

Article

Assistive Model to Generate Chord Progressions using Genetic Programming with Artificial Immune Properties

María Navarro-Cáceres ^{1,*}, Javier Félix Merchán Sánchez-Jara ²,
Valderi Reis Quietinho Leithardt ^{3,4} and Raúl García-Ovejero ¹

¹ Department of Computer Sciences, University of Salamanca, Pza de los Caídos s/n., 37007 Salamanca, Spain; raulovej@usal.es

² Department of Musicology, University of Salamanca, Pza de Escuelas Menores s/n., 37007 Salamanca, Spain; javiermerchan@usal.es

³ COPELABS, Universidade Lusófona de Humanidades e Tecnologias, 1749-024 Lisbon, Portugal; valderi@ippportalegre.pt

⁴ VALORIZA, Research Center for Endogenous Resources Valorization, Instituto Politécnico de Portalegre, 7300-555 Portalegre, Portugal

* Correspondence author: maria90@usal.es

Received: 18 July 2020; Accepted: 26 August 2020; Published: 31 August 2020



Abstract: In Western tonal music, tension in chord progressions plays an important role in defining the path that a musical composition should follow. The creation of chord progressions that reflects such tension profiles can be challenging for novice composers, as it depends on many subjective factors, and also is regulated by multiple theoretical principles. This work presents ChordAIS-Gen, a tool to assist the users to generate chord progressions that comply with a concrete tension profile. We propose an objective measure capable of capturing the tension profile of a chord progression according to different tonal music parameters, namely, consonance, hierarchical tension, voice leading and perceptual distance. This measure is optimized into a Genetic Program algorithm mixed with an Artificial Immune System called Opt-aiNet. Opt-aiNet is capable of finding multiple optima in parallel, resulting in multiple candidate solutions for the next chord in a sequence. To validate the objective function, we performed a listening test to evaluate the perceptual quality of the candidate solutions proposed by our system. Most listeners rated the chord progressions proposed by ChordAIS-Gen as better candidates than the progressions discarded. Thus, we propose to use the objective values as a proxy for the perceptual evaluation of chord progressions and compare the performance of ChordAIS-Gen with chord progressions generators.

Keywords: chord progression; tonal tension; tonal interval space; artificial immune systems; genetic programming

1. Introduction

In Western tonal music, tension is the anticipation a musical composition creates in a listener's mind for relaxation or release. The tonal tension is a perceptive phenomenon that can be produced by the combination of different musical properties, such as dynamics, reiteration or rhythm. The balance between tension and relaxation in a composition is usually denominated tonal tension profile, which can be represented with a curve. The movement between tense music and relaxed music keeps the music moving forward. Therefore it is an essential element for the composers to capture the receptor's attention while they are listening.

The tonal tension profile usually depends on the different musical elements that are part of the musical fragment. One of the most important elements are chord progressions [1,2]. Chord progressions consist of sequences of chords (a set of notes that can be played together) that are played throughout the music. Over the past decades, there are some theoretical proposals that aimed to regulate tension of chord progressions through their particular elements, such as consonance, musical tension and voice leading. Riemann [3], Schoenberg [4], and Schenker [5], among others, have discussed these principles extensively and proposed rules to create chord progressions.

In music composition, creating chord progressions which reflects a tonal tension profile commonly requires years of musical training. To make this process easier, technological solutions have been proposed to automatically model and analyze chord progressions, following different paradigms including statistical learning [6–9], rules [10–13], grammars [14,15], and biological principles [16–19].

In particular, genetic programming, genetic algorithms and artificial immune systems have been applied successfully to this field to create tools that can assist users into generating chord progressions [20]. One of the first works to apply evolutionary computation [16] proposed Vox Populi which uses genetic algorithms to evolve a population of chords by maximizing multiple musical criteria and following some users' preferences. Mcdermott et al. [21] propose a genetic algorithm that creates music represented by graphs. Sciera et al. [22] present a meta-composer, capable of creating chord progressions based on a hybrid genetic algorithm. Fukumoto et al. [17] apply genetic algorithms to automatically create chord progressions according to the user's feelings. Herremans et al. [23] propose an expert system to compose counterpoint based on different optimization functions. More recently, we [24] proposed ChordAIS, a system that makes use a standard Artificial Immune System (AIS) called opt-ainet to iteratively generate the next chord in a given sequence by optimizing an objective function. Although all of them obtained positive results, the mentioned proposals do not follow or capture tonal tension profiles or curves. For such purposes, the proposals would require a hierarchical analysis of the whole chord progression.

To address this limitation, several authors propose new works in which the tonal properties of a chord are measured by considering not only on the previous chords but also their hierarchical relationships, typically represented as a tree structure [25]. Granroth et al. [26] propose a system that generates chord progressions by learning the most common structures from an annotated corpus. It works in a jazz context, thus it does not follow western tonal principles and they did not apply bioinspired algorithms. Herremans et al. [27] present a chord generator to create progressions according to a tension profile. The authors use hybrid optimization with a function based on the spiral array representation. However, the work does not analyze how the properties of chords can influence the selection of a chord progression or another, and, although they applied a bio-inspired algorithm with successful results, it is not focused to help a novice composer or a user.

In this work, we propose a new system called ChordAIS-Gen, that assists the user to generate chord progressions by applying genetic programming (GP). To improve the performance of GP, we incorporate some properties derived from the Artificial Immune System [28]. At last, the system is capable of offering chord progressions with different musical properties but adapted to a tonal tension profile given as an input. To evaluate the musical features of chord progressions, we depart from the conceptual basis of Lerdahl's model and computationally generate an objective function that captures this tonal tension profile. In this work, tonal pitch indicators inspired by Lerdahl's model are computed using the Tonal Interval Space (TIS) by Bernardes et al. [13], a multidimensional space based on the discrete Fourier transform, where hierarchical tonal pitch relations are expressed as distances.

It is important to note that the application of the GP joined to an AIS has not been deeply explored. We aim to exploit the GP properties to create hierarchical structures that govern the chord progressions. Likewise, we also aim to take advantage of the immune properties of the AIS to generate multiple solutions different from one another. Therefore, while most state-of-the-art approaches propose a unique solution to the generation of chord progressions, the system proposed in this work is capable of proposing multiple different and good quality candidate chords. Furthermore, the lack

of hierarchical structure has been a long identified problem in the field of generative music [29]. ChordAIS-Gen addresses both linear and hierarchical dependencies of chord progressions and also aims to improve the accuracy of tonal indicators by adopting the perceptually-inspired TIS.

We carried out different experiments to evaluate the chords progressions created by performing a usability test with users. We also evaluated if the algorithm developed can get good solutions for our particular problem. In this case, we used a listening test to validate the candidates proposed by our system. We compared our model with two previous models: Lerdahl's proposal [30] and ChordAIS, a previous work [24]. Lerdahl's proposal captures similar musical properties as in this present work, but in a different topological space. With that comparison we aim to know if TIS is a good solution to capture musical properties. ChordAIS uses TIS to compute musical properties. However, the features captured differ from our work. ChordAIS measures the consonance and relatedness to the previous chord, key and harmonic function to create a chord progression. With this comparison we aim for the inclusion of new musical properties in the measure improves the results.

Additionally, we compared the performance of our algorithm against the classical version of GP and AIS. Finally, we aim to validate if the function can capture the tonal tension profile of a progression. Therefore, we created a listening test and asked the participants to select and rate the curve that fits best with the tonal tension profile. The correlation between the association of the curve made by the system and the subjective ratings will support our final conclusions.

This paper is organized as follows: Section 2 reviews the topological space used to represent music in this work and the musical paradigm followed. Section 3 describes the overall system with the integration of the fitness function and the bioinspired algorithm, along with the general workflow of ChordAIS-Gen. Section 4 details the musical properties captured by the function, while Section 5 exposes the technical information of the algorithm proposed. In Section 6, we give information about the experiments performed to evaluate our proposal, followed by the results and a discussion. Finally, Section 7 presents the conclusions and discusses future work.

2. Theoretical Background About Chord Progressions

2.1. Tonal Interval Space

The representation used to encode the notes plays an important role in the automatic generation of chord progressions. Representations where neighborhood relations among the notes reflect perceptual or music theory properties can be particularly useful, such as space where chords that are close together generate better transitions than chords that are farther apart. There have been proposals to organize the musical notes in geometrical spaces according to musical intervals and other musical properties. In these spaces, geometrical properties such as neighborhood reflect musical properties such as perceptual relatedness. The Tonnetz [31] is an early example of such a geometrical representation of the properties of the tonal system because small distances reflect close harmonic relations. Later, Chew [32] proposed a three-dimensional helix model to represent tonal pitch relationships. Harte et al. [33] mapped 12-bin chroma vectors to the interior of a six-dimensional polytope where close harmonic relations appear as small distances. Bernardes et al. [13] extended the latter space, toward better adapted perceptual pitch relations by computing all kind of intervals as the weighted discrete Fourier transform (DFT) of chroma vectors. In the Tonal Interval Space (TIS) [13], distances among multi-level pitch configurations capture perceptual relatedness and magnitudes indicate consonance. In this work, we propose to use the TIS [13] to represent multi-level pitch configurations such as pitch classes, intervals, chords, and keys because geometrical properties of the space are related to perceptual and musical properties. Tonal interval space has been used in several works with successful results [13,18,24]. Consequently, we decided to make use of this space to capture musical properties.

One of the most important properties of the TIS is that distances among different pitch configurations represented by their TIV reflect musical attributes. The Euclidean distance between two TIV of two different pitch configuration, $T_1(k)$ and $T_2(k)$ is given by

$$\rho(T_1, T_2) = \sqrt{\sum_{k=1}^M |T_1(k) - T_2(k)|^2}. \quad (1)$$

Perceptually, similar vectors result in smaller distances than dissimilar ones.

Another mathematical measure that captures musical properties and is used in the present work is the inner product between $T_1(k)$ and $T_2(k)$, given by:

$$T_1 \cdot T_2 = \sum_{k=1}^M T_1(k) \overline{T_2(k)}, \quad (2)$$

where $\overline{T_2(k)}$ is the complex conjugate of $T_2(k)$. Equation (2) yields higher values for perceptually similar vectors than dissimilar ones. The angle θ between T_1 and T_2 can be calculated from Equation (2) as

$$\theta(T_1, T_2) = \arccos\left(\frac{T_1 \cdot T_2}{\|T_1\| \|T_2\|}\right), \quad (3)$$

where $\|T_1\|$ denotes the magnitude of T_1 calculated as the distance $\rho(T_1, T_0)$ between T_1 and the center of the space T_0 using Equation (1). Equation (3) results in smaller angles to indicate a higher degree of similarity.

Equations (1)–(3) can be used to compute distances between pitch configurations of the same level (e.g., the distance between two chords) or across levels (e.g., the distance between a chord and a key). For example, the distance between two chords captures their tonal relatedness, the distance between a chord and a key measures the level of membership of the chord to the key, and the distance of a given chord to the three categorical harmonic functions (represented in the space as the triads of the tonic, subdominant and dominant degrees) measures how well the chord fits the categorical harmonic functions. All these properties are used in the proposed measure (Section 4) to capture the suitability of the chord in the context of a progression.

2.2. Lerdahl's Theory to Model the Tonal Fitness of Chord Progressions

In the literature, some authors have suggested different models of structural dependencies of chord progressions [25,34,35]. Lerdahl proposed an outstanding model [25] which consists of an evaluation of the melodic attraction and the tonal tension of a chord in a progression. Lerdahl's proposal analyzes the purpose of a chord inside the architecture of a chord progression. Besides, it considers four factors: tonal pitch distance, surface dissonance, voice leading and hierarchical structure.

Tonal pitch distance grasps the closeness of chords via an algebraic representation. It presents a space in order to evaluate chord distances through a reference table. We contemplate, in this table, the non-common tones between key distance, chords, and the interval distance amidst the chords root [36].

Surface dissonance assesses the psychoacoustic dissonance of chords in a progression. It measures the dissonance from the interaction within the vertical notes of the chords. Three elements constitute surface dissonance: "inversion", "nonharmonic tone" and "scale degree". "Inversion" explains the chord's bass note and on its upcoming construction as root position or inversion; "nonharmonic tone" explores the presence of tones outside the chord function; and "scale degree" examines the component scale degrees in a specific chord.

Voice leading grasps the melodic attraction between successive chords per voice. It is a horizontal assessment that evaluates the suitability of each note in a voice in accordance with the preceding note. In theory, the balance of the notes, the number of semitones and the whole fitness between the chords

are able to modify the assessment of the voices [36,37]. Lerdahl [36] points out that the balance of each note is linked with the distance to the key or tonal center. The nearer to the key, the steadier the note will be.

Hierarchical structure grasps the diverse levels of the musical building by picking tree-based structures conducted from functional harmonic categories. It can be segmented into local tension and inherent tension to find out the hierarchical relation of the chords. Local tension determines the distance between the chord we choose to assess and the chord right above it in the tree, which is named “parent chord”. Furthermore, all the distances between the parent chord and the chords above are considered as inherent tension.

This model has been applied in different works to measure the tonal tension of a musical composition [25,38,39]. Due to the successful results obtained in the literature, we selected this model and adapt it to our proposal.

2.3. Rohrmeier’s Hierarchical Tree Structure

Rohrmeier implements concepts from theories concentrated in the hierarchical dimension of music through a binary grammar that follows tonal rules in order to create chord progressions and explicitly designed. It allows it to be computationally verifiable and achievable [35]. This hierarchical model of tonal music leans on four items. The functional level is the most important here. Therefore, this grammar is utilized for this work to evaluate the hierarchical structure.

The behavior of a grammar is identified by the rules in Equations (4a)–(4g) according to Rohrmeier. This behavior constitutes the functional regions in a chord progression. Besides, the rules include two kinds of symbols: terminal symbols portrayed by lower-case letters and non-terminal symbols portrayed by capital letters. The non-terminal symbols describe functional sections, for example, the dominant region DR, the subdominant region SR, the tonic region TR and any of the preceding regions XR. Non-terminal symbols enable the spread into different harmonic regions sticking to the rules in Equations (4a)–(4d) or into terminal symbols following Equations (4e)–(4g). Dominant, subdominant or tonic chords are portrayed by the terminal symbols t , d , s in the sequence respectively. However they cannot be expanded further.

$$TR \longrightarrow TR \quad DR \quad (4a)$$

$$TR \longrightarrow DR \quad t \quad (4b)$$

$$DR \longrightarrow SR \quad d \quad (4c)$$

$$XR \longrightarrow XR; \quad XR \quad \forall \text{ non-terminal} \quad (4d)$$

$$TR \longrightarrow t \quad (4e)$$

$$DR \longrightarrow d \quad (4f)$$

$$SR \longrightarrow s \quad (4g)$$

3. System Overview

According to musical standards, a chord is a set of three or more notes that are played together. When you play several chords, you generate a chord progression. The structure of a chord progression is a hierarchy that can be represented with a tree. The idea of chord progression and how to encode the tree structure is illustrated in Figure 1.

Firstly, the tree is constructed following some instructions that, in this work, are proposed by Rohrmeier’s theory. The first node in Figure 1 is a tonic region TR and it determines the key of the chord progression. Secondly, Rule (4b) is applied to get the nodes in level 2. Next, TR derives into t (Rule (4e)) and the second child employs Rule (4b). The last levels are obtained by applying Rules (4c), (4f) and (4g). Finally, each leaf node is replaced by a chord that complies with the same harmonic function (tonic, subdominant or dominant). Initially, in the algorithm, this replacement is made randomly.

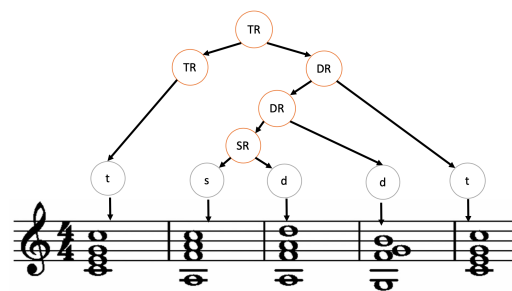


Figure 1. Illustration of a chord progression.

In this work, we must construct chord progressions that follow a concrete tonal tension profile given by the user. The tonal tension profile can be represented with a curve and usually depends on the musical features of the chords and the tree structure associated with the chord progression. The difficulty lies in capturing such subjective properties through a mathematical model. To address this issue, the chords will be encoded as points in topological spaces such as TIS, capable of capturing musical properties [13], and we apply some mathematical measures to capture all these musical properties. The four musical properties captured are: perceptual distance δ , dissonance ζ , hierarchical tension λ and *voice leading* ϕ . All of them are linearly combined in an objective function M . This function will give a score to each chord according to their musical properties. If you join the set of scores given by M to the chords that are part of the progression, you can generate a curve that represents the tonal tension profile of a chord progression. To know if the tonal tension profile of a chord progression adapts to a curve given by the user, we calculate the regression R^2 between both curves. Higher values of R^2 will be translated into chord progressions that adapt to the tonal tension profile that the user is looking for. Therefore, we are looking for an optimization of R^2 . Due to the wide amount of chord progressions that can be created manually, we need a heuristic algorithm to look for different optima according to the regression function, in this case, we used GP with some immune properties that improve its performance to create chord progressions.

Consequently, in our work the selection of the chord progressions can be considered as a search problem in a geometrical space (in this case TIS [13]) and subjected to musical constraints. The regression R^2 between the points M and the initial curve drawn by the user will constitute the fitness function of our bioinspired algorithm.

The regression function R^2 is the fitness of our bioinspired algorithm which, along with the constructed interface, constitutes a system that presents multiple options so that the user can choose among these chord progressions according to his/her preferences. Figure 2 shows an overview of ChordAIS.

ChordAIS-Gen workflow is illustrated in Figure 2. In a first step, the user sets some initial parameters, such as the key, the number of chords and the tonal tension profile of the chord progression. The tonal tension profile is a function that should be drawn by the user, and can have any shape. This information will be stored in ChordAIS-Gen for the subsequent iterations. The bioinspired algorithm generates a population of chord progressions. Each chord progression is represented as a tree, which captures its specific hierarchical structure, which has been created following the grammar proposed by [35]. The leaf nodes will be the chords that are part of the chord progression. According to musical standards, a chord is a set of three or more notes that are played together. In this particular case, the chord is represented as a vector with n elements, and each element is a number between 0 and 128, according to the MIDI standards. We use MIDI because the synthesizers used can understand this codification very easily. However, to measure the tonal tension of each MIDI chord, we encode them into the TIS and to get the tonal tension we apply our function M . The result is a curve or profile whose points are the numbers that each chord has obtained according to M . This curve is compared with the tonal tension profile that the user drew in the previous stage. The population of the chord progressions evolve to obtain individuals whose profile adapts best to the curve proposed by the user.

Finally, ChordAIS-Gen presents the best three minima (chord progressions) it could find to the user, who finally selects which chord progressions should be used according to his/her preferences and can play it through a simple synthesizer.

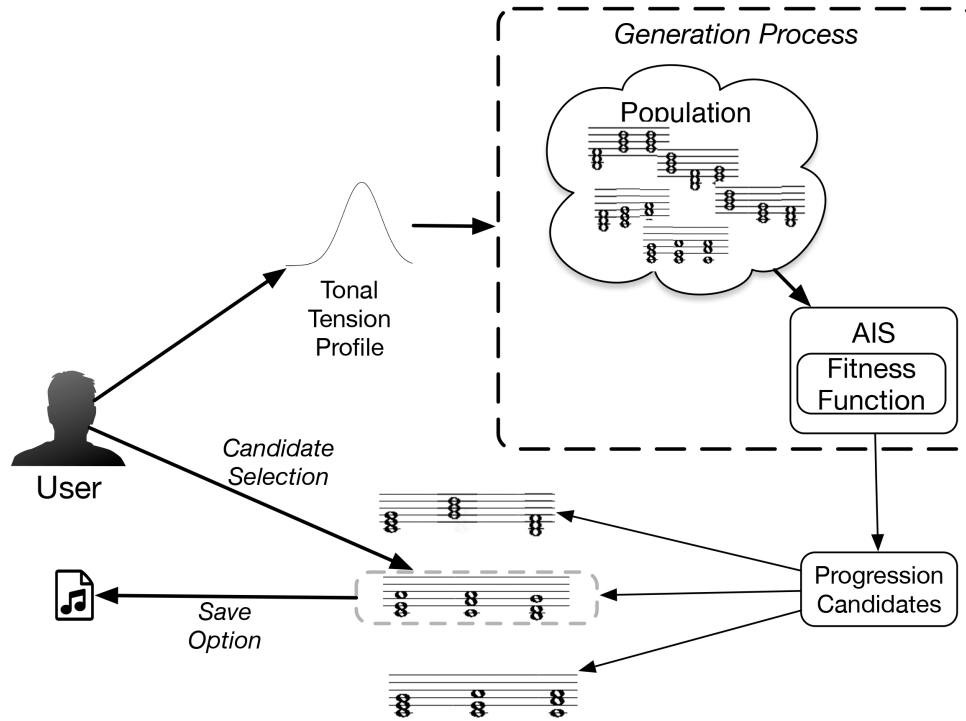


Figure 2. Overview of the approach proposed in this work. We did not draw the trees associated to each chord progression for more clarity.

4. Creating the Tonal Tension Profile of a Chord Progression

The foremost indicator utilized to evaluate the tonal tension of a chord are the properties of the TIS resulting from mathematical assessments like the angle between TIVs or the Euclidean distance [13]. In this work, we attempt to generate a measure that grasps the tonal fitness of a chord via its musical features. We generate a measure in Equation (5) that captures the properties described in Lerdahl’s proposal: voice leading (m), tonal pitch distance (δ), hierarchical structure (h) and surface dissonance (c):

$$M(T_i, P) = k_1\delta(T_i, P) + k_2c(T_i) + k_3m(T_i, P) + k_4h(T_i, P), \quad (5)$$

where k_j are constants that correspond to the weights of each parameter and T_i is the i -th chord of the progression M . The codification of an element proposed in Section 2 is related to Lerdahl’s proposal but encoded in the TIS. The following subsections will detail how the four items are encoded in the TIS in order to be able to assess them mathematically.

4.1. Tonal Pitch Distances

The tonal pitch distance between two sequential chords is encoded by the measure δ . In line with Section 4.1, δ assesses three musical features: distance to the key, distance to the tonal function and distance to the previous chord in the sequence. In Equation (6), δ is divided as:

$$\delta(T, P) = \rho(T_i, T_{i-1}) + \theta(T_i, T_{key}) + \theta(T_i - T_{key}, T_f), \quad (6)$$

where θ uses Equation (3) and $\rho(T_i, T_{i-1})$ utilizes Equation (1). The degree of membership of T_i to the main key of the full chord progression is evaluated by $\theta(T_i, T_{key})$; and the measures of the resemblance to the tonal function designated by the tree created previously is assessed by $\theta(T_i - T_{key}, T_f)$.

Chords musically close to the key have low distances between the key and chord configurations in the TIS. T_{key} represents the key. It is the result of the TIV of the chroma vector $C_{key}(n)$ which includes all the notes of the scale of the key. Besides, the angle between the center of the key T_{key} and the chord T_i is estimated by Equation (3)

In order to assess the closeness of one chord to a tonal function in the TIS, we inspect if the chord T_i is lined up with the dominant V, tonic I or subdominant IV, or by employing the angle measure that is explained in Equation (3). T_f is a vector that corresponds to one of the harmonic functions I, IV, and V referenced by T_{key} and $T_i - T_{key}$ is the chord also using T_{key} as reference. We want to minimize $\theta(T_i, T_{key}, T_f)$ through T_f , sticking to the harmonic sequence created by the tree built in the preceding step, to secure T_i in line with one of them.

4.2. Consonance

The rule $\|T_i\|$ evaluates the consonance c of the chord that is portrayed by T_i , in the TIS [13]. The chord is quite consonant, if c leads to a large value for $\|T_i\|$. Hence, we attempt to maximize $c(T_i)$.

4.3. Voice Leading

The melodic attraction of two successive chords is represented by m . Three factors shaping the melodic attraction are proposed by Lerdahl. These are the number of semitones between the notes, the stability of the notes (to what extent the rest of the notes are attracted) and the perceived distance between the chords. According to these factors, the voice leading measure between T_i and T_{i-1} is codified as

$$m(T_i, P) = \frac{\sum_{l=1}^3 v(n_{l_{i-1}}, n_{l_i})}{\rho(T_i, T_{i-1})}, \quad (7)$$

where ρ is the Euclidean distance from Equation (1) that captures the perceived distance between the chords and v is a measure of voice-leading for each voice l that is generated between the note $n_{l_{i-1}}$ of the preceding chord and the note n_{l_i} of the current chord. We calculate the distance of its correlative pitch class to the key in the TIS in order to measure the stability of the notes of a chord in a progression. We calculate the distance to the key and the number of semitones as

$$v(n_{l_i}, n_{l_{i-1}}) = \rho(T_{n_{l_i}}, T_{key})e^{0.05s} \quad (8)$$

where the number of semitones between n_{l_i} and $n_{l_{i-1}}$ is s

4.4. Hierarchical Tension

According to Lerdahl's theory, the measure of h refers to the tonal tension concept. It calculates distances between chords for a progression considering a tree structure. We calculate the tension linked to the hierarchical structure of the progression as

$$h(T_i, P) = \rho(T_i, T_j) + \frac{\sum_{k=j}^N \rho(T_i, T_k)}{N}, \quad (9)$$

where T_j is the parent chord of T_i , N is the number of chords in the tree and T_k is the parent chord of T_{k-1} based on the tree structure of the chord progression M . Besides, ρ is the Euclidean distance between two chords' encoding which is measured with Equation (1).

Following Rohrmeier's proposal, the hierarchical tree is generated. Nevertheless, it is still important to know the relevance of a chord in the hierarchical structure and, therefore, their intrinsic

tension and local tension. We must recognize which chords dominate (possess more tension) the other chords in order to measure their tension.

The substitution of each node of the tree generated beforehand with the tensest chord, picked from its children, portrays the hierarchy of “dominated” chords. In the current work, a method was implemented to get a tree with the “dominant” chords of each level, and consequently, in order to enable the calculations of the intrinsic and local tensions. At first, the concrete chords substitute the leaf nodes. Automatically, the leaf chord is represented by the parent nodes with just one child. Additionally, the child with the same harmonic function as the parent also automatically portrays the parents with children symbolizing diverse tonal regions. Finally, the most stable chord in conformity, measured as described in Section 4.1, substitutes the parent with children of the same tonal regions.

5. Genetic Programming and Immune Systems

Genetic programming (GP) is a heuristic algorithm that makes use of the chromosome evolution process to solve a problem [40]. GP is specifically designed to evolve individuals which are traditionally represented as tree structures [41]. The algorithm applies different operations to evolve the population and retrieve an optimal solution. Those operations are commonly crossover, mutation and selection. The crossover operation involves exchanging random branches of two trees to produce a new and different individual that is incorporated into the population in a future generation. Mutation involves modification of some properties in an individual. The selection consists of deciding which trees (individuals) will be preserved for the next generation.

GP has been applied to many applications such as robotics engineering, circuit design, art, financial engineering and so on. The nature of this algorithm makes it ideal to generate hierarchical structures that govern the chord progressions. Consequently, we decided to apply GP to optimize M and generate music. Although the traditional GP can only obtain a quasi-optimal solution, for this particular purpose the subjective criterion of the user plays an important role and cannot be predicted by our system. Therefore, it was strongly desired that ChordAIS-Gen searches for multiple solutions simultaneously, to give the user the opportunity to choose among different options.

To address this problem, we looked for similar algorithms that can be fused with GP to improve its behavior. We found that de Castro and Timmis [42] developed an artificial immune system for multi-objective optimization called opt-aiNet. In opt-aiNet, each candidate solution (called antibody) is encoded as a vector whose quality is measured with an objective function. A population of antibodies, which represents a pool of candidate solutions, evolves following the immunological principles of clonal expansion, mutation and suppression. The evolution of the antibodies results in the optimization of the objective function because the objective value (or quality) of the solutions encoded by the antibodies increases. Initially, the antibodies are randomly initialized to explore the search space. Next, some antibodies are selected and cloned based on their quality. Each clone undergoes a mutation inversely proportional to its objective value. These mutated clones are subsequently re-evaluated with the objective function to determine if mutation increased their quality. Those mutated clones with higher quality values than before are introduced in the current population, while the ones whose quality decreased are discarded. In order to maintain diversity in the pool of candidate solutions, the algorithm eliminates those antibodies that are close together in the search space. The affinity between two antibodies is measured with the Euclidean distance between them, such that smaller distances mean greater affinity. Those antibodies whose affinity is higher than a given threshold are removed from the population while keeping the ones with the highest objective value. Finally, a number of newly generated antibodies are incorporated into the network. The algorithm converges when the quality of the current solutions cannot be further improved and no new solutions are found.

We developed a GP incorporating some of the operations proposed in opt-aiNet. The pseudo-code of the algorithm is shown in Figure 3 and can be explained as follows:

1. Randomly initialize a population of tree structures and chords.
2. While the stopping criterion is not met do.

- (a) Determine the fitness of each antibody (chord progression) according to M and normalize the matrix of profiles for the population.
- (b) Generate a number N_c of clones for each antibody (chord progression).
- (c) Mutate each chord progression according to a probability Q_m . The mutation can follow two different procedures:
 - Mutate a leaf node (a chord) proportionally to the fitness of its parent.
 - Mutate a parent node (a part of the tree) proportionally to the fitness of its parent.
- (d) Determine the fitness of all the clones.
- (e) Select the clone with the highest fitness.
- (f) Exchange two branches of different trees according to a probability Q_e . (cross-over operation). The node in which the new branch is inserted will depend on the fitness of the parent.
- (g) Calculate the average fitness of the selected population.
- (h) If the average error of the population is not significantly different from the previous iteration, then continue. Else, return to Step (a)
- (i) Determine the affinity among all antibodies. Suppress all but the highest fitness of those antibodies whose affinities are less than the suppression threshold st and determine the number of antibodies, named memory cells, after suppression.
- (j) Introduce a percentage d of randomly generated antibodies and return to Step 2.

3. End While

The algorithm works as follows. One antibody represents a chord progression. Each progression consists of a tree, which is its representation of the hierarchical structure. Note that the leaf nodes will be the set of chords that will be played in the final step. This tree has been created following Rohrmeier's grammar [35]. The chord progressions are randomly initialized to explore the search space (Step 1) and evaluated to discover their fitness. The fitness is calculated as follows:

- First, for each chord, we measure the tonal tension with M .
- With the number obtained, we join the points to get the tonal tension profile.
- The fitness will be the regression value R^2 between the tonal tension profile and the curve that the user has drawn.

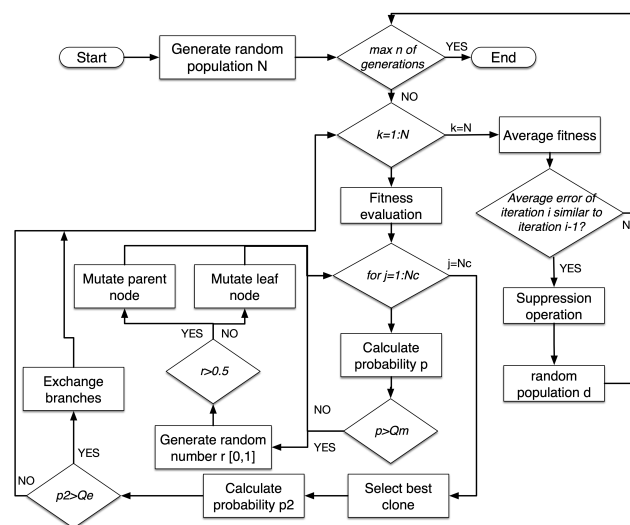


Figure 3. Workflow of the genetic programming (GP) with the artificial immune properties.

Afterwards, some of them are selected and cloned according to the probability Q_m (Step 2.b), which is empirically set. Each clone (chord progression) undergoes a mutation inversely proportional to their fitness (Step 2.c). As the chord progression is represented by a tree, we can mutate a leaf node or a parent node. If we mutate a leaf node (a chord), we have to mutate their individual notes. Note that a chord is represented as a vector, and each note is a number between 0 and 128 (according to the MIDI standards). To mutate a chord, we have to change one or several of these notes, which are simple numbers. To select the number of notes to be mutated, we framed the value of the chord according to M between 0 and n , where n is the number of notes of the chord. The i -note or i -notes to be changed are randomly selected. The mutation value of each i -note of the clone is calculated according to the value of c_i , represented in Equation (10), where $N(0, 12)$ is a random value between 0 and 12, M is the evaluation function and β is a correction factor empirically set.

$$c_i = (1/\beta) \exp(-P) \cdot N(0, 12) \quad (10)$$

In this work, the trees have to be feasible, meaning that all have to follow the rules defined by the grammar. That means when we mutate a parent node, we have to re-structure all the leaf nodes of the branch. Therefore, the higher the position of the parent node, the higher the mutation will be. To select the node of the tree to be mutated, we sum the values of the chords according to M and we framed it between 0 and n_2 , where n_2 is the number of the parent nodes of the tree, decreasingly ordered according to their depth. From this node, we randomly apply new rules and generate new chords until we get the same number of nodes as before.

These mutated clones are subsequently re-evaluated with the objective function to determine if mutation increased their quality (Step 2.d). Those mutated clones with higher quality values than before are introduced in the current population, while the ones whose quality decreased are discarded (Step 2.e).

In the following step, some of the chord progressions exchange branches according to the probability Q_e (Step 2.b), which is empirically set. When a tree t_1 is selected, we randomly select a tree t_2 . To avoid unfeasible trees (which violated rules of the grammar), t_2 must have a branch with the same parent node than the tree t_1 . To select the node of the tree to be exchanged, we sum the values of the chords according to M and we framed it between 0 and n_3 , where n_3 is the number of the parent nodes of the tree t_2 which are the same, decreasingly ordered according to their depth. From this node, we exchange the branches of both trees.

In order to maintain diversity in the pool of candidate solutions, the algorithm eliminates those antibodies that are close together in the search space. The affinity between two antibodies is measured with the Euclidean distance between them, such that smaller distances mean greater affinity. Those antibodies whose affinity is higher than a given threshold are removed from the population while keeping the ones with the highest objective value (Step 2.g). Finally, a number of newly generated antibodies are incorporated into the network (Step 2.h). The algorithm converges when the quality of the current solutions cannot be further improved and no new solutions are found.

6. Evaluation

The aim of the evaluation is threefold. Firstly, we aim to validate the objective function M we designed as a good measure that captures the tonal tension profile. Secondly, we aim to validate the use of GP with immune properties in the overall system as a good solution, comparing it with other proposals. Finally, we aim to determine the usability of ChordAIS-Gen through the users' experience.

Firstly, we developed a prototype of our algorithm by using MATLAB tool. MATLAB was selected because it can be successfully utilized for bioinspired and evolutionary computation and also because it can be run on multiple platforms. We created a basic interface where the user can draw a curve for the tonal tension. Our specific proposal of GP was implemented in MATLAB version 2019a using standard libraries for common operations such as creating random numbers, plotting curves and

calculating the fitness with the correlation values. Due to the specific features of the algorithm (which include AIS and GP properties), it was programmed entirely without any external library specifically designed for genetic programming.

According to the theoretical description presented above, in our algorithm, the following parameters must be set:

- st : suppression threshold
- N : initial population size
- N_c : number of clones per antibody
- β : range of the affinity proportional mutation
- Q_m : Probability of mutation
- Q_e : Probability of cross-over
- d : percentage of random population.

Empirically, the parameters used were set as follows: $st = 0.001$; $N = 200$; $N_c = 5$; $\beta = 2.6$; $Q_m = 0.6$; $Q_e = 0.4$; $d = 10\%$.

Note that the evaluation aims to demonstrate that M reflects the tonal tension of the chord progression. Therefore, we need to evaluate M in optimal conditions. That means we need to find the best values for weights k_j in Equation (5) by applying cross-validation. A previous work, [43] has demonstrated the independence of the hierarchical properties of the chord progressions, independently of the key. Additionally, we used cross-validation to calculate the weights k_i that best fit the scores resulting from a listening test with a total of 20 chord progressions. The final weights obtained were $k_1 = 4.22$, $k_2 = 2.13$, $k_3 = 2.06$, $k_4 = 3.76$.

The evaluation remains a challenge in music generation with expert systems because of the inherent subjectivity of the final product [44]. Most works in the literature use a human expert [12,23,45] or a group of human listeners [11,46,47] to evaluate the results. They usually design a listening test and a comparative study between the present system and previous works, with successful results. Therefore, in this work, we will also perform a listening test, followed by statistical analysis and a comparative study between other proposals.

6.1. Evaluating the Objective Function

Firstly, we address the validation of the objective function M as a proxy for the perceptual evaluation of tonal tension profile in chord progressions. We expect the objective function to reflect the perceptual quality of the chord progressions independently of the tree structure associated.

We designed a listening test that presents several chord progressions generated automatically. We created a total of 12 chord progressions with a total between 5 and 8 chords each one. Each chord progression had a different tree structure. All of them were in major or minor keys. We used a total of four different keys. The rhythm in which the chords are played was always regular to put the focus on the harmonic properties instead of in rhythmic features. The chord progressions were generated by our system and all of them follow some of the curves presented as options in Figure 4. Each chord progression was presented with a reproduction bar, and they were synthesized with a standard piano. The presentation of each chord progression in the listening test is shown Figure 4.

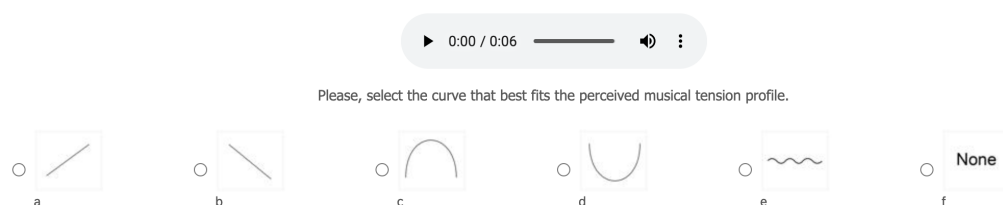


Figure 4. Question of the listening test.

The listeners were presented the chord progressions in a random order. The progressions could be played as many times as they need. The listeners are given the options in Figure 4, and they have to select which curve adapts best to the tonal tension profile of the chord progression. In total, 38 people took the test, among which 11 declared no musical training, 10 considered themselves amateurs, and 17 professional musicians.

We grouped the results according to the curves that the chord progressions represented. To analyze the results we calculated the hit rate of the subjective ratings versus the fitness values for the chord progressions included in the listening test. Note that the fitness value is the correlation between the curve expected by the system and the curve obtained evaluating M for all the chords. In general, chord progressions with higher fitness values have a higher hit rate for all curves tested.

We also aim to demonstrate that M captures the tonal tension profile of any given chord progressions. Thus, we compared the fitness values of the chord progressions with all the curves given in the options. We calculated the correlation between the fitness of each curve and the hit rate that the users gave. As they were two datasets, we can apply the Pearson correlation. We present the Pearson correlation value ρ and the p -value in Table 1. The p -value has been calculated by transforming the correlation to create a t-statistic measure to proof the null hypothesis. The null hypothesis states that there is no relation between the objective measure and the subjective values obtained from the participants of the listening test. Therefore, if one of the p -values is lower than 0.05 the Pearson correlation is considered significant.

Table 1. Comparison of the statistics resulting from organizing the subjective ratings as a function of measures from other tonal tension models.

		M	Lerdahl's Model	ChordAIS
Curve A	p -value	2.5×10^{-4}	3.8×10^{-3}	2.7×10^{-3}
	ρ	0.8128	0.7345	−0.6177
Curve B	p -value	4.9×10^{-6}	6.1×10^{-6}	3.2×10^{-5}
	ρ	0.8999	0.8547	−0.7490
Curve C	p -value	9.2×10^{-6}	1.7×10^{-5}	8.7×10^{-4}
	ρ	0.9346	−0.9190	−0.7812
Curve D	p -value	3.1×10^{-5}	2.2×10^{-5}	9.2×10^{-4}
	ρ	0.8727	0.8326	−0.6906
Curve E	p -value	1.2×10^{-4}	3.9×10^{-3}	1.8×10^{-2}
	ρ	0.8727	0.8156	−0.6357

The first column gives the statistical measures corresponding to the tonal tension profile calculated by the measure M . The results suggest that the objective function captures the perceptual quality of the chords. The p -values are all much lower than 1% for the null hypothesis and the Pearson correlation coefficients ρ are above 0.8. This indicates that the subjective ratings of the users correlate strongly with the tonal tension profile that our model predicts.

To validate M against other proposals, we use two works proposed previously: the Tonal Tension Model proposed by Lerdahl, which is one of the most prominent models in the community, was described in Section 2 and captures hierarchical and linear properties following a different topological space, and ChordAIS, a previous work proposed in [24], which captures only linear properties of chord progressions. ChordAIS captures the following properties: consonance, perceptual relatedness to the previous chord, perceptual relatedness to the key and the harmonic function it represents. In our present work, apart from these properties, we also included hierarchical properties, like the hierarchical tension, and also the voice leading. Table 1 shows a comparison of the statistics resulting from organizing the subjective ratings obtained in the listening test as a function of distance

measures from other chord representations. To clarify the statistical analysis, we only calculated the statistical analysis with the curve that correlates best with the subjective values, ignoring the other possibilities, that resulted in lower correlation values in all the cases, independently of the model applied.

M is the objective measure from Equation (5). For the other columns, L stands for the distance according to the Lerdahl's model of tonal tension, and C stands for the ChordAIS model, in which only linear properties of the music are considered.

The statistical analysis shows that M captures the tonal tension profile of the chords better than L and D . The high ρ values obtained indicate the linear regression of the subjective ratings as a function of the objective values and the p -values are all below the 1% threshold for the null hypothesis, which suggests that the listeners' ratings correlate with the fitness measure well. Note that Lerdahl's measure L also correlates with the subjective ratings of the listening test, but obtaining slightly lower values for ρ and p -value. Likewise, the ρ and p -value in D are correlated, but below the rates obtained for M .

6.2. Evaluation of the Gp with Immune Properties

The proposed algorithm is capable of finding multiple local minima of the objective function while preserving diversity. The progressions with the highest fitness values would be considered most appropriate to be selected, as they comply with the specifications given by the users in the input. However, there are other chord progressions that can also comply with the rules. All these potential solutions can be local minima and can have different fitness values. We aim to evaluate if the chords selected by the algorithm are all considered better candidates than the chords that were discarded.

For this purpose, we performed a listening test to evaluate the perceptual quality of the chords proposed by the algorithm. We run the system 10 times with different curves and selected 2 candidate chords selected by the algorithm and 2 additional chords that were rejected, sampling the objective function regularly from lower to higher fitness values.

Finally, we designed a listening test that presents these 40 chord progressions ordered randomly with a picture of a curve that represents their tonal tension profile. As mentioned above, we included chords selected by the algorithm as well as chords rejected by it to validate the objective function. Each chord progression can be played multiple times before assessing it. This listening test also presents chord progressions with between 5 and 8 chords in each one. As in the previous listening test, each chord progression may have a different tree structure, with different keys. The rhythm in which the chords are played was always regular. Each chord progression was presented with a reproduction bar, and they were synthesized with a standard piano. The presentation of each chord progression in the listening test is shown Figure 5.

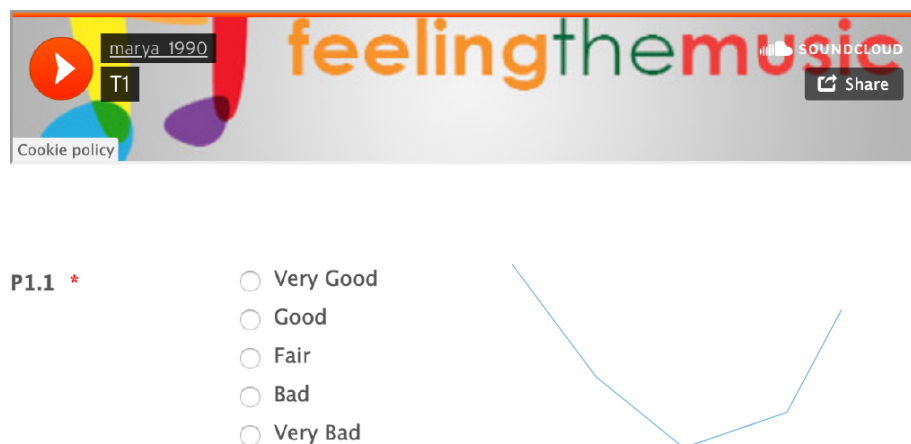


Figure 5. Shows the results of the fitness value proposed by ChordAIS-Gen against the users' subjective ratings. The red * symbol means this question is mandatory to finish the listening test.

The listeners were asked to rate how well the curve represents the tonal tension profile of the chord progression from 1 (Very bad) to 5 (Very good). In total, 35 people took the test, among which 8 declared no musical training, 10 considered themselves amateurs, and 17 professional musicians. We expect the chords selected by our algorithm to be rated positively by the listeners, while we expect the discarded chords to be rated negatively.

Figure 6 shows the values of the subjective ratings (represented in the vertical axis) versus the fitness values (in the horizontal axis) for the chord progressions included in the listening test. To simplify the analysis, we calculated the mean of the fitness values of the chord progressions that our algorithm ranked first, second, third, and successively until the eighth. We normalized these values between 0 and 1, and these mean fitness values were represented in the horizontal line. We also calculated the mean and standard deviation of the subjective ratings of the participants. The standard deviation is represented with the vertical lines, and the points are the cut points between the objective fitness of our model and the subjective ratings of the users for each chord progression. The first four points are chord progressions discarded by the algorithm, while the rest are progressions that our algorithm considers as good candidates. According to Figure 6 all the progressions selected lie above the horizontal line that cuts through 3. This indicates that our algorithm selects chord progressions that follow the tonal tension profile presented, according to the listening participants. Most chords discarded lie below the fair rating. Therefore, according to our results, we can conclude that our algorithm is able to select options that adapt to their tonal tension profile.

Our goal is to show that globally, our system can capture the tonal tension profile of a chord progression according to both musicians and non-musicians perceptions. As a future work, we will perform a deeper analysis to show how the results differ between people with and without musical training.

We aim to demonstrate that the proposed algorithm improves the performance of algorithms like AIS and GP. Therefore, we replace our algorithm with a standard version of GP and a standard version of AIS. We selected opt-aiNet, an optimization AIS proposed by [42] which is quite similar to the present proposal and was used in previous work, and also selected a classical approach for GP. We configured the parameters of the algorithms to its best performance empirically. Table 2 shows the values for the three algorithms.

There were two main reasons to create a new algorithm. The first one is that the different mutation patterns allowed to improve the performance of the algorithms. The second was to offer options with different properties. Therefore, we run the algorithms 50 times and calculated the mean and standard

deviation of the fitness value of the best individual. All of them made use of the *M* function. As we can see in Table 2, the fitness value was better in our algorithm, although the GP and the AIS also performed positively.

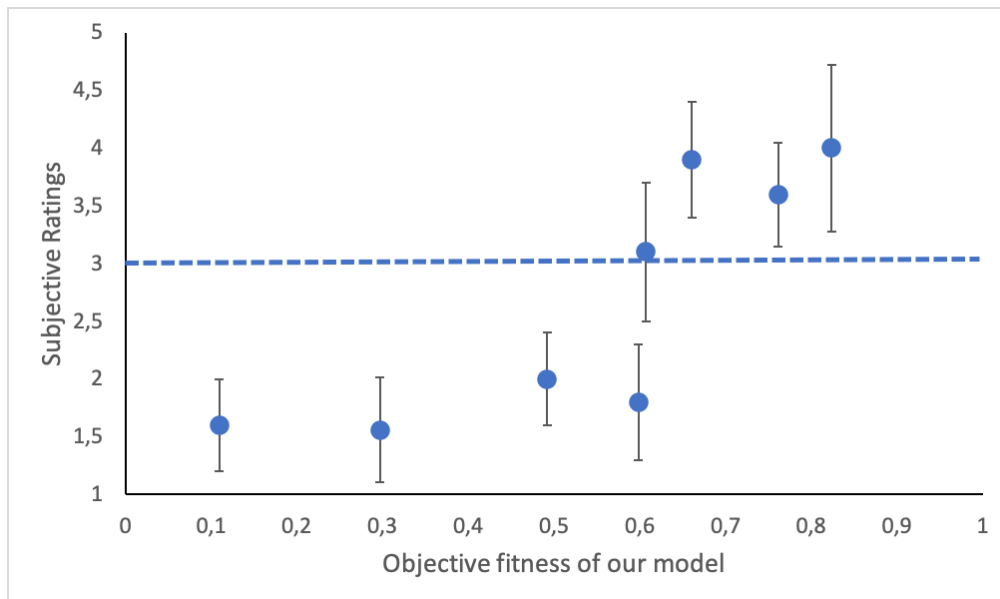


Figure 6. Shows the results of the fitness value proposed by ChordAIS-Gen against the users' subjective ratings.

Additionally, we aim to validate that our algorithm obtains candidates with different properties. For this purpose, we calculated the Euclidean distance between the three candidates proposed by our algorithms against the AIS, as GP only finds a single option. We also run the algorithms 50 times and calculated the mean and standard deviation of the normalized similarity measure of the three candidates in each run. The results in Table 2 show that the similarity value is significantly better for the algorithm proposed.

Table 2. Results of the objective function for three different configurations of the GA.

Parameters	opt-aiNet	GP	AIS+GP
Mutation Rate	According to the fitness	0.3	0.6 and the fitness
Cross-over Rate	-	0.4	0.4 and the fitness
Selection Function	-	Tournament	-
Number of clones	15	10	5
Initial Population	250	300	200
Suppression value	0.02	-	0.001
Random population	10%	-	10%
Fitness value	0.8413 ± 0.2045	0.7964 ± 0.1546	0.8734 ± 0.2385
Similarity	0.38 ± 0.1845	-	0.67 ± 0.2705

According to the results, our algorithm proposes other candidate chord progressions that are different from the best candidate and are still rated positively according to the fitness value. Our algorithm keeps these candidate chords, which are less trivial to find with traditional methods, because of the maintenance of diversity feature.

One of the limitations of our work is that, as with most of the bioinspired algorithms mentioned, our algorithm is computationally intensive. We calculated a mean of the CPU consumed during the

execution and the time it takes to output a complete chord progression. The CPU consumed was approximately 12.07% and the mean time was 9.85 min. In a future work, we would analyze how to improve the time of execution to get a more interactive tool capable of giving a response to the user in a shorter time.

6.3. Evaluating the Usability of Chordais-Gen as an Assistive Tool

In this third experiment, we aimed to evaluate the usability of ChordAIS-Gen from a user perspective. The system presented aims to help those users who might not have enough knowledge about tonal music to generate a good chord progression. The algorithm facilitates generating different but good options for candidate chords which can follow the progression, but we also want to validate if these chords are suitable selections for the user.

In total, 8 users without musical training were selected and asked about their experience with ChordAIS-Gen. Initially, they used the system to generate 5 chord progressions with any degree of freedom, to become familiar with ChordAIS-Gen. Then, they had to generate a total of 10 progressions. Each chord progression consists of a minimum of 5 chords and a maximum of 16 chords in a concrete key signature and with different curves. Initially, the users select the key, the number of chords and the tension profile. Then, ChordAIS-Gen proposes three-chord progressions. The user can play the progressions whenever they need to and select which one is more desirable according to their criterion. They can modify a concrete chord if they wish to. The interface constructed is shown in Figure 7.

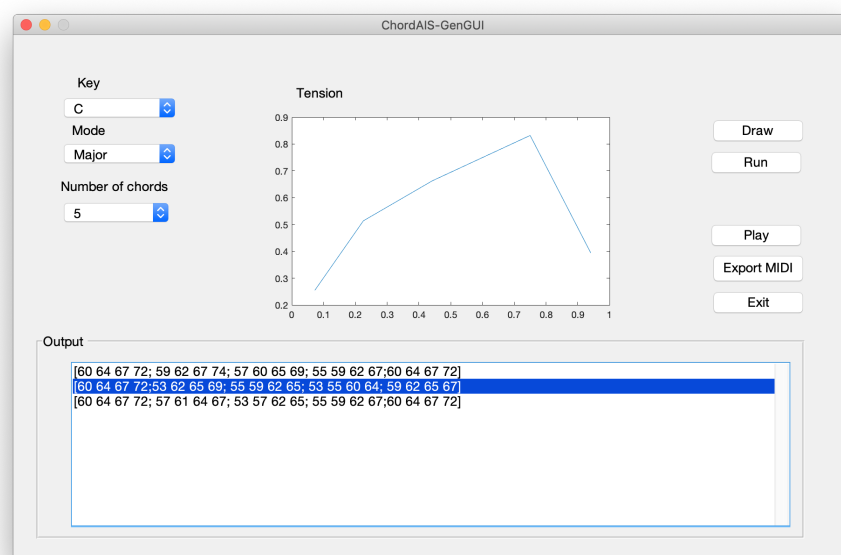


Figure 7. Interface of ChordAIS-gen.

To also compare the ChordAIS-Gen system with our previous work, ChordAIS, we repeated the experiment with the same users. They used the ChordAIS system to generate 10 chord progressions, in the same conditions as in ChordAIS-Gen.

In every case, the chord progressions that both systems proposed are not ordered by their fitness, in order not to bias the selection of one particular progression. All the participants have used the tool for at least 10 min, and have generated a minimum of 8 progressions. With that experience, they had to respond to a questionnaire that aims to capture their opinion about the system. In particular, the users gave their opinion about three items:

- The system complies with their expectation. The system is capable of giving good solutions that follow the initial parameters (tonal tension, key and number of chords).

- The system can give nice chord progressions that the user could not otherwise have composed.
- The system presents solutions that are different among them.

All the questions could be rated from 1 (“Completely unsatisfied”) to 5 (“Completely satisfied”). Table 3 shows the mean ratings and the standard deviation of the experiment.

Table 3. Shows the percentages of satisfaction when the users finished testing the ChordAIS-Gen system.

	Expectation	Usability	Different Solutions
ChordAIS-Gen	4.29 ± 0.50	4.15 ± 0.67	4.04 ± 0.75
ChordAIS Values	4.17 ± 0.75	4.03 ± 0.67	3.17 ± 0.75

The table shows that, in general, ChordAIS-Gen satisfies users’ requirements (4.29). The users are satisfied with the system, as it is capable of retrieving good and different candidate progressions that adapt to the tonal tension profile, with a mean of 4.45. Additionally, the users consider that amount of options to select chord progressions to be quite adequate (4.34), meaning that ChordAIS-Gen can retrieve solutions which are different one from another. The authors consider that this feature can be improved in future work by adding a learning component to understand the users’ preferences and incorporate this to the selection of chord progressions.

The previous version, ChordAIS also obtained good results. The satisfaction degree and the usefulness of the ChordAIS also had high ratings, although slightly lower than in the new version. However, ChordAIS cannot retrieve information about the tonal tension profile, only about the key and number of chords. Additionally, the users gave a lower value when they were asked about the diversity of solutions (3.17). Thus, the users think that both systems can assist them but ChordAIS generates more expected chord progressions than ChordAIS-Gen.

It is important to note that we also gave space to make some qualitative comments about our application and the experiments with the curves. Regarding the interface of our system, musicians would prefer to show the progressions with a musical score instead of a MIDI list. As the results can be exported to MIDI files, this shortcoming was partially addressed. In the experiments carried out, we obtained some comments in which they stated that they felt sometimes that the curves are too simple, and they would need a more complex curve to totally express the tonal tension profile of the chord progression.

On the contrary, the most common comment for non-musicians was that the extra description of the tonal tension was helpful to wholly understand the work. Additionally, as the curves represent general profiles, matching them with the chord progressions was easier than expected. However, in the usability test, the non-musicians needed deeper training to fully understand the tension profiles. Even though, some of them needed some help at some point while we were running the system to interpret the results and contrast them against the curve they depicted. In a future work, we will try to improve the interface. We also planned to create a new experiment in which the musical expert can draw their own curve instead of selecting among several standard curves.

7. Conclusions and Future Work

This present paper proposes a system that assists a user to generate tonal chord progressions, following a tonal tension profile. The tonal tension profile is a curve that modulates the path of the music played, and it depends on different musical properties. The system is able to search for different chord progressions that comply with musical constraints such as consonance, voice leading, hierarchical tension and perceptual distance. To capture all these musical properties, a function M , encoded in the Tonal Interval Space (TIS), has been designed. A Genetic Programming with artificial immune system properties has been developed and integrated into ChordAIS-Gen to find multiple solutions in parallel, resulting in multiple chord progressions that follow the tonal tension profile.

This new algorithm can be applied to problems in very different contexts, including Healthcare, energy optimization or smart cities.

A listening test was carried out to demonstrate that the chord progressions proposed by ChordAIS-Gen adapt to the tonal tension profile. Most listeners appreciated the same curve that our model predicted with the function M . Additionally, most listeners also rated the chord progressions proposed by ChordAIS-Gen more positively than the progressions discarded. Statistical analysis showed that our algorithm is able to propose good candidates according to the subjective ratings. Additionally, we made a comparison to evaluate if the incorporation of AIS properties to a GP algorithm can improve the performance of those algorithms individually.

We would like to highlight the subjective nature of this task, the generation of music, is therefore hard to evaluate through mathematical measures. We developed the listening tests, which are subjective evaluations, and thus, whose results depend on factors such as culture, musical education or personal preferences. Those factors are not encoded in the objective function. Consequently, even when the participants involved were familiar with the Western tonal music, the results of the listening test can vary between individuals. For example, people with musical training may be more receptive to capture tonal tension due to wide exposure to Western tonal music, or can interpret dissonances as possibilities within the tonal music paradigm.

In this work, we focused on the long-term and hierarchical aspects of the automatic generation of chord progressions. Furthermore, we focused on the ability to provide multiple options that are considered good candidate solutions with variety among themselves. However, the system does not learn from previous experiences to understand the users' preferences. Future work should investigate how to automatically incorporate this knowledge and combine it with the function constructed. Additionally, we will compare our results with other bioinspired algorithms and proposals based on artificial neural networks, which have demonstrated positive results in other areas [48,49].

Author Contributions: Conceptualization by M.N.-C. and J.F.M.S.-J.; methodology, M.N.-C. and J.F.M.S.-J.; software, M.N.-C. and R.G.-O.; validation, V.R.Q.L. and R.G.-O.; data curation, V.R.Q.L.; writing—original draft preparation, M.N.-C. and V.R.Q.L.; writing—review and editing, J.F.M.S.-J. and R.G.-O.; supervision M.N.-C. and V.R.Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: Research partially funded by the project “Co-POEM: Platform for the Collaborative Generation of European Popular Music” (ES01-KA201-064933) under the program Erasmus+: KA201-Strategic Partnership, and also partially supported by Fundacao para a Ciencia e a Tecnologia under Project UIDB-04111-2020.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Riemann, H.; Haggh, R.H. *History of Music Theory*; University of Nebraska: Lincoln, Nebraska 1962.
2. Schoenberg, A. *Theory of Harmony*; University of California Press: Berkeley, CA, USA, 1978.
3. Agmon, E. Functional Harmony Revisited: A Prototype-theoretic Approach. *Music. Theory Spectr.* **1995**, *17*, 196–214.
4. Schöenberg, A. *The Musical Idea and the Logic, Technique and Art of its Presentation*; Indiana University Press: Bloomington, IN, USA, 2006.
5. Stock, J. The Application of Schenkerian Analysis to Ethnomusicology: Problems and Possibilities. *Music Anal.* **1993**, *12*, 215–240.
6. Cope, D. Computer Modeling of Musical Intelligence in EMI. *Comput. Music. J.* **1992**, *16*, 69–83.
7. Paiement, J.F.; Eck, D.; Bengio, S. A Probabilistic Model for Chord Progressions. In Proceedings of the International Conference on Music Information Retrieval, London, UK, 11–15 September 2005; pp. 312–319.
8. Manaris, B.; Roos, P.; Machado, P.; Krehbiel, D.; Pellicoro, L.; Romero, J. A Corpus-based Hybrid Approach to Music Analysis and Composition. In *Proceedings of the National Conference on Artificial Intelligence*; AAAI Press (MIT Press): Menlo Park, CA, USA; Cambridge, MA, USA; London, UK, 2007; Volume 22, p. 839.
9. Eigenfeldt, A.; Pasquier, P. Realtime Generation of Harmonic Progressions Using Controlled Markov Selection. In Proceedings of the ICCX-Computational Creativity Conference, Edinburgh, UK, 12–14 December 2005; pp. 16–25.

10. Anders, T.; Miranda, E.R. A Computational Model that Generalises Schoenberg's Guidelines for Favourable Chord Progressions. In Proceedings of the 6th Sound and Music Computing Conference, Porto, Portugal, 23–25 July 2009.
11. Delgado, M.; Fajardo, W.; Molina-Solana, M. Inmamusys: Intelligent multiagent music system. *Expert Syst. Appl.* **2009**, *36*, 4574–4580.
12. López-Ortega, O.; López-Popa, S.I. Fractals, fuzzy logic and expert systems to assist in the construction of musical pieces. *Expert Syst. Appl.* **2012**, *39*, 11911–11923.
13. Bernardes, G.; Cocharro, D.; Caetano, M.; Guedes, C.; Davies, M.E. A multi-level tonal interval space for modelling pitch relatedness and musical consonance. *J. New Music. Res.* **2016**, *45*, 281–294.
14. Ebcioğlu, K. An expert System for Harmonizing Chorales in the Style of JS Bach. *J. Log. Program.* **1990**, *8*, 145–185.
15. Steedman, M.J. A Generative Grammar for Jazz Chord Sequences. *Music. Percept.* **1984**, *2*, 52–77.
16. Moroni, A.; Manzoli, J.; Von Zuben, F.; Gudwin, R. Vox populi: An Interactive Evolutionary System for Algorithmic Music Composition. *Leonardo Music J.* **2000**, *10*, 49–54.
17. Fukumoto, M. Creation of Music Chord Progression Suited for User's Feelings Based on Interactive Genetic Algorithm. In Proceedings of the 3rd International Conference on Advanced Applied Informatics (IIAIAI), Kita-Kyushu, Japan, 31 August–4 September 2014; pp. 757–762.
18. Navarro, M.; Caetano, M.; Bernardes, G.; de Castro, L.N.; Corchado, J.M. Automatic Generation of Chord Progressions with an Artificial Immune System. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 175–186.
19. Kaliakatsos-Papakostas, M.A.; Floros, A.; Vrahatis, M.N. Interactive music composition driven by feature evolution. *SpringerPlus* **2016**, *5*, 826.
20. Loughran, R.; O'Neill, M. Evolutionary music: applying evolutionary computation to the art of creating music. In *Genetic Programming and Evolvable Machines*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1–31.
21. McDermott, J.; O'Reilly, U.M. An executable graph representation for evolutionary generative music. In Proceedings of the 13th annual conference on Genetic and evolutionary computation, Dublin, Ireland, 20 June 2011; pp. 403–410.
22. Scirea, M.; Togelius, J.; Eklund, P.; Risi, S. Metacompose: A compositional evolutionary music composer. In *Proceedings of the International Conference on Computational Intelligence in Music, Sound, Art and Design*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 202–217.
23. Herremans, D.; Sørensen, K. Composing fifth species counterpoint music with a variable neighborhood search algorithm. *Expert Syst. Appl.* **2013**, *40*, 6427–6437.
24. Navarro-Caceres, M.; Caetano, M.; Bernardes, G.; de Castro, L.N. ChordAIS: An assistive system for the generation of chord progressions with an artificial immune system. *Swarm Evol. Comput.* **2019**, *50*, 100543.
25. Lerdahl, F. *Tonal Pitch Space*; Oxford University Press: Oxford, UK, 2005.
26. Granroth-Wilding, M.; Steedman, M. Statistical parsing for harmonic analysis of jazz chord sequences. In Proceedings of the ICMC, Ljubljana, Slovenia, 9–15 September 2012.
27. Herremans, D.; Chew, E. MorpheuS: Generating structured music with constrained patterns and tension. *IEEE Trans. Affect. Comput.* **2017**, *10*, 510–523.
28. De Castro, L.N.; Timmis, J. *Artificial Immune Systems: A new Computational Intelligence Approach*; Springer: Berlin/Heidelberg, Germany, 2002.
29. M'ullensiefen, D.; Wiggins, G. *Polynomial Functions as A Representation of Melodic Phrase Contour*; NA: Ann Arbor, MI, USA, 2011.
30. Lerdahl, F.; Jackendoff, R. *A Generative Theory of Tonal Music*; MIT Press: Cambridge, MA, USA, 1983.
31. Cohn, R. Neo-riemannian operations, parsimonious trichords, and their “tonnetz” representations. *J. Music. Theory* **1997**, *41*, 1–66.
32. Chew, E. Out of the Grid and Into the Spiral: Geometric Interpretations of and Comparisons with the Spiral-Array Model. *Comput. Musicol.* **2007**, *15*, 51–72.
33. Harte, C.; Sandler, M.; Gasser, M. Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimed*, Santa Barbara, CA, USA, 27 October 2006; ACM: New York, NY, USA, 2006; pp. 21–26.
34. Schenker, H.; Oster, E. *Free Composition: Volume III of New Musical Theories and Fantasies*; Pendragon Press: Hillsdale, NY, USA, 1979.

35. Rohrmeier, M. Towards a generative syntax of tonal harmony. *J. Math. Music.* **2011**, *5*, 35–53.
36. Lerdahl, F.; Krumhansl, C.L. Modeling tonal tension. *Music. Perception Interdiscip. J.* **2007**, *24*, 329–366.
37. Quinn, S.; Watt, R. The perception of tempo in music. *Perception* **2006**, *35*, 267–280.
38. Lester, J. Rameau and eighteenth-century harmonic theory. In *The Cambridge History of Western Music Theory*; Cambridge University Press: Cambridge, UK, 2002; pp. 753–777.
39. Farbood, M.M. A Quantitative, Parametric Model of Musical Tension. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2006.
40. Banzhaf, W.; Nordin, P.; Keller, R.E.; Francone, F.D. *Genetic Programming*; Springer: Berlin/Heidelberg, Germany, 1998.
41. Affenzeller, M.; Wagner, S.; Winkler, S.; Beham, A. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*; CRC Press: Boca Raton, FL, USA, 2009.
42. de Castro, L.N.; Timmis, J. An artificial immune network for multimodal function optimization. In *Proceedings of the Congress on Evolutionary Computation, CEC'02, IEEE, Honolulu, HI, USA, 12–17 May 2002*; Volume 1, pp. 699–704.
43. Navarro-Cáceres, M.; Caetano, M.; Bernardes, G. Objective Evaluation of Tonal Fitness for Chord Progressions Using the Tonal Interval Space. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 150–164.
44. Pearce, M.; Meredith, D.; Wiggins, G. Motivations and methodologies for automation of the compositional process. *Music. Sci.* **2002**, *6*, 119–147.
45. Herremans, D.; Weisser, S.; Sørensen, K.; Conklin, D. Generating structured music for bagana using quality metrics based on Markov models. *Expert Syst. Appl.* **2015**, *42*, 7424–7435.
46. Pearce, M.T.; Wiggins, G.A. Evaluating cognitive models of musical composition. In *Proceedings of the 4th International joint Workshop on Computational Creativity, Goldsmiths, London, UK, 17–19 June 2007*; pp. 73–80.
47. Collins, T.; Laney, R.; Willis, A.; Garthwaite, P.H. Developing and evaluating computational models of musical style. *Artif. Intell. Eng. Des. Anal. Manuf.* **2016**, *30*, 16–43.
48. Majewski, M.; Kacalak, W. Smart control of lifting devices using patterns and antipatterns. In *Computer Science On-line Conference*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 486–493.
49. Majewski, M.; Kacalak, W. Innovative intelligent interaction systems of loader cranes and their human operators. In *Computer Science On-line Conference*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 474–485.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).