Four-part chorale transformation by harmonic template voicing

Louis Bigo, Darrell Conklin

Four-part chorale transformation by harmonic template voicing

Louis Bigo Darrell Conklin

Universidad del País Vasco
Departamento de Ciencias de la Computación e Inteligencia Artificial
Technical report - April 2016

Abstract

This work presents a method to transform a four-part chorale in the style of J. S. Bach into a new one. The transformation conserves the harmonic structure of the original chorale which is therefore seen as a *harmonic template*. Four new sequences of notes which respect the harmonic template are computed to constitute the voices of the transformed chorale. The transformation is performed by running a local search algorithm that minimizes two concurrent objective functions that compute the likelihood of the horizontal and vertical organization of pitches regarding a statistical model built from a reference corpus of 185 four-part chorales from J. S. Bach. Results show that the algorithm performs well in generating voices in the style of J. S. Bach and could constitute a promising tool for four-part chorale chord transformation.

1 Introduction

Four-part chorale harmonization constitutes a widely studied problem in music generation. This task typically is to generate three voices from one which is given as input (generally the soprano voice). Voices are generated with the help of a model that can be either rule-based, either statistical. A good overview of existing approaches to the modelling of harmony is provided by Whorley et al. (2013). Constraint-based methods include the rule-based system CHORAL (Ebcioğlu, 1988) and different approaches described by Pachet and Roy (2001). Hidden Markov models are useful in automated harmonization as they model pertinently the hidden aspect of a chord sequence which is searched from a given melodic line (Raphael and Stoddard, 2003; Allan and Williams, 2005). Models that require chord information as input and models that do not are discussed and compared throughout different Bayesian network models by Suzuki and Kitahara (2014). The harmonization problem has also been approached with interpolated probabilistic models (Raczyński et al., 2013), evolutionary methods (Phon-Amnuaisuk et al., 1999; Donnelly and Sheppard, 2011), neural networks (Hild et al., 1992), multiple viewpoint models (Whorley et al., 2013; Whorley and Conklin, 2015) and N-gram models (Biyikoglu, 2003).

In this work, we propose an approach based on statistical models for chorale generation, guided by the transformation of a *template chorale* from which the intra-opus harmonic structure is inherited. The transformation is performed by a local search algorithm that iterates local modification on an initial chorale. Local search algorithms have already proven to be useful in music generation (Herremans, 2014; Herremans et al., 2014; Davismoon and Eccles, 2010; Lo and Lucas, 2006). A motivation of the transformational approach to music generation is to benefit from *conserved* high-level structures (especially underlying harmonic structures in the case of that work) that are hard to control along the generation of a music surface. The generation therefore focus on some *variable* musical objects, producing a transformation of the initial sequence that maintains its particular structural aspects. A typical application of such system is to provide some tools to the composer along the creative process. A transformation system can indeed be used at any step of the composition process to provide some *alternative* realisations of an abstract musical idea.

The presented approach differs from classical harmonization methods in two main aspects. First, it handles the generation of four original voices (instead of only three). Secondly, it takes as an input an existing chorale as a *template* for the generation of a new chorale. The presented method corresponds then to the task of transformation rather than generation or harmonization.

The presented transformation process includes a preliminary rhythmic fragmentation task which allows to represent the template chorale as a succession of pitch slices. It then comprises the computation of a pitch value in each slice for each of the four voices. The computation of pitches is constrained by the harmonic structure inherited from the chorale template. The harmonic structure of a chorale is represented by a sequence of key segments, a sequence of chord segments and a *harmonic label* for each note of the template. These labels represent the relation of the voices notes with the current chord and key throughout the chorale.

The transformation is performed by a local search algorithm which looks for solutions that respect the input harmonic template and that minimize two concurrent objective functions created from a corpus of chorales from J. S. Bach. The *horizontal* objective function describes the likelihood of the chorale by focusing on the melodic lines that are produced. The *vertical* objective function describes the likelihood of the chorale focusing on the way the successive chords are realized through the melodies. Starting from an initial solution, the search iteratively performs local transformations that modify the pitch of the notes.

A unique feature of this method is that the computation of the new pitches is performed given the template chord and key sequence. An advantage of this approach is that the chord/key sequence of the template can be modified or re-generated by the user (or by a generative system) before the voicing process.

Section 2 reviews the representation method of *viewpoints* used by the transformation process. Section 3 presents horizontal and vertical objective functions that are used by the local search algorithm. Section 4 presents the local search algorithm illustrated with the transformation of the chorale BWV267 of J. S. Bach. The transformed chorale, as some additional transformation examples, can be found on a companion web page.

2 Representation of harmonic features

This section reviews different tools and methods for music representation. Special attention is given to *viewpoints* which provide a representation of both melodic and harmonic aspects of a chorale. Finally, two rhythmic transformations that are necessary to perform the transformation algorithm described in Section 4 are presented.

2.1 Viewpoint representation for transformations

The method of *viewpoints* is used to represent musical sequences. This representation method has already proven to be efficient in music prediction (Conklin and Witten, 1995), music classification (Conklin, 2013) and pattern discovery (Conklin, 2010). The viewpoint representation method has already been used for harmonic transformations of tonal music (Bigo and Conklin, 2015) and is adapted here for the specific case of four-part chorale transformation.

Musical sequences are represented at the surface level as sequences of events that have basic features including duration, onset time and additional values depending on the nature of the events (for example, note-based events include a pitch, chord-based events include a chord symbol, etc.). A *viewpoint* is a function mapping events to more abstract derived features. The function is partial, therefore it may be undefined ($\perp$) for some events. An event e is abstracted by application of a viewpoint τ to produce the abstract feature $\tau(e)$.

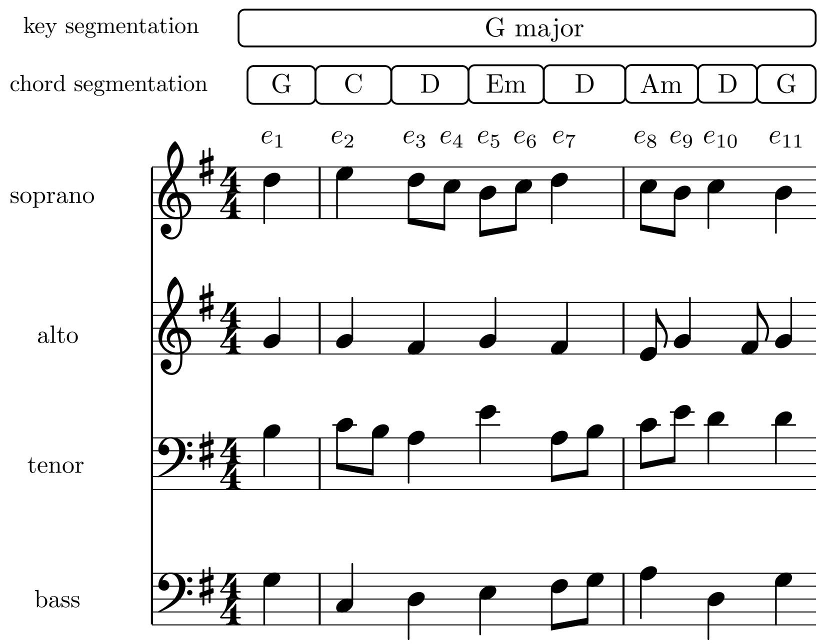
The application of k viewpoints $\tau_1 \dots \tau_k$ to an event sequence $e_1 \dots e_n$ may be represented as a $k \times n$ solution array where location (i, j) holds the value $\tau_i(e_j)$.

2.1.1 Note event based representation

The table of Figure 1 illustrates such an array, which describes with several note event viewpoints the eleven first notes of the soprano voice of the chorale BWV 267 *An Wasserflüssen Babylon* from J. S. Bach. For better readability, only the notes of the soprano voice (labeled by the events $e_1 \dots e_\ell$ on the upper staff) are represented with viewpoints. Values $\mathsf{int}(e_i)$ and $\mathsf{pc}(e_i)$ respectively correspond to the incoming pitch interval $\mathsf{pitch}(e_i) - \mathsf{pitch}(e_{i-1})$ and to the pitch class of the event $\mathsf{pitch}(e_i) \bmod 12$. Though not represented by viewpoints, the three other voices are kept on the figure to provide the reader the harmonic context, which is necessary to compute harmonic based viewpoints introduced below.

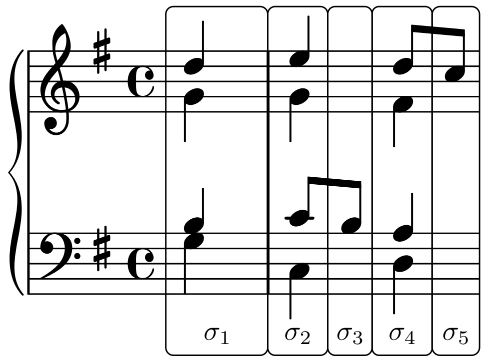
2.1.2 Pitch slice event based representation

In four part chorale composition, a large importance is given to the way chords are voiced throughout the different voices. This aspect can be described by applying a fragmentation of the rhythmic skeleton resulting from a pitch slice segmentation of



τ	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}	e_{11}
pitch	74	76	74	72	71	72	74	72	71	72	71
onset	1	3	5	6	7	8	9	11	12	13	15
duration	2	2	1	1	1	1	2	1	1	2	2
int	$\perp$	+2	−2	−2	−1	+1	+2	−2	−1	+1	−1
pc	2	4	2	0	11	0	2	0	11	0	11
key	G	G	G	G	G	G	G	G	G	G	G
chord	G	C	D	D	Em	Em	D	Am	Am	D	G
hlab	c	c	c	k	c	k	c	c	k	k	c

Figure 1: The opening of the chorale BWV 267 *An Wasserflüssen Babylon* from J. S. Bach. Key and chord segmentation are displayed at the top. The note events designating the first eleven notes of the Soprano voice are represented by different viewpoint functions in the array.



τ	σ_1	σ_2	σ_3	σ_4	σ_5
pitchset	$\{55, 59, 67, 74\}$	$\{48, 60, 67, 76\}$	$\{48, 59, 67, 76\}$	$\{50, 57, 66, 74\}$	$\{50, 57, 66, 72\}$
normpitchset	$(0, 4, 12, 19)$	$(0, 12, 19, 28)$	$(0, 11, 19, 28)$	$(0, 7, 16, 24)$	$(0, 7, 16, 22)$
duration	2	1	1	1	1

Figure 2: Slice description of the beginning of the chorale BWV 267.

the chorale. A pitch slice segmentation is based on appearance and disappearance of pitches throughout the time of the piece. Each time a pitch starts/stops to be played in any of the tracks/voices, a new slice begins. A slice σ is designated by the set of note events it includes. As this paper restrains to chorales constituted of four monophonic voices, $\sigma = [e_1 \dots e_n]$ with $n \leq 4$. This slice segmentation method is called *full expansion* and compared to other methods by Conklin (2002). Figure 2 illustrates the pitch slice segmentation of the beginning of the chorale BWV 267. Though not illustrated on the above example, pitch sets can include less than four elements if two voices are playing simultaneously the same pitch. Slice segments can have variable duration throughout the piece. In Figure 2, the first slice σ_1 has the duration of a quarter note. Slices σ_2 to σ_5 are each associated with the duration of a eighth note. In order to compare slices in a transposition-independent way, pitch-sets of slices are normalized by subtracting the value of the lower pitch from all the pitches of the slice. A normalized pitch-set naturally includes the same number of elements that the pitch set himself. The table under the Figure 2 illustrates the description of these successive slices by the viewpoints pitchset, normpitchset and duration that respectively provide the pitch set, the normalized pitch set and the duration associated with each slice.

2.2 Chord and key segmentation

A major task in harmonic analysis consists in labeling the score by chord and key annotations. A *chord segmentation* is a sequence of non-overlapping chord symbols, each associated with a duration, that cover the time-line of a musical sequence. Additionally, a *key segmentation* is a sequence of keys, each associated with a duration, covering the piece in the same way. The *harmonic segmentation* of a piece refers to the chord segmentation and the key segmentation resulting from the harmonic analysis of the piece. A harmonic segmentation is illustrated at the top of Figure 1. Though the harmony of the musical excerpt of Figure 1 is not ambiguous, in general there are no unique and exact methods for harmonic segmentation, in particular when inputs are MIDI files that do not include pitch spelling. Even when manually performed, harmonic segmentation depends on the analyst. Different methods trying to model this human cognitive ability have been investigated including an algorithm based on the spiral array (Chew, 2002), a dynamic programming approach (Rocher et al., 2009) that processes chord and key segmentation based on Lerdahl's tonal distance (Lerdahl, 2001), the Melisma system (Sleator and Temperley, 2001) and a system based on template matching and graph-search techniques (Pardo and Birmingham, 2002).

The transformation method presented in this paper requires a chord/key segmentation of the template chorale to compute chord and key viewpoints. This segmentation constitutes an additional input to the transformation. Whether it is manually performed or automatically computed by one of the previous systems does not impact the functioning of the transformation method. To process the transformation illustrated on subsection 4.3, a manual harmonic segmentation was performed.

2.2.1 Chord and Key Viewpoints

A chord segmentation induces a viewpoint chord that returns, for any note event e, the chord symbol of the chord segment in which e is included. The note viewpoint table of Figure 1 represents the chord viewpoint sequence (chord) associated with the chorale extract. In the same manner as chord segmentation, a key segmentation induces a viewpoint (key) that returns the key of the key segment that includes the event.

For an event e, the function $\mathsf{chord}_{\mathsf{pc}}$ returns the pitch class set associated with the chord symbol $\mathsf{chord}(e)$. Additionally, $\mathsf{key}_{\mathsf{pc}}(e)$ corresponds to the set of pitch classes containing all pitch classes in the scale of $\mathsf{key}(e)$. For example, in Figure 1, we have $\mathsf{chord}_{\mathsf{pc}}(e_6) = \{4, 7, 11\}$ and $\mathsf{key}_{\mathsf{pc}}(e_3) = \{0, 2, 4, 6, 7, 9, 11\}$, associated with the chord Em and the key G major.

2.2.2 Harmonic Label Viewpoint

The viewpoint hlab attributes a harmonic label to every note event of the template piece. For any event e, $\mathsf{hlab}(e)$ is computed from the values $\mathsf{pc}(e)$, $\mathsf{chord}_{\mathsf{pc}}(e)$ and $\mathsf{key}_{\mathsf{pc}}(e)$.

Though the notion of harmonic label can be defined in different ways, in particular depending on the musical style, a simple specification is proposed to illustrate the method. Four possible harmonic labels can be attributed to an event, depending on if its pitch belongs to its relating chord and key regarding the harmonic segmentation. More formally, we propose the harmonic labels $\mathcal{f}$ for "fundamental", $\mathcal{c}$ for "chord", $\mathcal{k}$ for "key" and $\mathcal{o}$ for "other" with:

$$\mathsf{hlab}(e_i) = \begin{cases} \mathcal{c} & \text{if } \mathsf{pc}(e_i) \in \mathsf{chord}_{\mathsf{pc}}(e_i) \\ \mathcal{k} & \text{if } \mathsf{pc}(e_i) \notin \mathsf{chord}_{\mathsf{pc}}(e_i) \text{ and } \mathsf{pc}(e_i) \in \mathsf{key}_{\mathsf{pc}}(e_i) \\ \mathcal{o} & \text{if } \mathsf{pc}(e_i) \notin \mathsf{chord}_{\mathsf{pc}}(e_i) \text{ and } \mathsf{pc}(e_i) \notin \mathsf{key}_{\mathsf{pc}}(e_i) \\ \mathcal{f} & \text{if } \mathsf{pc}(e_i) \text{ is the fundamental pitch class of the chord } \mathsf{chord}_{\mathsf{pc}}(e_i) \end{cases}$$

In Figure 1, we have $\mathsf{hlab}(e_4) = \mathcal{k}$ because $0 \notin \{2, 6, 9\}$ and $0 \in \{0, 2, 4, 6, 7, 9, 11\}$. The harmonic label $\mathcal{f}$ specifies that a note is playing the fundamental of the current chord. The harmonic label $\mathcal{o}$ describes a note whose pitch is outside the current key. This case is rare in J. S. Bach's four part chorales. Although, this label might be much more useful in the context of other musical styles in which chromatic tones appear more frequently as outlined by Bigo and Conklin (2015).

The above definition of hlab associates an harmonic label to any note event of the score. The harmonic label depends on the harmonic segmentation and on the variety of chord types and keys supported by the harmonic segmentation method. The set of available harmonic labels naturally impacts the precision of the harmonic description of the sequence.

2.2.3 Pitch candidate set

A *pitch candidate set* is a set of pitches in a precise register whose associated pitch-classes respect a particular harmonic label in the context of a chord and a key. For example, the pitch candidate set $\{55, 60, 64, 67, 72\}$ provides all pitches that respect the harmonic label $\mathcal{c}$ in the context of the chord CM in the key of F major reduced to the pitch register [53→73]. As an other example, the pitch candidate set $\{55, 59, 60, 64, 67, 71, 72\}$ provides all pitches that respect the harmonic label $\mathcal{k}$ in the context of the chord DM in the key of G major (i.e., pitches in the key of G major but outside the chord DM) reduced in this same pitch register. In the following, the pitch candidate set of a note event e will be designated by the term $\mathsf{C}(e)$.

2.3 Rhythmic slicing and merging

A note that overlaps two consecutive chord segments does not necessarily have the same harmonic function regarding the two corresponding chords. As a viewpoint can not attribute more than one value to a note event, a rhythmic fragmentation prior to the computation of harmonic labels can be applied on the template to avoid this problem. This fragmentation task follows the slicing principle mentioned in 2.1.2. A note that overlaps multiple successive slices is fragmented into several successive notes with same pitch, such that a note can not appear in more than one chord segment. This transformation is illustrated Figure 3 with the first measures of the chorale BWV267.

This rhythmic transformation naturally affects the represented piece and also any transformation, which inherits this artificial fragmentation. A simple post-transformation merging process can be applied on any fragmented piece in order to approximate rhythmic patterns generally encountered in four-part chorales from J. S. Bach. This merging process is based on a single rule specifying that two successive notes that have the same pitch, that don't overlap over more than a single measure and whose added duration does not exceed a quarter note and are merged into one single note. The application of this process on the fragmented extract illustrated on the right of Figure 3 permits the exact regeneration of the starting extract. Though not illustrated on this example, the merging task does not systematically allow to retrieve the starting piece.

A drawback of this method is that the merging process will not produce notes longer than a quarter note, avoiding in particular the production of half notes. The

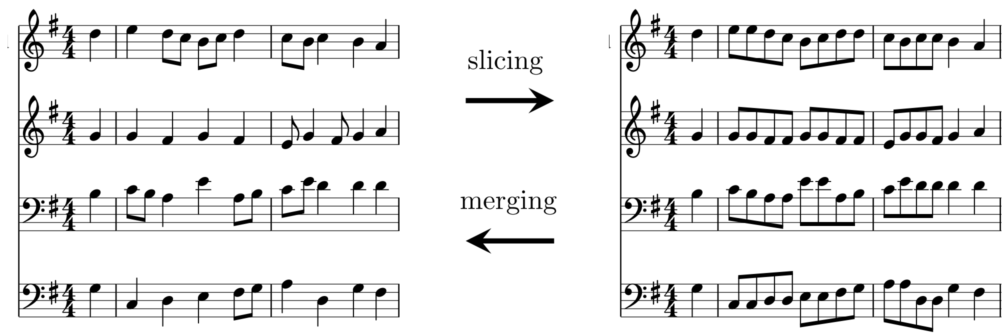


Figure 3: The slicing transformation performed on the first measures of the chorale BWV267.

compositional choice of a long note rather than a repeated short note frequently relies on the lyrics being sung. As this problem is out of the scope of this paper which focuses on pitch transformations, we will use the presented merging task to produce chorales without notes longer than a quarter note and without notes overlapping strong beats.

3 Voicing evaluation functions of chorales

In this section, we present the different functions used to evaluate specific aspects of a musical sequence. The computation of these evaluation functions allows to describe a four-part chorale regarding a statistical model. Chorales are evaluated from both horizontal and vertical points of view. Depending on which aspect is evaluated, a chorale will be either described horizontally as a set of four parallel monophonic sequences i.e., $\{s_1 \dots s_4\}$, either vertically as a sequence of pitch slices i.e., $\sigma_1 \dots \sigma_\ell$.

A corpus of 185 chorales by J. S. Bach, in midi format[1], is used to compute statistical models encoding both horizontal and vertical statistics of these pieces.

3.1 Horizontal evaluation

This section presents a function evaluating the melodic interval likelihood of a chorale regarding a corpus. One important composition rule in chorales in the style of J. S. Bach relies on the production of smooth melodic motions, for aesthetic reasons and also to facilitate the singing of the chorale. This rule is generally applied less strictly on the bass voice than on the three others, as this voice is frequently constrained by the necessity to sing the fundamental of the current chord. This observation highlights that the four voices are not composed following specifically the same rules, and that a voice should be evaluated regarding a model specific to its type (soprano, alto, tenor and bass). For that reason, each of the following models has been computed for each of the four voice types. A whole chorale is horizontally evaluated by averaging the horizontal evaluation of each voice computed regarding its own model.

[1]corpus available at `http://kern.humdrum.org/cgi-bin/browse?l=musedata/bach/chorales`

Chorale voices being monophonic, a melodic interval distribution is built from the corpus constituting then a zero order model of melodic intervals. The function $P(i)$ returns the probability of the interval i according to the distribution. The probability of a melodic sequence is given by the product of each successive interval probability regarding the model. More formally, the probability of a monophonic sequence $s = e_1 \dots e_\ell$ is given by

$$P(s) = \prod_{n=2}^{\ell} P(\mathsf{int}(e_n))$$

As it is more convenient to use logarithms when working with small values, the sequence s is described by its entropy. The sum of the negative logarithms is normalized to obtain the entropy $H_0(s)$ independently of the length of the sequence:

$$H_0(s) = -\frac{1}{\ell - 1} \sum_{n=2}^{\ell} \log_2 P(\mathsf{int}(e_n))$$

Although this interval model reveals melodic interval occurrences within the corpus, it fails at revealing some more specific composition strategies typical from four-part chorales in the style of J. S. Bach. For example, chorales in this style rarely include two successive leaps in the same direction. This type of behavior can however be encoded in a first order model of interval that can be built by counting occurrences of pairs of successive intervals. The entropy of the sequence s according to this model is given by:

$$H_1(s) = -\frac{1}{\ell - 1} \left(\log_2 P(\mathsf{int}(e_2)) + \sum_{n=3}^{\ell} \log_2 P(\mathsf{int}(e_n) \mid \mathsf{int}(e_{n-1})) \right)$$

A four-part chorale c corresponding to the set of voices $\{s_1 \dots s_4\}$ is horizontally evaluated by averaging the evaluations of its separate voices:

$$H(c) = \frac{1}{4} \sum_{i=1}^{4} H_1(s_i)$$

3.2 Vertical evaluation

A distribution of normalized pitch sets is computed from the slices of the reference corpus. To build the distribution, each normalized pitch-set encountered within the corpus was weighted by the duration of its slice in order to give higher importance to pitch-sets sounding longer. A total of 19209 slices are encountered within the corpus. These slices correspond to 1047 distinct normalized pitch-set whose probabilities vary from 2.13×10^{-5} to 5.86×10^{-2}. The normalized pitch-set having the highest probability is $(0, 16, 19, 24)$. The distribution directly attributes a probability $P(\sigma)$ to any slice whose normalized pitch-set appears within the corpus. A *minimum probability* p_0 is attributed to a normalized pitch-set that does not appear in the corpus at all. In this work, p_0 is arbitrarily chosen to be assigned to the value of the lowest normalized pitch-set probability encountered in the distribution.

The entropy of a chorale $c = \sigma_1 \dots \sigma_\ell$ regarding this model is given by:

$$V(c) = -\frac{1}{\ell} \sum_{n=1}^{\ell} \log_2 P(\mathsf{normpitchset}(\sigma_n))$$

This evaluation function refers to a slice model of order 0 and therefore describes purely vertical aspects of the chorale.

4 Transformation process

The presented transformation method involves the computation for each voice of a new melodic line fitting with the harmonic description of the template chorale. As a preliminary task, the slicing transformation presented in Section 2.3 is performed and allows therefore the computation of the viewpoint functions key, chord and hlab that attribute respectively to every note of the template chorale a key, a chord and an harmonic label. As outlined in Section 2.2.3, these values allow to compute for each voice of each slice a pitch candidate set. Candidate sets are restricted to pitch registers retrieved from the reference corpus ([C4→A5] for the Soprano voice, [E3→D5] for the Alto, [C3→A4] for the Tenor and [C2→E4] for the Bass). However, the choice of a pitch domain for generation can be handled in much more sophisticated ways as studied by Whorley et al. (2013).

The algorithm computes a pitch value for each voice at each slice. For example, the transformation of the extract displayed on the right of Figure 3 requires the computation of 60 pitch values (15 slices, 4 voices). At each position, a pitch is selected from the corresponding pitch candidate set.

To ensure that this computation produces a chorale in the style of the reference corpus, the selection of the pitches is viewed as a multi-objective optimization problem in which two objective functions (respectively called the horizontal and the vertical functions) are concurrently minimized. An initial solution is computed with attention to the horizontal function only. The initial solution is then iteratively modified in order to decrease its vertical function with minimum increase of its horizontal function.

4.1 Computation of an initial solution

The Viterbi algorithm is used to compute independently the four voices that minimize the horizontal objective function while conserving the original harmonic labels of the template chorale regarding its chord and key segments.

Let $e_1 \dots e_\ell$ be the event sequence representing one voice of the template chorale after application of the slicing process. Let S be the set of all possible substituting sequences $s' = e'_1 \dots e'_\ell$ respecting $\mathsf{pitch}(e'_i) \in \mathsf{C}(e_i)$ for every i in $[1 \dots \ell]$, where $\mathsf{C}(e_i)$ corresponds to the pitch candidate set of the note event e_i as introduced in 2.2.3. The Viterbi algorithm provides the sequence s_{init} which verifies $H_1(s_{\text{init}}) = \min_{s \in S} H_1(s)$. Applying this process for each of the four voices provides the transformed chorale that minimizes the horizontal objective function H.

It could seem interesting to use the template chorale as an initial solution. However, generating the initial solution provides much better chances to produce final results with more dissimilarity from the template piece. In the following, the chorale computed as the initial solution will be notated c_{init}.

4.2 Optimization by local search

During the computation of the initial solution, the four voices are generated independently one from each other, which tends to produce low probability slices, forming then a chorale with a weak vertical quality. In particular, the conservation of harmonic labels is not constraining enough to ensure a satisfying distribution of the pitch classes of a chord throughout the four voices taking part within a slice. This leads to the frequent production of incomplete chords (e.g., triads without third or without fifth) which hardly matches with the style of four-part chorales of J. S. Bach.

To address this problem, an optimization algorithm is performed on the initial solution to improve iteratively its vertical evaluation with minimal deterioration of its horizontal evaluation.

4.2.1 Slice neighborhood

The neighborhood $\mathsf{N}(\sigma)$ of a slice $\sigma = [e_1 \dots e_n]$ is the set of slices in which each event of σ may have its pitch replaced by a pitch of its own pitch candidate set. Though not prohibited, voice-crossing occurs rather infrequently and briefly in four-part chorales from J. S. Bach. This tendency being difficult to specify precisely, it is simplified by adding a constraint conserving the order of the pitch events which has the consequence to strictly forbid voice crossing and unison along the transformations.

More formally, a slice $\sigma' = [e'_1 \dots e'_n]$ belongs to the neighborhood of σ if

- $\forall e'_i \in \sigma',\ \mathsf{pitch}(e'_i) \in \mathsf{C}(e_i)$
- $\forall e'_i \in [e'_1 \dots e'_{n-1}],\ \mathsf{pitch}(e'_i) < \mathsf{pitch}(e'_{i+1})$

Given a slice statistical model, a slice σ' of a slice neighborhood $\mathsf{N}(\sigma)$ is said to belong to the *filtered neighborhood* $\mathsf{N}(\sigma, p)$ if its probability is strictly higher than the threshold p. In particular, the filtered neighborhood $\mathsf{N}(\sigma, P(\sigma))$ designates the set of neighbor slices having a greater probability than the original slice σ. Therefore, the higher is the probability of the slice σ, the smaller is the size of its filtered neighborhood $\mathsf{N}(\sigma, P(\sigma))$.

4.2.2 Chorale neighborhood

Let c be a chorale represented the sequence of slices $\sigma_1 \dots \sigma_\ell$. The chorale neighborhood $\mathcal{N}(c)$ is constituted by the set of chorales obtained by replacing one slice σ_i of c by an element of its neighborhood. In other words, the neighborhood $\mathcal{N}(c)$ is constituted by the set of chorales of the form $\sigma_1 \dots \sigma_{i-1}, \sigma'_i, \sigma_{i+1} \dots \sigma_n$ where $\sigma'_i \in \mathsf{N}(\sigma_i)$ for $i \in [1 \dots n]$. As a consequence, the size of this chorale neighborhood can be straightforwardly computed:

$$|\mathcal{N}(c)| = \sum_{i=1}^{n} |\mathsf{N}(\sigma_i)|$$

The filtered chorale neighborhood $\mathcal{N}_{\mathrm{f}}(c)$ is constituted by neighbor chorales with strictly lowest vertical entropy. This filtration is performed straightforwardly by considering the substitution of any slice σ by an element of its filtered slice neighborhood $\mathsf{N}(\sigma, P(\sigma))$. The size of the filtered chorale neighborhood is given by:

$$|\mathcal{N}_{\mathrm{f}}(c)| = \sum_{i=1}^{n} |\mathsf{N}(\sigma_i, P(\sigma_i))|$$

The task of substituting a chorale c by an element of its filtered neighborhood $\mathcal{N}_{\mathrm{f}}(c)$, which requires to review one by one the elements of this set, will therefore have a cost which relies to the the probabilities of the slices forming c.

4.2.3 Local search

Starting from the initial solution c_{init}, the algorithm iteratively substitutes the chorale being transformed by a chorale of its filtered neighborhood. This task decreases necessarily the vertical entropy of the transformed chorale. This property guarantees that the algorithm can not get stuck within a cycle.

The substitution of a chorale c by an element c' of its filtered neighborhood can be characterized by the value

$$\frac{H(c') - H(c)}{V(c) - V(c')}$$

Intuitively, this value corresponds to the slope which is produced between the two chorales represented within the bi-dimensional Cartesian plane in which points have for coordinates the two values $-V(c)$ and $H(c)$. Note that the necessary decreasing of the vertical entropy guarantees that $V(c) - V(c') > 0$. At each iteration of the search, the neighbor chorale that minimizes this value is selected to substitute the current chorale[2]. If the minimum slope is produced by more than one chorale of the neighborhood, priority is given to the elements returning the lowest horizontal evaluation. This task is iterated until there is no slice modification that allows to reduce the vertical evaluation (i.e., each successive slice is the highest probability slice in its own slice neighborhood leading to $\mathcal{N}_{\mathrm{f}}(c) = \varnothing$). The transformation process is summarized in Figure 4.

4.3 Experiment

The transformation process will here be illustrated using as a template the chorale BWV267. The chorale is given with its chord and key segmentation. These information

[2]This *greedy* approach which iteratively selects the solution resulting from the single element update that minimizes a function is known in the field of optimization under the term of *hill-climbing algorithm* (Russell and Norvig, 2003).

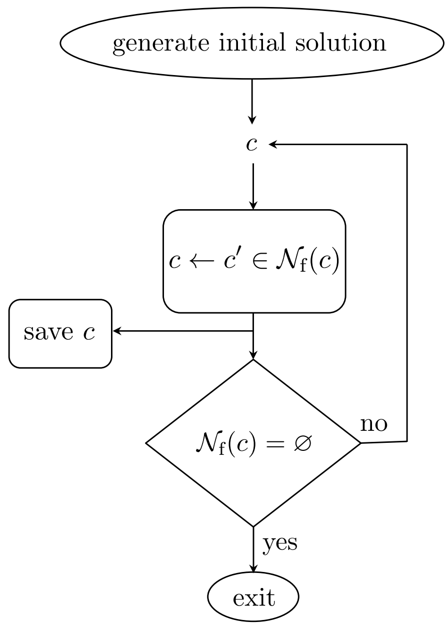


Figure 4: Overview of the optimization process.

attribute to every note a harmonic label within the set of values $\{c, k\}$ for the soprano, alto and tenor voices and $\{k, c, f\}$ for the bass voice. The harmonic label o is not used in this experiment as this chorale does not make appear any chromatic note (i.e., outside the current key). After the slicing task is processed, a pitch candidate set can be attributed to each note of the template given its corresponding harmonic label, chord and key. The local search algorithm can then be run.

Figure 5 illustrates the solutions computed successively throughout the local search. Transformed chorales are evaluated horizontally by their melodic first order interval entropy $H_1(c)$ (regarding a dedicated model for each of the four voices) and vertically by their 0-order slice entropy $V(c)$. The 185 chorales constituting the reference corpus are represented by crosses. The horizontal and vertical entropy of the corpus chorales are each computed regarding the model built without the chorale being evaluated. The template chorale is represented in red.

The initial solution computed with the Viterbi algorithm, which minimizes the function $H_1(s)$ has the values $H_1(c_{\mathrm{init}}) = 2.183$ and $V(c_{\mathrm{init}}) = 11.587$. Starting from the initial solution, the local search task performs 394 successive slice transformations (in about 18 minutes) until finding the chorale c_{end} that does not have any neighbor chorale with better vertical evaluation (i.e., $\mathcal{N}_{\mathrm{f}}(c_{\mathrm{end}} = \varnothing)$). We have $H_1(c_{\mathrm{end}}) = 4.586$ and $V(c_{\mathrm{end}}) = 5.511$. The horizontal evaluation of this final computed chorale is naturally far from the range of the corpus. The chorales successively computed between c_{init} and c_{end} are displayed in dark gray and restrained for a better readability to those having an horizontal evaluation lower than 3.5.

The trajectory taken by the algorithm shows that the transformation succeeds in producing chorales with reasonable horizontal and vertical evaluations values in com-

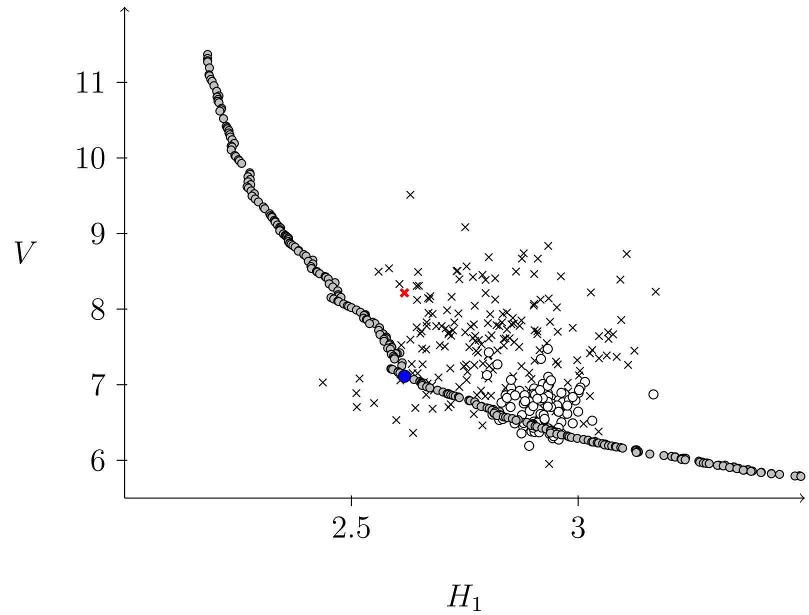


Figure 5: Solutions computed throughout the local search process. The chorales constituting the reference corpus are represented by crosses. The computed solutions are displayed in grey, including the selected solution in blue. The template chorale is the chorale BWV267 and is represented by the red cross.

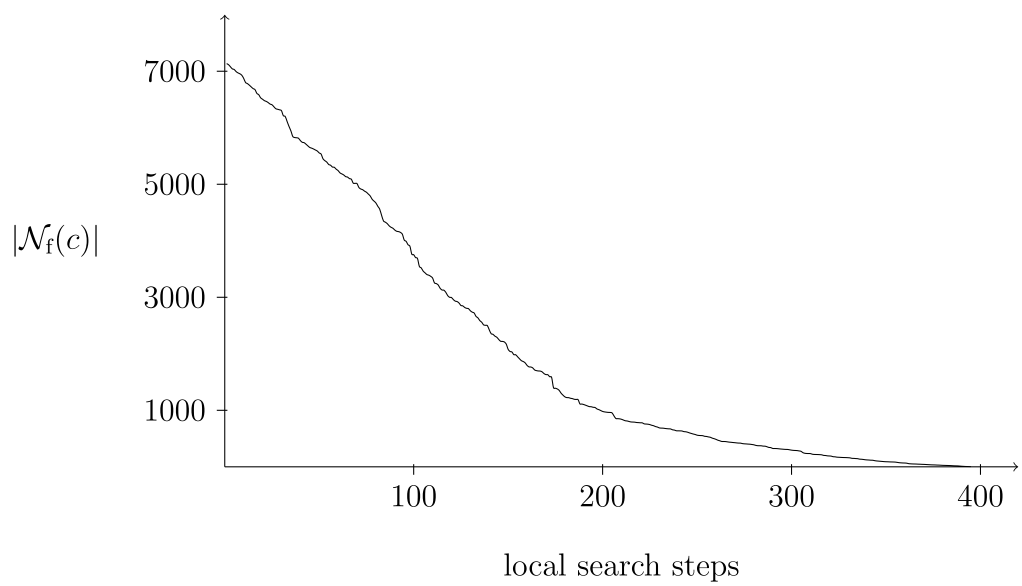


Figure 6: Decreasing of the size of visited chorales throughout the local search process.

parison with the corpus chorales. It outperforms the original template displayed in red on the diagram, by finding solutions having both better horizontal and vertical evaluation. At the beginning of the search, the algorithm succeeds in computing chorales with better vertical evaluation, almost without increasing the horizontal evaluation. Over the iterations, the search has more and more difficulties to find low horizontal entropy solutions. Figure 6 shows the evolution of the size of the chorale neighborhood throughout the 394 iterations of the local search. The decreasing number of visited chorales explains the growing lack of neighbor chorales limiting the increasing of the horizontal entropy.

A set of 100 harmonizations computed from the Soprano voice of the chorale BWV 267 with the method described in Whorley and Conklin (2015) is represented by white filled circles on Figure 5. This approach includes significant differences with the one presented in this paper. First, as an harmonization task, it includes the generation of only three of the four voices (the soprano voice is conserved). Secondly, the generation is not constrained by predefined chords and harmonic labels as it is the case in the presented method. The local search process succeeds in finding transformed chorales with horizontal and vertical evaluation in the same range than these harmonizations.

Selection of solutions Different strategies can be used to select one transformed chorale among the set of computed solutions. A natural preliminary selection is provided by the filtration of the set of dominant solutions (also called *Pareto set*). A solutions is dominant if there exists no other computed solution with both a better horizontal and a better vertical evaluation. From this set, one can select the solution which is the closest from the template chorale, or from the corpus barycenter. The solution displayed in blue on Figure 5 corresponds to the solution that reaches the lowest possible vertical evaluation without its horizontal evaluation exceeding the template horizontal evaluation (equal to 2.617). This solution is reached after 156 successive slice transformations and is designated under the term c_β. We have $H_1(c_\beta) = 2.617$ and $V(c_\beta) = 7.112$. The transformed chorale exhibits some interesting similarities with the template chorale. In particular, the second measure (beginning at the second note) is identical in the soprano and bass voices of both pieces and has only one note which differs in alto and tenor voices. The next measures exhibit more dissimilarities.

The first measures of the corresponding transformed chorale is illustrated on Figure 7 (1.b). The full transformed chorale can be found on the companion web page `https://soundcloud.com/harmonictransformations`, as the two other chorale transformations of Figure 7.

5 Conclusions

This paper presented an approach to four-part chorale transformation in the style of J. S. Bach. The specification of a chord sequence as an input parameter constitutes an strength of this approach. In this paper, the method has been illustrated by keeping the template chord sequence. However, voices could be computed regarding an alternative chord sequence, provided either by a composer, or by a generative system.

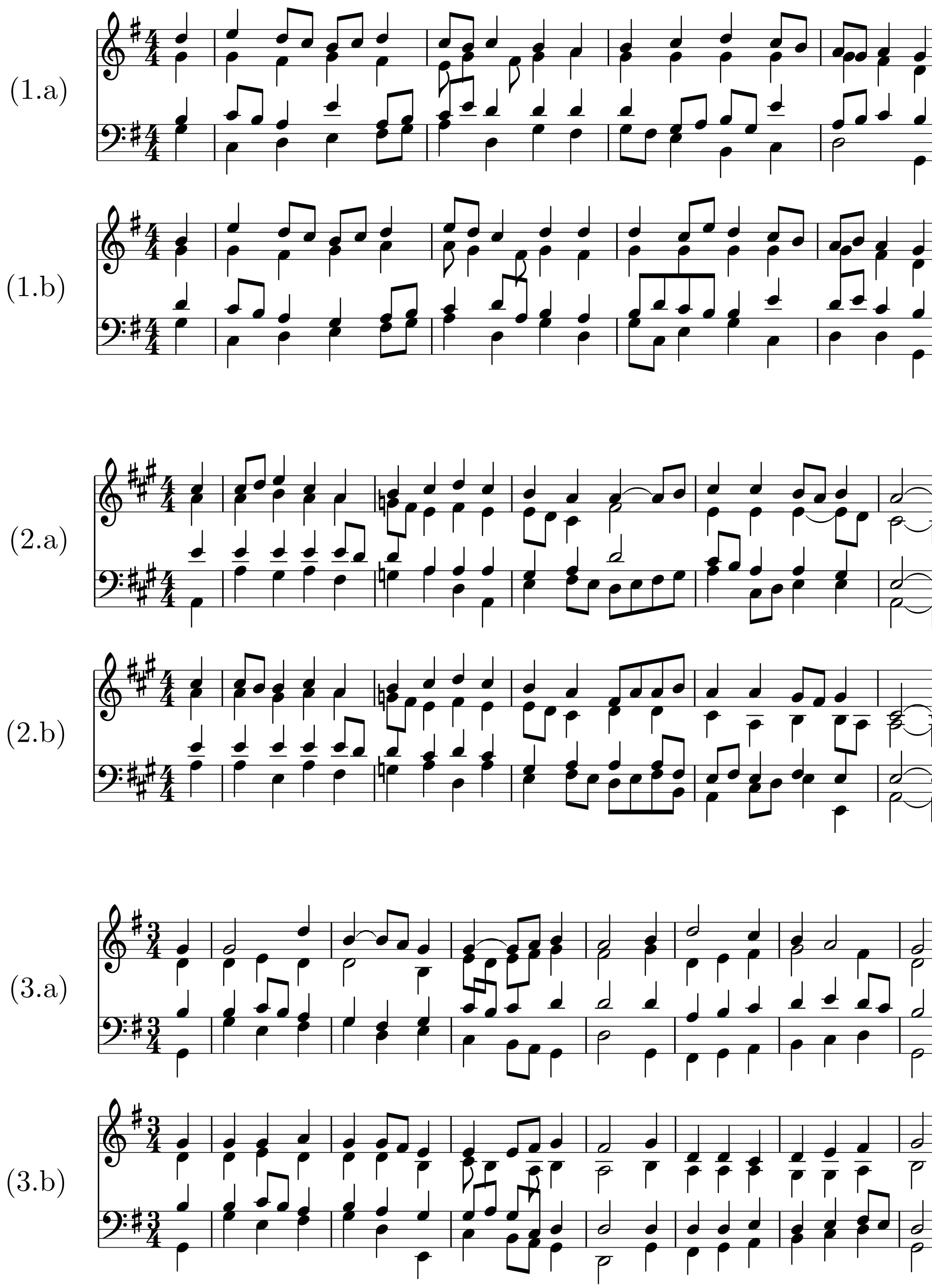



Figure 7: Fist phrases of chorales BWV267 (1.a), BWV253 (2.a) and BWV269 (3.a) and respective transformations (1.b, 2.b and 3.b).

The process tends to reach chorales for which both horizontal and vertical objective functions return low values. The algorithm does not ensure to reach optimal solutions. However, it enables to reach solutions with evaluation values in the same range than the corpus chorales. One of the major drawback of the local search approach is the possibility to be limited by low local optima. Future works include to investigate different approaches trying to overcome this problem including iterated local search, reactive search optimization or tabu search.

Furthermore, the method does not succeed at all aspects of style imitation concerning patterns involving both horizontal and vertical aspects. For example, the model is not able to avoid parallel fifths as it is usually assumed in four-part chorales in the style of J. S. Bach. In this work, the chord alphabet used for the chord segmentation process was reduced to major and minor triads. This reduction requires to reduce dominant seventh chords to major chords, attributing then to the note playing the seventh the harmonic label $\mathscr{k}$ reducing it then to the same level than the other possible non-chord tones. This drawback is partially handled by the slice distribution that attributes high probabilities to slices whose pitches form a dominant seventh chord. However, it could be much improved by enlarging the chord alphabet considered throughout the harmonic analysis task.

In this work, voices of the initial solution are computed with the Viterbi algorithm exclusively. Other strategies that have been experimented include the *greedy* algorithm, which reallocates pitches in each track from left to right, by selecting at each step the pitch improving the best the horizontal evaluation of the current sequence. Though efficient, this algorithm does not guarantee to reach the optimal solution. Alternatively, a random walk process can be applied at every note to generate an initial solution in an non-deterministic way. For each note, a new pitch is sampled from a pitch distribution resulting from the horizontal evaluation function. This method produces solutions having worse horizontal evaluation than the Viterbi solution. However, it provides more diversity as it produces a different solution at each iteration.

Finally, the whole transformation process presented in this work would naturally benefit of some more sophisticated merging rules. This aspect constitutes an important perspective to improve the presented approach.

Acknowledgments

The authors thank Dorien Herremans for valuable discussions and collaboration on this research, and Ray Whorley for providing harmonizations of the melody of BWV267. This research is supported by the project Lrn2Cre8 which is funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

References

Allan, M. and Williams, C. K. (2005). Harmonising chorales by probabilistic inference. *Advances in neural information processing systems*, 17:25–32.

Bigo, L. and Conklin, D. (2015). A viewpoint approach to symbolic music transformation. In *Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research*, pages 56–70, Plymouth.

Biyikoglu, K. M. (2003). A Markov model for chorale harmonization. In *Proceedings of the Triennial ESCOM Conference*, pages 81–84.

Chew, E. (2002). The spiral array: An algorithm for determining key boundaries. In *Music and artificial intelligence*, pages 18–31. Springer, Lecture Notes in Computer Science Volume 2445.

Conklin, D. (2002). Representation and discovery of vertical patterns in music. In Anagnostopoulou, C., Ferrand, M., and Smaill, A., editors, *Music and Artificial Intelligence: Lecture Notes in Artificial Intelligence 2445*, pages 32–42. Springer-Verlag.

Conklin, D. (2010). Discovery of distinctive patterns in music. *Intelligent Data Analysis*, 14(5):547–554.

Conklin, D. (2013). Multiple viewpoint systems for music classification. *Journal of New Music Research*, 42(1):19–26.

Conklin, D. and Witten, I. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73.

Davismoon, S. and Eccles, J. (2010). Combining musical constraints with Markov transition probabilities to improve the generation of creative musical structures. In *Applications of Evolutionary Computation*, pages 361–370. Springer.

Donnelly, P. and Sheppard, J. (2011). Evolving four-part harmony using genetic algorithms. In *Applications of Evolutionary Computation*, pages 273–282. Springer.

Ebcioğlu, K. (1988). An expert system for harmonizing four-part chorales. *Computer Music Journal*, pages 43–51.

Herremans, D. (2014). *Compose=Compute - Computer Generation And Classification Of Music Through Operations Research Methods*. PhD Thesis, University of Antwerp. Universitas, Antwerp.

Herremans, D., Sörensen, K., and Conklin, D. (2014). Sampling the extrema from statistical models of music with variable neighbourhood search. In *Proceedings of the joint International Computer Music Conference — Sound and Music Computing*, pages 1096–1103, Athens.

Hild, H., Feulner, J., and Menzel, W. (1992). Harmonet: A neural net for harmonizing chorales in the style of J.S.Bach. In *Advances in Neural Information Processing Systems*, pages 267–274.

Lerdahl, F. (2001). *Tonal Pitch Space*. Oxford University Press.

Lo, M. and Lucas, S. M. (2006). Evolving musical sequences with n-gram based trainable fitness functions. In *IEEE Congress on Evolutionary Computation*, pages 601–608.

Pachet, F. and Roy, P. (2001). Musical harmonization with constraints: A survey. *Constraints*, 6(1):7–19.

Pardo, B. and Birmingham, W. P. (2002). Algorithms for chordal analysis. *Computer Music Journal*, 26(2):27–49.

Phon-Amnuaisuk, S., Tuson, A., and Wiggins, G. (1999). Evolving musical harmonisation. In *Artificial Neural Nets and Genetic Algorithms*, pages 229–234. Springer.

Raczyński, S. A., Fukayama, S., and Vincent, E. (2013). Melody harmonization with interpolated probabilistic models. *Journal of New Music Research*, 42(3):223–235.

Raphael, C. and Stoddard, J. (2003). Harmonic Analysis with Probabilistic Graphical Models. In *Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR)*, Baltimore, MD.

Rocher, T., Robine, M., Hanna, P., and Strandh, R. (2009). Dynamic chord analysis for symbolic music. In *Proceedings of the International Computer Music Conference*, Montreal, Quebec, Canada.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education.

Sleator, D. and Temperley, D. (2001). The Melisma music analyzer. *Available online at www.link.cs.cmu.edu/music-analysis.*

Suzuki, S. and Kitahara, T. (2014). Four-part harmonization using Bayesian networks: Pros and cons of introducing chord nodes. *Journal of New Music Research*, 43(3):331–353.

Whorley, R. P. and Conklin, D. (2015). Improved iterative random walk for four-part harmonisation. In *International Conference on Mathematics and Computation in Music (MCM 2015), LNAI 9110*, pages 64–70, London. Springer.

Whorley, R. P., Wiggins, G. A., Rhodes, C., and Pearce, M. T. (2013). Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonization problem. *Journal of New Music Research*, 42(3):237–266.