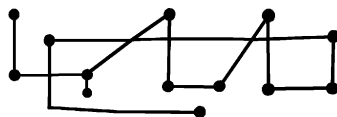


# A COMPUTATIONAL MODEL FOR RULE-BASED MICROTONAL MUSIC THEORIES AND COMPOSITION



TORSTEN ANDERS AND  
EDUARDO R. MIRANDA

## MOTIVATION

**M**ICROTONAL MUSIC is an influential facet of twentieth and 21st century music composition. Composers that contributed significantly to microtonal music include characters as diverse as Julián Carrillo, Ben Johnston, Harry Partch, Horațiu Rădulescu, Karlheinz Stockhausen, James Tenney, Ivan Wyschnegradsky and La Monte Young (in alphabetical order).

As microtonal music opens wide areas of uncharted musical territory, computational support can be very helpful for navigating this

unfamiliar landscape. For example, various software plays back microtonal music, which allows for listening to the music during the composition process. Most sound synthesis programming systems allow for microtonal sound generation (e.g., Csound (Boulangier 2000); SuperCollider (McCartney 2002); Max/MSP (Miller Puckette 2002) and PureData (M. Puckette 1996)). Other systems assist in the development and analysis of microtonal scales, such as Scala by Manuel Op de Coul; CSE by Aaron Hunt; and L'il Miss' Scale Oven by Jeff Scott. These programs can also help to retune various MIDI synthesizers and samplers.

In this paper, however, we are interested in computational support for the composition process itself, a field commonly called computer-aided composition (or algorithmic composition). For example, consider a composer who wants to create a progression of microtonal chords that follows some rules on harmony. Some of her rules are inspired by conventional harmony. For example, a relatively simple rule states that consecutive chords should often share common tones. She conceives other rules for a specific piece or section she is working on (e.g., certain chords should contain specific microtonal intervals). The composer plans to use different textures in her piece. For instance, some sections consist of a melody with accompaniment; other sections are contrapuntal. These textures should always express an underlying microtonal harmony progression. The rules on harmony are complemented by rules on the individual parts. For example, non-harmonic tones may be allowed for more smooth melodic lines, but these are restricted by specific rules in order to make the harmony still recognizable (e.g., passing tones may be allowed). Other rules restrict simultaneous notes (e.g., she may want to avoid unisons and octaves). The composer may also want that each part consists of certain motifs.

Many existing computer-aided composition systems support microtonal pitches, including often-used systems such as Max/MSP & PureData; OpenMusic & PWGL (Assayag et al. 1999; Laurson, Kuuskankare, and Norilo 2009); SuperCollider; JMSL (Larry Polansky, Phil Burk, and Rosenboom 1990; Didkovsky and Philip L. Burk 2001); Common Music (Taube 1997); and Fractal Tune Smithy by Robert Walker. However, complex music theories like the microtonal theories of harmony or counterpoint sketched above are difficult to model with these systems.

Music theories such as harmony or counterpoint are traditionally stated in a modular way by a set of rules, where musical parameters (e.g., a single pitch) are often affected by multiple rules at the same time. This approach allows for a formal description of a complex

network of interval relations in music, which is necessary for theories of harmony or counterpoint. Important interval examples are the sequence of intervals in a melody, the intervals between melodic peaks, the set of intervals between simultaneously sounding notes, the set of intervals that form an implied harmony (which may last longer than individual notes), the intervals between chord roots, the intervals that form an underlying scale (mode), how scales/modes are transposed in modulations and so forth.

The systems mentioned above only partly embrace this complexity. These systems make it very hard to describe a network of interval relations, because they make it hard to affect parameters (e.g., pitches) by more than a single rule at a time. For example, typically either only the horizontal (melodic) or only the vertical (harmonic) dimension is controlled. This restriction is caused by the underlying programming model of these systems, which efficiently map sets of known values to sets of values to compute (as in a function).

Complex music theories that describe a network of interval relations are far more easily formalized using a programming model based on bi-directional relations (as in first-order logic). We propose to use a computational model for microtonal computer-aided composition that stems from logic programming. More specifically, we use constraint programming (Apt 2003), which is more efficient than logic programming, especially for numeric relations (it employs consistency checking or constraint propagation algorithms).

A musical constraint satisfaction problem (CSP) can be seen as a computer program implementing a mathematical model of a music theory. It defines a music representation (score) where some aspects are represented by variables (i.e. unknowns), and relations between these variables are restricted by a set of constraints (rules). For example, the pitches of notes in the score and the underlying harmonic structure may be unknown in a CSP definition. Nevertheless, variables have a domain, that is a set of values they may take in a solution. A constraint solver finds one or more solutions for the problem. In a solution, the domain of each variable is reduced to a single value that is consistent with all its constraints.

Constraint programming has been used before for modeling music theories. Anders and Miranda (Anders and Miranda 2011) provide an extensive survey, which introduces various musical CSPs and systems. However, we are not aware of any previous research that applies constraint programming to modeling microtonal music.

The proposed model and all case studies presented in this paper have been implemented in Strasheela. Strasheela is a constraint-based

computer-aided composition system that allows users to model their own music theories as musical constraint satisfaction problems. Strasheela supports a wide range of music theories, and it provides a rich toolbox that simplifies such definitions.<sup>1</sup>

Our model extends Strasheela's core functionality by a constrainable representation for musical concepts such as scales and harmony. These representations support both the standard Western tuning (twelve tone equal temperament) and microtonal tuning systems. For simplicity, this paper refers to the proposed model as Strasheela (i.e. its implementation).

#### PLAN OF PAPER

The rest of this paper is organized as follows. The next four sections introduce a computational model for composing microtonal music. These sections explain how music is represented in this model, point out the variables in this representation that can be constrained, and discuss the definition and application of constraints.

The subsequent four sections present a number of concrete case studies. These sections demonstrate that the presented model is suitable for implementing microtonal music theories in the disciplines of harmony, melody and counterpoint. The paper ends with a summary.

#### PITCH REPRESENTATION

##### MICROTONAL PITCHES

Composers of microtonal music use various tuning systems, but two approaches to tuning are particularly popular. Keislar interviewed a number of important American composers of microtonal music (Keislar et al. 1991).

Some composers use equal temperaments (ET) that subdivides an interval—most commonly the octave—into a number of equal steps (Blackwood 1991). The ubiquitous example is twelve-tone equal temperament (12-TET) that consists of twelve equal steps per octave. Of the interviewed composers, John Eaton primarily uses 24-TET (his performers are nevertheless free to inflect these quartertones), Joel Mandelbaum uses 31-TET, and Easley Blackwood has explored all ETs from 12-TET to 24-TET.

Other composers prefer just intonation (in the interview, Lou Harrison and Ben Johnston). Just intonation (JI) uses intervals that can be represented by whole number frequency ratios. Small-integer ratios play an important role due to their perceptual quality as consonances (Doty 2002; Johnston 1964). For example, the ratio 3:2 corresponds to the interval of a pure fifth, and 7:4 is a harmonic seventh. The harmonic complexity of JI intervals is often quantified by their odd limit, which—for intervals up to an octave—is the largest odd number in a ratio (Partch 1974). For example, the just minor sixth 8:5 is five odd limit, while the subminor third 7:6 is seven odd limit. The set of all intervals denoted by some odd limit includes lower limit intervals: the set of seven odd limit intervals also includes 8:5. A related concept is the prime limit—the largest prime factor present in any JI intervals, which is more commonly used for discussing scales instead of individual intervals. Note that in contrast to ETs, JI has an infinite number of pitches per octave: the repeated transposition by any JI interval always brings up new pitches.

Further temperaments reduce the total number of pitches and thus the cognitive workload required by closely approximating JI intervals. Such temperaments level JI intervals that are very close to each other (i.e., temper out certain commas) and distribute the resulting pitch shift over the pitches of the temperaments. A well-known example is meantone temperament (Barbour 2004; Leedy 1991), where the syntonic comma is tempered out. For the interval C-E, this temperament does not distinguish whether these tones are one just major third or four octave-transposed fifths (a Pythagorean third) apart. Quartercomma meantone tempers out the difference between these intervals by slightly reducing the size of all fifths but leaving major thirds in JI. Note that the difference between a just and a Pythagorean third is not expressed in common Western music notation either: meantone temperament was not only a compromise for reducing the number of keys on keyboard instruments, but deeply influenced Western musical thinking and compositional practice as well.

The three approaches to tuning systems outlined above are related. ETs can also approximate JI, like the third approach discussed above. For example, 31-TET (Fokker 1955) very well approximates quarter-comma meantone, which in turn closely approximates the intervals of 7-limit JI (odd limit). Regular temperaments are a formal approach to tuning that unifies these three approaches. A regular temperament (Milne, Sethares, and Plamondon 2007) generates all pitches of a tuning with a finite number of intervals called generators.

For example, the generators of meantone are its flat fifth and the octave: all meantone pitches can be generated by repeated transpositions of these intervals. JI requires a unique generator for each prime, while the generator for any ET is its smallest interval.

Finally, well temperaments also approximate JI intervals, but certain intervals and in particular interval combinations in certain keys are tuned more close to their JI counterpart than others (Barbour 2004; L. Polansky et al. 2009). As a result, these tuning systems are irregular temperaments where different keys sound differently.

#### PITCHES AND PITCH CLASSES

Strasheela allows for highly complex music theory definitions, and therefore efficiency is crucial for practical use. Constraint programming provides highly optimized algorithms for solving problems involving integer domains and sets of integers. Our model therefore makes only use of variables with such domains.

The pitches of equal temperaments are naturally expressed by integers. The proposed model and its implementation in Strasheela supports arbitrary equal temperaments. However, which pitch a pitch-integer actually means depends on the chosen number of pitches per octave. By default, pitch-integers are interpreted as midi keynumbers (i.e. 12-TET, where 60 is middle C) when outputting resulting music to sound synthesis formats (e.g., MIDI, Csound) or music notation. Arbitrary other equal temperaments can be chosen. Popular examples are 19-TET, 22-TET, 31-TET, 41-TET, 53-TET and 72-TET, because these approximate certain JI intervals very well. In addition, high pitch resolutions such as cent (1200-TET) or even millicent (120000-TET) are available.<sup>2</sup> In the rest of this paper, whenever we use the term *pitch* in the context of music modeling we refer to pitch-integers.

Pitches are variables in our model, and so are higher-level pitch-related concepts such as pitch classes or scale degrees. Remember that the value of variables can be unknown, and that variables can be constrained. For example, compositional rules applied by users are constraints; these will be discussed later. Some constraints are implicitly applied and are part of the music representation. In particular, the interrelation between the different pitch-related concepts (e.g., pitches and pitch classes) are defined as constraints. Formally, there is no difference between user constraints and these implicit constraints.

Example 1 shows an implicit constraint example: it defines the well-known relation between a pitch on the one hand, and the corresponding pitch class and octave on the other hand for arbitrary equal temperaments.<sup>3</sup> As customary, pitch classes are represented by integers starting from the first note C (pitch class 0) and numbering all tones within a single octave. Following a convention from logic programming, all variables are notated starting with a capital (*psPerOctave*, the pitches per octave is fixed per CSP).

$$\textit{Pitch} = \textit{PitchClass} + (\textit{Octave} + 1) \times \textit{psPerOctave}$$

EXAMPLE 1: CONSTRAINT DEFINITION: RELATION BETWEEN A PITCH AND THE CORRESPONDING PITCH CLASS AND OCTAVE

Pitch operations such as transposition are also formulated as constraints on variables. Pitches of arbitrary ETs are transposed simply by adding a transposition interval. Pitch class transposition “wraps around” at the number of pitches per octave, which is implemented with a modulus constraint. More generally, intervals between variables in the music representation can be seen as intervals in the sense of Lewin, and constraints can express Lewin’s transformations (Lewin 1987).

#### SCALE DEGREES AND CHORD DEGREES

A *degree* is an integer variable that serves quasi as an index into a pitch class sequence. The combination of a degree and its associated accidental (also an integer variable) corresponds to a pitch class, relative to a given pitch class sequence.

The notion of a scale degree is well-known (typically notated by a roman numeral). The C-major scale is represented in 12-TET by the sequence of pitch classes [0, 2, 4, 5, 7, 9, 11]. The third scale degree (roman numeral III) of C-major is the pitch class 4—if the degree accidental is zero (natural). The accidental is a pitch class trans-position interval that serves as an offset from the actual pitch class at the degree position. For example, degree six with accidental -1 (flat) of C-major corresponds to the pitch class 8 (VI $\flat$  of C-major is Ab).

Strasheela's degree constraint is applicable for chords as well. The notion of a chord degree makes it possible to refer to specific chord pitches. For example, the third degree of a major triad is its fifth, with an accidental -1 it is a diminished fifth.

Example 2 formally defines the degree concept for arbitrary ETs. The pitch class at the position *Degree* in the pitch class sequence *PCs* is accessed with the *select*-constraint and bound to the auxiliary variable *NthPC*. The corresponding pitch class *PC* is *NthPC* pitch-class-transposed by the *Accidental*. Again, the degree, its accidental and the pitch class are variables and so are the pitch classes in *PCs*.

$$\begin{aligned} NthPC &= select(PCs, Degree) \\ PC &= (NthPC + Accidental) \bmod pcsPerOctave \end{aligned}$$

EXAMPLE 2: CONSTRAINT DEFINITION: RELATION BETWEEN A PITCH CLASS AND THE CORRESPONDING DEGREE AND ACCIDENTAL

The formal definition of degree transposition is left out for brevity. Again, this transposition “wraps around” at the boundaries of the degree sequence, but the presence of accidentals makes it more complex.

## HIERARCHIC MUSIC REPRESENTATION

So far we only discussed pitch representations. Strasheela's music representation in fact supports arbitrary symbolic score information, and it organizes this information in a hierarchic fashion.

The Strasheela music representation is designed in such a way that ultimately users control what information is contained in the score. For example, variables for chord or scale degrees introduced above are only contained in the score if required. The music representation predefines models for a range of music theory concepts such as notes, intervals, scales, motifs and so forth. Users construct a score by assembling these score objects as required.

Score objects encapsulate a number of attributes. For example, the attributes of a note include its start time, duration, end time, amplitude, its pitch and so forth.<sup>4</sup> The values of these attributes are variables that can be constrained.



Score objects are hierarchically nested: container objects can hold other objects, including other containers. Two kinds of containers are particularly important. A sequential container imposes an implicit constraint that its contained objects follow each other in time, while a simultaneous container constraints that its content runs parallel in time.<sup>5</sup>

Scores can be output into various formats for music notation (e.g., Lilypond, and MusicXML via Fomus) and sound synthesis (e.g., MIDI and Csound). These export facilities can be flexibly customized by programming. For example, users can define how specific score objects are output. Using this approach, the microtonal notation examples in this paper have been created by mapping pitch classes to certain pitch notations with Lilypond.

Although Strasheela's music representation building blocks model their music theory concepts in a highly generic way, users may require additional building blocks that model their own theory. Because Strasheela is a programming system, its music representation is highly extendable by programming means. For example, building blocks are implemented as classes in the object-oriented programming sense, and users can extend them by inheritance. Nevertheless, this paper describes Strasheela's capabilities for musicians and not programmers, and therefore leaves out implementational details. For an extensive discussion of technical details, the interested reader is referred to (Anders 2007).

## REPRESENTING INTERVALS, CHORDS AND SCALES

The Strasheela music representation provides extensive support for analytical information. Score objects such as chords and scales do not sound when a score is played back, but explicit representations of this information greatly simplify the definition of music theories such as harmony or counterpoint. The following paragraphs describe the representation of these score objects. The description focuses on chords and scales, because these objects play a particular important role in the case studies presented later. The representation of intervals uses a similar overall programming approach.

Chord and scale objects contain a number of attributes. For example, important attributes are their *Root* (a pitch class) and *PitchClasses* (a set of pitch classes, which includes the root). In the following we only describe the chord definition; the scale definition is exactly the same (both classes inherit from the same superclass).

Chord objects should be able to distinguish between different chord types (e.g., major vs. minor chord), and these types should be user-definable. This requirement is addressed by a “database” of possible chords. Each database entry contains a number of fields that describe a chord. For example, the following database entry defines the harmonic seventh chord (also called 7-limit dominant seventh) in pseudo code syntax that resembles Strasheela’s Oz syntax.

```
chord(pitchClasses: [4:4 5:4 6:4 7:4]
      root: 4:4
      comment: 'harmonic seventh')
```

Pitch classes in the database can be notated in different formats including pitch class integers, ratios (as in the example above), or symbolic note names. Internally, any format is transformed to pitch class integers for the current number of pitches per octave in order to allow for integer constraint propagation. Nevertheless, the original format is preserved as well and can be used for interpreting solutions (e.g., the original ratios can be used for adaptive just intonation, see below).

Users can define their own databases, or use (and extend) existing databases. Strasheela predefines a large set of database entries. For example, it provides over 50 different chords and 100 scale types for 31-TET (many entries stem from the Scala software).

Entries in the chord database on the one hand and chord objects on the other hand are linked by the object’s attribute *Index*. The meaning of this attribute depends on the current database. If *Index=1*, then the chord object’s type is set to the type of the first chord in the database and so forth. The index is a variable that can be constrained and which effects other chord attributes (e.g., its pitch class set).

While a chord database always contains only a single transposition of a chord, a chord object instance can transpose entries in the database. The attribute *Transposition* measures the transposition amount as a pitch class interval.

As explained above, a Strasheela score is typically nested in a temporal hierarchy of sequential and simultaneous containers. An underlying harmonic analysis of chord and scale objects can be stored in sequential containers that run parallel to the rest of a score (or a

score segment). By default, notes are implicitly related to their simultaneous chords and scales.

It should be mentioned again that Strasheela is highly extendable. The representation scheme of intervals, chords, and scales has been extended to represent additional information. For example, an extension (subclass) of the chord representation stores the chord inversion. Other chord extensions control the relation between chords and an underlying scale, and store the scale degree of a chord root.

Additional information can also be added to the databases for intervals, chords and scales, and these score objects can be constrained accordingly. For example, the dissonance degree of an interval or chord can be added (e.g., Euler's *gradus suavitatis*), or essential pitch classes of a chord can be marked (e.g., a dominant seventh chord is recognized unmistakably by its root, third and seventh, while the fifth is not essential for this chord). Most importantly, users can extend the representation according to their own needs.

#### USER CONSTRAINTS

The previous sections presented a music representation for microtonal music: users model microtonal music theories by assembling a music representation instance with the building blocks provided, and by defining and applying constraints to the variables in this music representation instance. For example, a homophonic chord progression can be modeled with a sequence of chord objects running in parallel to multiple note sequences representing the parts. While some constraints are implicitly applied by the system (e.g., the relation between the pitch and pitch class of a note, as presented above), most constraints are explicitly applied by the user. For example, the user may constrain that the roots of all chord objects are pairwise distinct. A constraint restricts the relation between a set of variables, as has been shown in Example 1 and Example 2 above.

Constraints are freely applied to arbitrary sets of variables. However, a single constraint is often applied multiple times to similar variable sets, and such variable sets can be rather complex. For example, in conventional counterpoint a passing tone that is relatively short and on a weak beat can be dissonant. When this rule is modeled as a constraint, the constraint is applied to every potential passing tone. Also, this constraint must have access to a complex set of variables in order to decide whether a certain note can be dissonant or not. Strasheela supports a convenient and fully generic mechanism for constraint application (Anders and Miranda 2010).

While Strasheela's true power lies in the fact that users can define their own constraints from scratch, for convenience the system pre-defines a wide range of constraints. For example, Strasheela provides many pattern constraints that restrict a sequence of integers in various ways, and it makes generalized versions of many harmonic or counterpoint rules available.

The paper has so far proposed a computational model for microtonal music, which consists of a music representation and support for constraining variables in this representation. We will now present a number of concrete case studies of microtonal music theories implemented with this model. These case studies are situated in different music theory sub-disciplines such as harmony, melody and counterpoint. The case studies also demonstrate different equal temperaments. All music theory models have been implemented with Strasheela.<sup>6</sup>

## HARMONY

The case studies presented in this section model harmony. Above we introduced the notion of analytical chord objects. This representation is an essential building block for the harmony models below.

### DIATONIC CADENCE IN TWELVE-TONE EQUAL TEMPERAMENT

The first case study models a harmony task from common practice music: it creates diatonic cadences. Its music representation consists of a sequence of analytical chord objects and a scale object. The pitch classes of these chords and scales are represented in 12-TET, to start with a well-known tuning system.

The model applies the following constraints to its music representation.<sup>7</sup> The scale object is set to a C-major scale (for simplicity, the root 1/1 has been set to C for all examples in this paper). The chord database specifies only triads (major, minor, diminished, augmented). Only diatonic chords are permitted: the pitch class set of each chord must be a subset of the scale's pitch class set. Consecutive chords in the sequences share common pitch classes (harmonic band, Schoenberg 1986), but consecutive chords must be distinct (i.e. either the chord types, transpositions or both differ). The first and last chords must be equal. Finally, the chord sequence ends in a cadence: the union of the pitch classes of the last three chords constitutes the full

pitch class set of the scale.<sup>8</sup> Also, the root of the last chord is constrained to the root of the scale.

For 5 chords, there exist only three solutions for this music theory model. Strasheela supports finding all solutions of a CSP. One solution is shown in Example 3, where the chord objects are notated using common chord symbols.

C    F       Dm    G       C

EXAMPLE 3: DIATONIC CADENCE SOLUTION, MODELED IN 12-TET

#### 7-LIMIT HARMONY

The next case study introduces 7-limit intervals, and thus goes beyond the scope of common practice harmony. Also, this case study defines a clearly more complex theory of harmony.

We use 31-TET, because this temperament provides close approximations of 7-limit intervals, as has been mentioned above. 31-TET can be notated with the common sharps and flats plus quartertone accidentals. A quartertone sharp raises by one 31-TET step (38.71 cent), a chromatic semitone (e.g., C-C $\sharp$ ) are two steps, and a diatonic semitone (e.g., C-D $\flat$ ) are three steps. The interval 7:4 is represented by the augmented sixth (e.g., C-A $\sharp$ , 25 steps): this interval is only 1.1 cent smaller than the just 7:4.

31-TET allows for very many colorful chord types that are outside conventional harmony. This case study uses two tetrads that are interesting because they consist of only consonant intervals: the harmonic seventh chord (ratios 4:5:6:7) and the subharmonic seventh chord (1/4:1/5:1/6:1/7).

In this case study we do not want to impose any key (unlike the cadence example above): the two chord types can be transposed to any of the 31 tones of the temperament in principle. Nevertheless, various constraints are applied in order to obtain a smoothly connected chord progression.

The most important harmonic constraints are inspired by Schoenberg 1986. Schoenberg distinguishes between so-called ascending progressions (e.g., V-I or III-I; using common roman numerals for notating root progressions), descending progressions (e.g., I-V or III-V), and super-strong progressions (e.g., I-II).

We generalized Schoenberg's root progression guidelines for microtonal music by formalizing his explanation instead of his actual rules (Anders and Miranda 2009a). Briefly summarized, two consecutive chords form an ascending progression if both chords share common pitch classes, but the root of the second chord does not occur in the first chord. In a descending progression, a non-root pitch class of the first chord becomes root in the second chord. A super-strong progression consists of two chords that do not share any pitch classes.

This case study follows Schoenberg's recommendation for the treatment of descending progressions. Ascending progressions are used freely, but a descending progression must be 'resolved,' quasi as a 'passing chord:' in a sequence of three chords C1,C2,C3 the sequence C1,C2 can only be descending if C1,C3 is ascending. This case study completely disallows super-strong progressions to obtain a more smooth progression.

Example 4 shows a solution of this case study. The music representation of this case study consists of four sequences of notes and a sequence of chord objects that run in parallel. The notation of the chord objects extends the common chord symbol notation by symbols for the 7-limit chords. The annotation *harm7* indicates a harmonic seventh chord and *subharm7* a subharmonic seventh chord. This case study shapes its result with several further harmonic constraints. The first and last chord are set to the harmonic seventh over C. All chords are in root position, and the roots of all chords (but the first) are pairwise distinct. 30-70 percent of the chords must be of the type subharmonic seventh chord. Also, the chord types form a

C harm7      D<sup>b</sup> subharm7      A harm7      G<sup>b</sup> harm7  
<sup>3</sup>  
 B<sup>b</sup> subharm7      C<sup>♯</sup> harm7      E<sup>b</sup> harm7      E<sup>♯</sup> subharm7      C harm7

EXAMPLE 4: 7-LIMIT CHORD PROGRESSION NOTATED IN 31-TET

cycle pattern that repeats after every three chords (in Example 4, the pattern is harm7, subharm7, harm7. . .).

The case study also applies various voice-leading rules. The domain of the four parts is restricted to the tessiturae of vocal music. Melodic intervals are restricted to a fifths at most (larger intervals are allowed in principle for the bass, though they do not occur in the presented solution). The harmonic intervals between upper voices are restricted not to exceed an octave (larger intervals are allowed between bass and tenor). Open and hidden parallels of perfect consonances are prohibited. Voice crossing is not allowed. Finally, if consecutive chords share pitch classes, then these are repeated in the same part and octave (a simplified version of Bruckner's "law of the shortest way" (Schoenberg 1969)).

Remember that the theory of harmony implemented in this section is only an example; the strength of the proposed model is in fact that users can implement their own theory. For example, instead of applying the Schoenberg-inspired constraints above, consecutive chords could be connected by a smooth voice leading (Straus 2003). In this alternative approach, the voice leading distance is the minimal sum of pitch class intervals between two chords (not pitch intervals). For example, the voice leading distance between the chords C7 and Ab-maj7 in 12-TET is two ( $C-C=0 + E-Eb=1 + G-Ab=1 + G-G=0$ ).<sup>9</sup> Constraining the voice leading distance to a small value results in a smooth chord progression and vice versa. Such constraint could also be combined with other constraints on harmony, for example, chord pitch classes can be restricted to some underlying scale, or consecutive chords could be constrained to share common pitch classes (harmonic band).

The search process has been randomized in the above and the following case studies. In other words, different solutions are typically found if a CSP is solved multiple times.

#### ADAPTIVE JUST INTONATION

Most musical instruments allow performers to inflect pitches considerably. Such instruments do not have the limitations of static scales, because the musicians can adapt their intonation depending on the context. When computationally modeling microtonal music theories, it is desirable to also support such an adaptive tuning behavior. Sethares (2005) surveys several technical approaches to adaptive JI.

A classical approach to adaptive JI has been proposed by Nicola Vicentino in the sixteenth century (Vicentino, Maniates, and Palisca 1996). In Vicentino's approach, two manuals of the harpsichord are tuned to 1/4-comma meantone (ideally each with nineteen notes per octave), but one manual is tuned a 1/4-comma higher than the other. This arrangement allows a musician to justly intonate all triads that are available in meantone: the just major thirds over any root are provided by meantone itself, and the narrow meantone fifths and minor thirds can be adjusted by using the corresponding tones of the other manual. Note that this approach tunes chords in JI, but the intervals between chord roots are still meantonetempered, which avoids the problem of pitch drift, common for JI.

Strasheela supports an approach to adaptive JI that generalizes Vicentino's idea. As has been explained above, chord database entries can be defined with JI ratios in Strasheela. These ratios can be used for tuning. In Strasheela's adaptive JI, chord roots are tuned according to the current ET (or alternatively a given static tuning table). The tuning of other chord tones, however, is adapted according to the JI ratios of the underlying harmony.

Example 5 retunes the example presented previously in Example 4. The examples below the staves report how the notes are retuned with respect to 31-TET (measured in cents). All chords are in root position in this example, and so the offset values for the bass notes are all zero cent. The first chord is a harmonic seventh chord (4:5:6:7). In 31-TET, a fifth is flat by 5.2 cent, a major third is sharp by 0.8 cent, and a

| Measure             | 1            | 2           | 3           | 4            | 5     |
|---------------------|--------------|-------------|-------------|--------------|-------|
| Treble Clef         | -0.8c, -4.1c | 5.2c, 1.1c  | 1.9c, -0.8c | 1.1c, 1.1c   | -0.8c |
| Bass Clef           | 5.2c, 1.9c   | 1.1c, -0.8c | 1.1c, 5.2c  | -0.8c, -4.1c | 5.2c  |
| Bass Clef (Offsets) | 1.1c, 1.1c   | -0.8c, 5.2c | -4.1c, 1.1c | 5.2c, 1.9c   | 1.1c  |
| Bass Clef (Roots)   | 0c, 0c       | 0c, 0c      | 0c, 0c      | 0c, 0c       | 0c    |

EXAMPLE 5: ADAPTIVE JI OF THE 7-LIMIT CHORD PROGRESSION PRESENTED IN EXAMPLE 4, CENT OFFSET VALUES RELATIVE TO 31-TET



harmonic seventh is flat by 1.1 cent. The adaptive JI algorithm corrects all these intervals. If a strictly just result is not intended, for example because some slow beating is preferred, then it is also possible to specify how far the static temperament should be adapted towards JI (e.g., only half-way).

#### CHORD FIGURATION

The proposed model and its Strasheela implementation supports arbitrary musical textures. While the harmony case studies of the previous section consisted of plain homophonic chord progressions, this section demonstrates how a single chord can be elaborated with chord figurations.

The present case study uses tones of a 7-limit JI chord from La Monte Young's *The Well-Tuned Piano*, called the Lost Ancestral Lake Region chord (Gann 1993). It is a subminor seventh chord with an added major second (ratios 12:14:18:21:27).<sup>10</sup> The intervals of the *The Well-Tuned Piano* are closely approximated by 41-TET. The 41-TET error is very small for 7:4 (-2.97 cent), and extremely small for 3:2 (0.48 cent). This case study therefore uses 41-TET pitch classes.

Example 6 shows the Lost Ancestral Lake Region chord transposed to C. The chord is written in the Extended Helmholtz-Ellis JI Pitch notation (EHE notation), proposed by Marc Sabat and Wolfgang von Schweinitz (Sabat and Tenney 2008). This notation indicates Pythagorean tuning (3-limit) with the conventional intervals, and introduces a new accidental for a prominent comma of every new prime limit. The accidental for the septimal comma 64:63 resembles the digit seven (this comma indicates the difference between a



EXAMPLE 6: LA MONTE YOUNG'S 7-LIMIT CHORD LOST ANCESTRAL LAKE REGION IN EXTENDED HELMHOLTZ-ELLIS JI PITCH NOTATION (EHE NOTATION); THE ACCIDENTALS THAT RESEMBLE THE DIGIT SEVEN TURNED UPSIDEDOWN SHIFT THE PITCH DOWN BY A SEPTIMAL COMMA (64:63)

Pythagorean seventh—two stacked fourths—and a harmonic seventh). Although EHE notation was originally proposed for JI, it can be used for various temperaments as well, much like Sagittal notation (Secor and Keenan 2004), which has been explicitly designed for this purpose.

Example 7 shows a solution of the present case study. The texture of this example is represented with a hierarchic music representation, as discussed above. For example, the dyads of the upper staff are represented by a number of simultaneous containers with two notes, which in turn are contained in a sequential container.

The chord figuration is shaped by several constraints; the most important constraints are listed below. The sequence of directions of melodic pitch intervals (the pitch contour, Larry Polansky and Bassein 1992) is individually constrained for every part. In the two parts in the upper staff, the pitch contours follow the same given envelope (pitch repetitions are disallowed for the upmost part). The contour of the bass continuously descends, and ends in the chord root. Other constraints control simultaneous pitches. At any time, three different pitch classes are present. Also, at least 50 percent of the upper-staff dyads are 7-limit consonances (in other words, the number of 3-limit intervals is restricted).

Note that the latter constraint expresses a restriction on JI ratios (i.e. their limit), even though all pitches are internally represented by an ET (41-TET). Such JI-constraints are possible for ETs that uniquely map the corresponding ratios to pitch classes of the ET. 41-TET provides unique pitch classes for many 7-limit intervals.

The image displays two systems of musical notation. The first system consists of two staves: a treble clef staff on top and a bass clef staff on the bottom. The time signature is 3/4. The notation uses EHE notation, with notes and chords represented by small symbols and lines. The second system also consists of two staves: a treble clef staff on top and a bass clef staff on the bottom. The time signature is 3/4. The notation uses EHE notation, with notes and chords represented by small symbols and lines. A small '3' is written above the first measure of the second system's treble staff.

EXAMPLE 7: FIGURATION OF A 7-LIMIT JI-CHORD (EHE NOTATION)

## MELODY

This case study examines 7-limit melody composition. In the proposed approach a melody expresses an implicit harmony, and so this case study makes use of harmony definitions presented earlier. In addition, it introduces non-harmonic tones and their treatment as well as formal aspects such as motifs.

The present study uses a 7-limit scale that Erlich (1998) proposed together with 7 related decatonic scales. Example 8 shows the static symmetrical major scale, notated in EHE for 22-TET. Note that 22-TET does not temper out the syntonic comma. Arrows attached to accidentals indicate a shift by a tempered syntonic comma.<sup>11</sup> Instead, a temperament for Erlich's decatonic scales must temper out the two commas of 64:63 and 50:49. For example, the major second of these scales serves as both Pythagorean 9:8 and septimal 8:7 (64:63 comma tempered out), and its tritone represents both 7:5 and 10:7 (50:49 comma vanished). Temperaments that temper out these commas are called Pajara on the Alternate Tunings Mailing List, and Strasheela's tuning table can be set to such a temperament. Instead, we are using 22-TET for this case study, which also tempers out these commas (and other commas as well). This scale generalizes several properties found in the well known diatonic scales for the 7-limit. For example, there are only two different step sizes: a small step (marked s in Example 8; pitch class interval 2 in 22-TET), and a large step (marked L, pitch class interval 3). The sequence of s and L explains why this scale is called symmetrical. Like diatonic scales are constructed from two tetrachords that subdivide a fourth (4:3) into four tones (Chalmers 1993), this scale is constructed from two "pentachords" (marked by brackets) that subdivide a fourth into five tones (however, these pentachords are a tritone away from each other). Finally, consonant 7-



EXAMPLE 8: THE *STATIC SYMMETRICAL MAJOR SCALE* PROPOSED BY PAUL ERLICH IN 22-TET, NOTATED IN EHE (ARROWS SHIFT THE PITCH BY A TEMPERED SYNTONIC COMMA)

limit tetrads that only consist of scale tones can be constructed on all but two scale degrees (the remaining two degrees carry augmented triads). Example 9 shows a melody solution. For clarity, its implicit harmony is notated on the second staff. This staff depicts the actual chord objects: chord pitch class sets are notated as grace notes and chord roots as “normal” notes that also indicate the duration of the underlying harmony. The underlying harmony forms a quasi plagal cadence of 7-limit tetrads, which only use tones of the symmetrical major scale. Consecutive chords are connected by common tones.

Although the underlying harmonic structure resembles a traditional cadence, the chords and even more so the scale employed are certainly not conventional. It was therefore important to add constraints that ensure harmonic clarity. For example, all tones of the underlying harmony are present in the melody. Further, the melody features non-harmonic tones (marked by crosses), but strict constraints ensure that such tones do not affect the harmonic clarity. A non-harmonic tone cannot follow another non-harmonic tone; and they are always stepwise approached and resolved (this particular solution shows only passing tones). The ornamental character of non-harmonic tones is further safeguarded by a constraint that takes note durations into

The image displays a musical score for Example 9, consisting of three systems of music. Each system has a melody line on a treble clef staff and a lower staff showing the underlying harmony. The first system is labeled "Implicit harmony" and shows a melody with non-harmonic tones marked with 'x' and a lower staff with "harmonic 7th" chords. The second system is labeled "4" and shows a melody with non-harmonic tones marked with 'x' and a lower staff with "harmonic 7th" chords. The third system is labeled "8" and shows a melody with non-harmonic tones marked with 'x' and a lower staff with "subharmonic 7th" and "harmonic 7th" chords. Bar lines separate the systems, and the lower staff shows the underlying harmony.

EXAMPLE 9: MELODY IN STATIC SYMMETRICAL MAJOR IN 22-TET USING EHE NOTATION; BAR LINES SEPARATE EXPLICITLY DECLARED MOTIFS; THE LOWER STAFF SHOWS THE UNDERLYING HARMONY

account: a non-harmonic note must be preceded and followed by a note that is at least as long as the non-harmonic note itself.

Compared with disciplines like harmony and counterpoint, melody composition has been addressed far less frequently in the literature. An important reason for this difference may be that melody composition can be less formalized than harmony. In the proposed approach we therefore do not try to fully formalize melody composition. Instead, important motivic aspects are defined manually, but the actual melody pitches and the implicit harmony found by the computer (Anders and Miranda 2009b).

The melody of this case study is constructed from two motifs, for which specific features are composed manually. For example, note durations and the pitch contour (interval directions) are given for motif *a* (e.g., measure 1 in Example 9). The motif declaration also states where skips and steps occur in the motif. This declaration still allows for considerable flexibility: motifs can be transposed freely and the actual size of skips and steps is variable as well. In addition, motif variations are defined by a function that changes the motif declaration (e.g. the durations and the contour): the variation used here removes one or more of the shortest motif notes (e.g., compare measures 4–7). Technically, motifs have been implemented as sub-CSPs and the full CSP has been defined by assembling these sub-CSPs in time using Strasheela’s temporal containers.

Obviously, a faster harmonic rhythm is possible by preserving the harmonic clarity, if we add an accompaniment, which also allows to unambiguously present further chords, for example, the subminor seventh (12:14:18:21) as in Example 10.<sup>12</sup> A larger set of possible chords also allows for a more refined theory of harmony: in this case only ascending progressions are permitted (see the 31-TET case study

The image shows a musical score with two staves. The upper staff contains a melody in a 22-TET system, with notes marked with 'x' above them. The lower staff, labeled 'Implicit harmony', shows three chords: 'harmonic 7th', 'subharmonic 7th', and 'subminor 7th'. The chords are connected by a horizontal line, indicating a progression. The notation includes various microtonal intervals and accidentals.

EXAMPLE 10: MELODY WITH ACCOMPANIMENT IN STATIC SYMMETRICAL MAJOR  
(22-TET, EHE NOTATION)

above). Also, further melodic constraints can be applied with such a more flexible pitch set: in this CSP all intervals between the melodic peaks of motifs are constrained to be steps upward.

### COUNTERPOINT

Finally, we model 2-part microtonal counterpoint. Like the previous case study, this section uses the 7-limit scale static symmetrical major (Example 8), and is tuned in 22-TET.

The study implements “harmonic counterpoint”: the contrapuntal lines express an underlying harmonic structure (as Baroque counterpoint does, in contrast to Renaissance counterpoint). This case study thus again draws on microtonal harmony definitions discussed before.<sup>13</sup>

Example 11 shows a solution, the underlying harmony is explicitly notated as explained above. The music representation consists of two sequential containers with notes, and a chord sequence, all contained in a simultaneous container. The constraints on the underlying harmony of the first melody example are in force again (consonant tetrads, only using scale pitch classes and chords are connected by common pitch classes), complemented by a few further harmonic constraints. At any time, two different pitch classes are played. On a strong beat, only chord tones must be played. Also, harmony is restricted to root positions or second inversions, which has been implemented by constraints between chord degrees and bass notes. A local minimum of the pitches in the bass must be either chord degree 1 (i.e. the root) or 2 (the third of the two possible chords harmonic

The image displays musical notation for Example 11. It consists of three staves. The top two staves are in 22-TET tuning and show two melodic lines. The bottom staff is labeled "Implicit harmony" and shows two chords: "harmonic 7th" and "subharmonic 7th".

EXAMPLE 11: 2-PART COUNTERPOINT IN STATIC SYMMETRICAL MAJOR (22-TET, EHE NOTATION)

seventh or subharmonic seventh). For example, the second note in measure 1 of Example 11 is a local minimum, and it is the third of the underlying harmonic seventh chord over C.

The melody is constrained as follows. Melodic intervals form a “ballistic curve”: a skip up (down) is followed by a smaller (larger) skip/step in the same direction or a step into the opposite direction. No direct pitch repetitions are permitted, and the pitch contour must undulate, i.e. the number of intervals going in the same direction is restricted by a lower bound (here 3) and an upper bound (8). Nevertheless, directly after the first and before the last note a direction change is permitted as well. The treatment of non-harmonic notes is restricted as in the melody case study above, but a few additional constraints are applied to improve the harmonic clarity. For example, simultaneous non-harmonic tones must be consonant to each other. Also, if one voice resolves a dissonance (a non-harmonic tone, again marked by crosses), then the other voice cannot start a new dissonance at the same time and that way masks the dissonance resolution (in the solution shown, it so happens that no simultaneous dissonances occur). Besides, neither open nor hidden parallels of perfect consonances are permitted.

Some constraints on the rhythm have been applied as well. However, as this paper addresses microtonal pitches, the rhythmic structure is very simple, and only few rhythmic constraints have been used. The domain for all note durations consists of the note values half note, quarter note and eighth note, and a 4/4 meter is set. In order to make the metric structure more clear, syncopations over bar lines are not allowed, and the first note value of a bar must be at least a quarter note.

## SUMMARY AND DISCUSSION

This paper presented a computational model for microtonal music theories and composition that makes use of the constraint programming paradigm. The fundamental layer of this model is its pitch representation, which introduces variables for *pitches*, *pitch classes*, and (chord or scale) *degrees*. This pitch representation supports arbitrary equal temperaments (ET) including the common twelve-tone equal temperament (12-TET). We proposed constraints that define the relation between these representations; as well as define transpositions for each representation. Further, we proposed a constrainable music repre-

sentation for higher level pitch-related concepts such as *chord* and *scale* objects.

The model has been implemented in Strasheela, so that this model can be used together with other Strasheela features. For example, Strasheela's constrainable representation of temporal score object hierarchies is available.

This paper demonstrated the proposed model in a number of case studies that implement microtonal theories of harmony, melody and counterpoint. These case studies also showed how the model supports various ETs. We modeled a diatonic cadence in 12-TET; a 7-limit harmony progression in 31-TET and adaptive just intonation (JI); a chord figuration of a chord from La Monte Young's *The Well-Tuned Piano* in 41-TET; and finally a melody and harmonic counterpoint with Erlich's static symmetrical major scale in 22-TET.

A possible criticism questions computer-aided composition of microtonal music in general. While we have centuries of knowledge for the rules that can be applied for 12-TET composition, how can we know and define rules that lead to sensible music for tuning systems that have been rarely or not at all explored so far?

We don't think that this problem really exists for a composer who is interested in microtonal music to explore fresh resources. In our experience—regardless whether we compose for 12-TET or microtonal music—selecting and defining rules is actually an integral part of the composition process. When evaluating the resulting music by listening we often heard some shortcoming, which we then tried to address by a new rule. For example, the 7-limit chord figuration in Example 7 originally contained many 3-limit intervals such as open fifths between the two upper voices, which we felt sounded rather “empty.” After we added a rule that required many 7-limit intervals between these voices, the resulting sound became clearly more rich, even fancy. Also, we hope the examples above demonstrate that we do not necessarily need to start from scratch when composing microtonal music. It is possible to apply certain conventional rules virtually unchanged (e.g., some melodic rules such as a treatment of non-harmonic tones). Other conventional rules are possibly generalized for microtonal music, as we showed with our generalized version of Schoenberg's directions for better progressions. In general, it is often desirable that musical results display some consistency. For example, we may want that a motif sequence shows some pattern, or we may want to avoid that the dissonance degree of a chord progression jumps wildly back and forth. Compositional rules can help to enforce consistency regarding various



musical aspects, and such rules are no more difficult to formulate for microtonal music than for 12-TET.

Some limitations of the proposed model should be mentioned. Because this model integrates pitch classes, it only supports equal temperaments that repeat each octave. A counter example is the Bohlen- Pierce scale that repeats every  $3/1$  interval, called a tritave (Mathews and Pierce 1989). Nevertheless, octave-repeating scales are particularly common. Strasheela itself has also some limitations. For example, arbitrary musical textures can be expressed with its music representation, but constraining the hierarchic nesting of score objects is severely restricted (Anders 2007).

The presented model supports ETs only: we are currently working on an extension for arbitrary regular temperaments. Remember that regular temperaments can also express arbitrary just intonations. Regular temperaments will be represented by a subset of the pitches (pitch classes) of an equal temperament with a high-resolution such as 1200-TET (cent) or even 120000-TET (millicent).

In summary, the case studies illustrate that the presented model allows the user to apply various constraints to a microtonal music representation. We shaped each case study in its own way with a certain set of constraints. Obviously, these constraint sets are only examples, very different constraints may be applied if different results are intended. In fact, this capability is the major strength of Strasheela.

#### ACKNOWLEDGMENTS

We are grateful for helpful comments by John H. Chalmers, John Rahn and William R. Sethares on drafts of the paper. This work was supported by the EPSRC project ‘Learning the Structure of Music’ (LeStruM), EP/D062934/1.

## NOTES

1. Strasheela is freely available at <http://strasheela.sourceforge.net/>.
2. In addition, Strasheela supports user-defined tuning tables, where the pitches of an ET are mapped to actual pitches. The format of this tuning table is similar to the scale file format of the Scala program. Each pitch within an octave is declared either as a float (measured in cent) or a frequency ratio for JI intervals.
3. This formula implements the convention that middle C is situated in Octave 4, hence the added 1.
4. Having explicit representations of the start time, duration, and end time at the same time is not redundant as these pieces of information can be undetermined.
5. There can be rests between objects in a sequential container or before objects in a simultaneous container, either represented by an explicit rest object or the offset parameter supported by all temporal object (e.g., notes, sequential and simultaneous containers).
6. The full source code of all music theory case studies presented in this paper is available at <http://strasheela.sourceforge.net> (last accessed March 19, 2011).
7. For brevity, we present any constraints of theory model case studies only in English. Please refer to the provided source code for the full formal details.
8. This constraint implements a particular strict notion of a cadence, where all scale notes must sound. A less strict version requires that only pitch classes which distinguish a scale among all other likely scales are sufficient (e.g., the pitch classes G, B, and F are sufficient to distinguish C-major between all major scales) (Rothenberg 1977).
9. Only the minimal intervals towards the pitch classes of the second chord are taken into account; the B $\flat$  of the first chord is ignored.
10. Young's ratios are treated as pitch classes in this case study that can be octave transposed. For example, both the major second and the major ninth are possible. Nevertheless, the root is preserved and specially treated, see the rules for the bass.

11. When notating 22-TET with conventional accidentals for 3-limit intervals and accidentals that indicate a syntonic comma shift for 5-limit intervals, then pairs of enharmonically equivalent pitches occur. For example,  $C^+$  is the same pitch as  $D^b$  in 22-TET, and so are  $C^\#-$  and  $D^{b+}$  (plus/minus indicating a syntonic comma up/down).
12. The enharmonic spelling of Example 8 is kept in Example 10 as well, so the scale pitches can be more easily recognized. However, doing so compromises the enharmonic spelling of the subminor seventh chord.
13. Strasheela also supports counterpoint where no explicit underlying harmonic structure is defined. For example, the Strasheela website (<http://strasheela.sourceforge.net/>) presents examples that implement Fuxian first species counterpoint and florid counterpoint in 12-TET. The approaches shown there can be used for modeling microtonal music as well.

## REFERENCES

- Anders, Torsten. 2007. "Composing Music by Composing Rules: Design and Usage of a Generic Music Constraint System." Ph.D. diss., School of Music and Sonic Arts, Queen's University Belfast.
- Anders, Torsten, and Eduardo R. Miranda. 2011. "A Survey of Constraint Programming Systems for Modelling Music Theories and Composition." *ACM Computing Surveys*, 43(4) (forthcoming).
- . 2009a. "A Computational Model that Generalises Schoenberg's Guidelines for Favourable Chord Progressions." In *6th Sound and Music Computing Conference*, ed. Fabien Gouyon, Alvaro Barbosa, and Xavier Serra. Porto, Portugal. <http://smc2009.smcnetwork.org/programme/pdfs/151.pdf>.
- . 2009b. "Interfacing Manual and Machine Composition." *Contemporary Music Review* 28, no. 2: 133–147.
- . 2010. "Constraint Application with Higher-Order Programming for Modeling Music Theories." *Computer Music Journal* 34, no. 2 (June 1): 25–38.
- Apt, Krzysztof R. 2003. *Principles of Constraint Programming*. Cambridge: Cambridge University Press.
- Assayag, Gerard, Camilo Rueda, Mikael Laurson, Carlos Agon, and Olivier Delerue. 1999. "Computer Assisted Composition at IRCAM: From PatchWork to OpenMusic." *Computer Music Journal* 23, no. 3: 59–72.
- Barbour, James Murray. 2004. *Tuning and temperament*. Courier Dover Publications.
- Blackwood, Easley. 1991. "Modes and Chord Progressions in Equal Tunings." *Perspectives of New Music* 29, no. 2: 166–200.
- Boulanger, Richard. 2000. *The Csound Book. Perspectives in Software Synthesis, Sound Design, Signal Processing, and Programming*. Cambridge, MA: MIT Press.
- Chalmers, John. 1993. *The Divisions of the Tetrachord*. Hanover NH: Frog Peak Music.
- Didkovsky, Nick, and Philip L. Burk. 2001. "Java Music Specification Language, an introduction and overview." In *ICMC*.

- Doty, David B. 2002. *The Just Intonation Primer. An Introduction to the Theory and Practice of Just Intonation*. Third edition. San Francisco, CA: Just Intonation Network.
- Erlich, P. 1998. "Tuning, tonality, and twenty-two-tone temperament." *Xenharmonikón* 17 (Spring). <http://lumma.org/tuning/erlich/erlich-decatonic.pdf>.
- Fokker, Adriaan Daniel. 1955. "Equal Temperament and the Thirty-one-keyed Organ." *The Scientific Monthly* 81, no. 4: 161–166.
- Gann, K. 1993. "La Monte Young's The Well-Tuned Piano." *Perspectives of New Music* 31, no. 1: 134–162.
- Johnston, Ben. 1964. "Scalar Order as a Compositional Resource." *Perspectives of New Music* 2, no. 2: 56–76.
- Keislar, Douglas, Easley Blackwood, John Eaton, Lou Harrison, Ben Johnston, Joel Mandelbaum, and William Schottstaedt. 1991. "Six American Composers on Nonstandard Tunings." *Perspectives of New Music* 29, no. 1: 176–211.
- Laurson, Mikael, Mika Kuuskankare, and Vesa Norilo. 2009. "An Overview of PWGL, a Visual Programming Environment for Music." *Computer Music Journal* 33, no. 1: 19–31.
- Leedy, Douglas. 1991. "A Venerable Temperament Rediscovered." *Perspectives of New Music* 29, no. 2: 202–211.
- Lewin, David. 1987. *Generalized Musical Intervals and Transformations*. Yale University Press.
- Mathews, Max V., and John R. Pierce. 1989. "The Bohlen-Pierce Scale." In *Current Directions in Computer Music Research*. MIT Press.: 165–173.
- McCartney, James. 2002. "Rethinking the Computer Music Language: SuperCollider." *Computer Music Journal* 26, no. 4 (Winter).
- Milne, Andrew, William Sethares, and James Plamondon. 2007. "Isomorphic Controllers and Dynamic Tuning: Invariant Fingering over a Tuning Continuum." *Computer Music Journal* 31, no. 4 (December 1): 15–32.
- Partch, Harry. 1974. *Genesis of a Music: An Account of a Creative Work, Its Roots and Its Fulfillments*. 2nd ed. New York, NY: DaCapo Press.

- Polansky, L., D. Rockmore, K. Johnson, D. Repetto, and W. Pan. 2009. "A Mathematical Model for Optimal Tuning Systems." *Perspectives of New Music* 47, no. 1 (Winter): 69–110.
- Polansky, Larry, and Richard Bassein. 1992. "Possible and Impossible Melody: Some Formal Aspects of Contour." *Journal of Music Theory* 36, no. 2: 259–284.
- Polansky, Larry, Phil Burk, and David Rosenboom. 1990. "HMSL (Hierarchical Music Specification Language): A Theoretical Overview." *Perspectives of New Music* 28, no. 2 (Summer): 136–178.
- Puckette, M. 1996. "Pure Data: another integrated computer music environment." *Proceedings of the Second Intercollege Computer Music Concerts*: 37–41.
- Puckette, Miller. 2002. "Max at Seventeen." *Computer Music Journal* 26, no. 4.
- Rothenberg, D. 1977. "A model for pattern perception with musical applications part II: The information content of pitch structures." *Theory of Computing Systems* 11, no. 1: 353–372.
- Sabat, Marc, and James Tenney. 2008. "Three Crystal Growth Algorithms in 23-limit Constrained Harmonic Space". *Contemporary Music Review* 27, no. 1: 57.
- Schoenberg, Arnold. 1969. *Structural Functions of Harmony*. Second revised. New York: W. W. Norton.
- . 1986. *Harmonielehre*. 7th ed. Wien: Universal Edition, Rev. ed. 1922. Trans. by Roy Carter as *Theory of Harmony*. Berkeley and Los Angeles: University of California Press. 1978.
- Secor, George D., and David C. Keenan. 2004. Sagittal. "A Microtonal Notation System." *Xenharmonikón, An Informal Journal of Experimental Music* 18. <http://www.sagittal.org/sagittal.pdf>.
- Sethares, William A. 2005. *Tuning, Timbre, Spectrum, Scale*. 2nd ed. London: Springer.
- Straus, Joseph N. 2003. "Uniformity, Balance, and Smoothness in Atonal Voice Leading." *Music Theory Spectrum* 25, no. 2 (Autumn): 305–352.
- Taube, Heinrich. 1997. "An Introduction to Common Music." *Computer Music Journal* 21, no. 1: 29–34.

Vicentino, Nicola, Maria Rika Maniates, and Claude V. Palisca. 1996. *Ancient Music Adapted to Modern Practice*. New Haven, CT: Yale University Press. Translation of Vicentino (1555). *L'antica musica ridotta alla moderna prattica*.