0, 2, 4, 5, 6, 7, 8, 10, 12, 35, 352, 354, 355, 356, 357, 358, 360, 362}      |
| 27225| {0, 9, 15, 18, 24, 27, 30, 36, 39, 45, 54, 3025, 3034, 3040, 3043, 3049, 3052, |
|      | 3055, 3061, 3064, 3070, 3079, 6050, 6059, 6065, 6068, 6074, 6077, 6080, 6086, |
|      | 6089, 6095, 6104}                                                           |

5. Conclusions and future work

We introduced a new integer linear model to find the aperiodic complements of a given rhythm. We ran several tests to prove the time efficiency of our method, especially when it comes to determining if there exists an aperiodic complement of a given rhythm. Indeed, Auricchio et al. (2022) proved that our model is very effective to compute tiling rhythmic canons: they showed that SAT (satisfiability) encoding is the best way to solve the ILP problem we introduced in Section 3.

Our future aim is to characterize the polynomial induced by a rhythm that does not satisfy condition (T2) through a linear programming model. It could lead to discovering insightful information on the structure of those canons. We would also like to further improve our algorithm by partitioning the solutions set into smaller disjoint sets. This would allow us to further decrease the runtimes by parallelizing the algorithm.

Disclosure statement

The authors report there are no competing interests to declare.

ORCID

Gennaro Auricchio  http://orcid.org/0000-0002-4285-8887
Luca Ferrarini  http://orcid.org/0000-0002-8903-2871
Greta Lanzarotto  http://orcid.org/0000-0001-9743-0573

References

Amiot, E. 2004. “Why Rhythmic Canons are Interesting.” Perspectives of Mathematical and Computer-Aided Music Theory, 190–209.
Amiot, E. 2005a. “À Propos Des Canons Rythmiques.” Gazette des Mathématiciens 106: 43–67.
Amiot, E. 2005b. “Rhythmic Canons and Galois Theory.” Colloquium on Mathematical Music Theory, Grazer Mathematische Berichte 347: 1–21. (Graz, Austria).
Amiot, E. 2011. “Structures, Algorithms, and Algebraic Tools for Rhythmic Canons.” Perspectives of New Music 49 (2): 93–142. doi:10.7757/persnewmusi.49.2.0093.
Andreatta, M. 2004. “On Group-theoretical Methods Applied to Music: Some Compositional and Implementational Aspects.” Perspectives in Mathematical and Computational Music Theory, 169–193.
Auricchio, G., L. Ferrarini, S. Gualandi, G. Lanzarotto, and L. Pernazza. 2022. “A SAT Encoding to Compute Aperiodic Tiling Rhythmic Canons.” In Integration of Constraint Programming, Artificial Intelligence, and Operations Research, 14–23. Springer International Publishing. doi:10.1007/978-3-031-08011-1_2.
Conforti, M., G. Cornuéjols, and G. Zambelli. 2014. Integer Programming, Graduate Texts in Mathematics. Springer International Publishing, Germania.
Coven, E. M., and A. Meyerowitz. 1999. “Tiling the Integers with Translates of One Finite Set.” *Journal of Algebra* 212 (1): 161–174. doi:10.1006/jabr.1998.7628.
de Bruijn, N. G. 1953a. “On the Factorization of Finite Abelian Groups.” *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A: Mathematical Sciences* 61 (3): 258–264.
de Bruijn, N. G. 1953b. “On the Factorization of Cyclic Groups.” *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A: Mathematical Sciences* 61 (4): 370–377.
Fidanza, G. 2007. “Canoni Ritmici a Mosaico.” Master’s thesis, Università degli Studi di Pisa.
Gilmore, P. C., and R. E. Gomory. 1961. “A Linear Programming Approach to the Cutting-stock Problem.” *Operations Research* 9 (6): 849–859. doi:10.1287/opre.9.6.849.
Gurobi Optimization LLC, 2021. 2021. *Gurobi Optimizer Reference Manual*. Available online at: www.gurobi.com.
Hajós, G. 1950. “Sur La Factorisation Des Groupes Abéliens.” *Časopis Pěst. Mat. Fys* 74: 157–162.
Jedrzejewski, F. 2006. *Mathematical Theory of Music, Collection Musique/sciences*. Ircam-Centre Pompidou, Parigi.
Jedrzejewski, F. 2013. “Enumeration of Vuza Canons.” *arXiv preprint arXiv:1304.6609*.
Kolountzakis, M. N., and M. Matolcsi. 2006. “Complex Hadamard Matrices and the Spectral Set Conjecture.” *Collectanea Mathematicae* 57 (Extra): 281–291. doi:10.48550/arXiv.math/0411512.
Kolountzakis, M. N., and M. Matolcsi. 2009. “Algorithms for Translational Tiling.” *Journal of Mathematics and Music* 3 (2): 85–97. doi:10.1080/17459730903040899.
Łaba, I., and I. Londner. 2022. “The Coven-Meyerowitz Tiling Conditions for 3 Odd Prime Factors.” *Inventiones Mathematicae*. (Springer). doi:10.1007/s00222-022-01169-y.
Land, A. H., and A. G. Doig. 1960. “An Automatic Method of Solving Discrete Programming Problems.” *Econometrica* 28 (3): 497–520. doi:10.2307/1910129.
Lanzarotto, G., and L. Pernazza. 2022. “Extended Vuza Canons.” In *Mathematics and Computation in Music*, 112–126. Springer International Publishing. doi:10.1007/978-3-031-07015-0_10.
Marchand, H., A. Martin, R. Weismantel, and L. A. Wolsey. 2002. “Cutting Planes in Integer and Mixed Integer Programming.” *Discrete Applied Mathematics* 123 (1-3): 397–446. doi:10.1016/S0166-218X(01)00348-1.
Rédei, L. 1950. “Ein Beitrag Zum Problem Der Faktorisation Von Endlichen Abelschen Gruppen.” *Acta Mathematica Academiae Scientiarum Hungarica* 1: 197–207. doi:10.1007/BF02021312.
Sands, A. D. 1974. “On the Factorisation of Finite Abelian Groups, III.” *Acta Mathematica Academiae Scientiarum Hungarica* 25: 279–284. doi:10.1007/BF01886085.
Tijdeman, R. 1995. “Decomposition of the Integers as a Direct Sum of Two Subsets.” In *Number Theory: Paris 1992–3*, London Mathematical Society Lecture Note Series, 261–276. Cambridge University Press.
Vuza, D. T. 1991–1993. “Supplementary Sets and Regular Complementary Unending Canons (Part One, Two, Three, Four).” *Perspectives of New Music*, (29(2), 30(1), 30(2), 31(1)), (22–49, 184–207, 102–124, 270–305). (https://doi.org/10.2307/833429; https://doi.org/10.2307/833290; https://doi.org/10.2307/3090628; https://doi.org/10.2307/833054)
Wolsey, L. A. 1998. *Integer Programming, Wiley Series in Discrete Mathematics and Optimization*. Wiley, Germany.