Journal of New Music Research

ISSN: 0929-8215 (Print) 1744-5027 (Online) Journal homepage: http://www.tandfonline.com/loi/nnmr20

Music Generation from Statistical Models of Harmony

Raymond P. Whorley & Darrell Conklin

To cite this article: Raymond P. Whorley & Darrell Conklin (2016) Music Generation from Statistical Models of Harmony, Journal of New Music Research, 45:2, 160-183, DOI: 10.1080/09298215.2016.1173708

To link to this article: http://dx.doi.org/10.1080/09298215.2016.1173708

Published online: 27 Apr 2016.

Journal of New Music Research, 2016
Vol. 45, No. 2, 160–183, http://dx.doi.org/10.1080/09298215.2016.1173708

Routledge
Taylor & Francis Group

Music Generation from Statistical Models of Harmony

Raymond P. Whorley[1] and Darrell Conklin[2,3]

[1]*The Open University, UK;* [2]*University of the Basque Country UPV/EHU, Spain;* [3]*IKERBASQUE, Basque Foundation for Science, Spain*

(Received 15 August 2015; accepted 24 March 2016)

Abstract

This article investigates the problem of sampling from statistical models of music, motivated by the fact that music generated by random walk is generally atypical in style and of vastly inferior quality compared with pieces in the corpus. Specifically, we employ a multiple viewpoint system of four-part harmony, in conjunction with a small set of general rules of harmony, to evaluate an improved iterative random walk technique which efficiently finds low cross-entropy (high probability) solutions. By first using standard iterative random walk, we show that a range of very high cross-entropy harmonizations is generated. Furthermore, we demonstrate a relationship between cross-entropy and the number of rule violations; that is, there are fewer violations at lower cross-entropy. Motivated by this result, the application of an improved sampling technique was used to produce solutions of very low cross-entropy, confirming that the correlation between cross-entropy and rule violations extends into the low cross-entropy region. Various diversity measures, such as the number of non-diatonic notes per 100 melody notes, also exhibit trends with respect to cross-entropy. These findings are likely to influence future research involving the generation of samples from statistical models, in the realm of music and beyond.

Keywords: machine learning, harmonization, random walk, probability threshold, multiple viewpoint system, statistical model

1. Introduction

Music generation can be viewed as a process of drawing samples from statistical models learned from a style (Conklin, 2003). A statistical model of music assigns overall probabilities to pieces of music by calculating the product of probabilities (determined by the model) of individual *events* within those pieces. For existing music, this process is called *prediction*; whereas *generation*, the focus of this article, involves sampling from the model's probability distributions to create novel music. Another use for such models is *classification*, where multiple models represent different styles or genres of music. A piece of music is predicted by each model, and is classified as being in the style represented by the model assigning the highest overall probability. In each case, statistics are drawn from a corpus of music in a homogeneous style by means of machine learning techniques.

Statistical models of music have generally employed *context models* of various types, including *Markov models* (e.g. Biyikoğlu, 2003; Ponsford, Wiggins, & Mellish, 1999), *hidden Markov models* (e.g. Allan & Williams, 2005) and *Bayesian networks* (e.g. Raphael & Stoddard, 2003; Suzuki & Kitahara, 2014). The general idea is that prediction probabilities are conditioned on a relatively small context of notes or chords; for example, given the short contextual sequence C4 D4 E4, what is the probability of the next note being F4? In our present work, we use a hierarchical model employing *multiple viewpoint systems* (Conklin & Witten, 1995), which have a proven track record in many aspects of music informatics. For example, they are used for the modelling of melodic expectation (Cherla, Weyde, & d'Avila Garcez, 2014; Pearce & Wiggins, 2006), in models to predict musical phrase and section grouping (Pearce, Müllensiefen, & Wiggins, 2010; Potter, Wiggins, & Pearce, 2007), for tabla sequence generation (Chordia, Sastry, & Şentürk, 2011), for folk music classification (Conklin, 2013), for harmonizing melodies (Whorley, Wiggins, Rhodes, & Pearce, 2013), and for the prediction (Rohrmeier & Graepel, 2012) and classification (Hedges, Roy, & Pachet, 2014) of chord sequences.

A common way to generate music from a statistical model is by *random walk*. A random walk repeatedly adds events (which can be notes, chords, segments, etc.) to a growing

Correspondence: Dr Raymond Whorley, Department of Computing and Communications, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK. E-mail: raymond.whorley@open.ac.uk

event sequence, choosing at each stage from the currently constructed probability distribution of events (conditioned on the prevailing context). Random walk, while simple to implement and efficient enough to be applied in real-time applications, is known to suffer from several problems. The main problem is that the procedure, when viewed as sampling from the space of solutions over event sequences, can only be expected to produce (except in the infinite theoretical limit of multiple samples) sequences in close proximity to the mean of the distribution. The further one wants to reach into the high probability tail of the sequence distribution, the less likely it is that such high probability sequences will be produced by a random walk, or indeed iterative random walk (the creation of many samples by repeated application of the procedure).

Iterative random walk (Herremans, Sörensen, & Conklin, 2014; Whorley & Conklin, 2015), Gibbs sampling (Conklin, 2003) and the Markov constraints method (Pachet & Roy, 2011) are true sampling algorithms, in the sense that they implicitly or explicitly attempt to construct the true, but unknown, sequence distribution (in other words, the set of all possible sequences), rated in accordance with an objective function such as cross-entropy. Sampling algorithms can be used for optimization by retrieving low cross-entropy solutions from the generated sequence distribution: algorithms can retain an arbitrary number of the lowest cross-entropy solutions found so far, while discarding the rest (Herremans et al., 2014).

Techniques exist which can be expected to generate higher probability pieces than iterative random walk. The Viterbi algorithm (Viterbi, 1967) is used to find the single most probable sequence of hidden states from a hidden Markov model (an optimization task). It does this in such a way that it is not necessary to visit the full sequence distribution. It can, for example, find the most probable sequence of chord symbols (hidden states) given observed events in the form of a melodic sequence (Allan & Williams, 2005). Unfortunately, the time complexity of Viterbi decoding is quadratic in the length of the hidden state context, and therefore the algorithm is not suitable for use in conjunction with more complex statistical models.

For many problems it is not possible to establish what the single most probable sequence is, and so optimization algorithms must be used. These find the best solution that they can (or a few good solutions) according to an objective function. Practically, this is normally good enough. Such techniques include the Metropolis and Metropolis–Hastings algorithms, which are able to use an existing piece of music as a starting point (Conklin, 2003; Pearce & Wiggins, 2007). Both of these methods involve iteratively choosing a constituent event at random, substituting it with a different random event and re-evaluating the probability of the revised sequence containing the new event. Whereas Gibbs sampling (which follows a similar process) is an unbiased sampling method, both Metropolis methods are guided towards higher probability solutions: a slightly altered piece definitely proceeds to the next iteration if it has a higher overall probability, according to the model, than the piece as it existed in the previous iteration. Taking a lower probability piece forward is not completely ruled out; but the probability of rejection increases as sampling continues. Other optimization algorithms include stochastic hill climbing, simulated annealing (Kirkpatrick, Gelatt Jr., & Vecchi, 1983), variable neighbourhood search (Herremans et al., 2014, in the domain of music) and genetic (evolutionary) algorithms. All of these techniques are drawn to local minima; therefore they only search a small portion of the overall sequence distribution unless deliberate steps are taken to periodically break free of the current minimum.

Several different methods for melody generation using sampling and optimization from statistical models have been proposed. For example, Lo and Lucas (2006) describe an automatic melody composition method based on an *evolutionary algorithm* (EA), otherwise known as a *genetic algorithm*. This is used in conjunction with a *fitness function* (for evaluating the evolving music) implemented as an n-gram model trained on examples of existing music. Specifically, the EA utilized here is a *random mutation hill-climber*. Two ways of representing melodies are tried: MIDI pitch values and musical intervals in semitones. In both cases, there is an additional value representing a rest. Note and rest durations are ignored during n-gram model training; therefore generated notes and rests are of equal duration. Starting with a randomly generated melody, the EA applies an operator to randomly selected subsequences of one to ten notes such that the melody evolves. The performance of ten different operators was compared using pitch representation models trained on a corpus of Mozart melodies, in an experiment carried out over 100,000 generations. Limited comparison with the more abstract interval bigram model indicates that the latter may perform better.

Pachet and Roy (2011) propose a method which enables a user to steer the generation of Markovian sequences by introducing arbitrary constraints. The method relies on converting Markovian transitions into *Elementary Markov Constraints* (EMCs), which are 3-tuples containing n-gram context, continuation (prediction) and transition probability. For melody generation, a user might specify the note to be used in particular sequence positions, or draw a curve representing pitch contour. The generated melody is then a solution to a constraint satisfaction problem in which, for example, the melody is constrained by given notes, but is otherwise optimized by minimizing a cost function. One such function is the logarithm of the overall probability of the melody, which is calculated from the EMC probabilities (derived from a suitable corpus). In a melody generation study, backtracking occurred nearly 3000 times while optimizing a 22 note melody, compared with well under 1000 times for 14 notes. As the principle application of this work is a real time interactive music system, in practice music is generated in successive short chunks.

In a study of melody harmonization based on multiple viewpoint systems, Whorley et al. (2013) note that random walk alone generates unsatisfactory musical results. This is because the sampling of a low probability chord in the sequence may mean that it is inappropriate given its context. Such a chord (now as part of a context) can additionally affect the subsequent selection of chords, thus having a detrimental effect on the stylistic coherence of the generated harmonization. The solution is not simply to enforce very high probability events at every position, however, because this will lead to a dramatic loss of diversity of generated pieces, and furthermore does not even guarantee that the highest probability sequences will be produced (this is discussed in more detail in Section 3.6). If we view a good statistical model as providing a good fit to a corpus, it is reasonable to want to sample high probability sequences, which we hypothesize are closer in style to the pieces in the corpus. In this paper, we use melody harmonization to evaluate a new sampling method, based on iterative random walk (Whorley & Conklin, 2015), which is designed to explore the high probability tail of the solution distribution in a time efficient way. An important aspect of this work is the use of an objective method for evaluating the quality of the harmonizations, based on that used by Suzuki and Kitahara (2014). The generated samples are computationally checked for violations of some general rules of harmony. A sample with fewer rule violations than another is deemed to be of higher quality.

In this introduction we have discussed the difference between sampling and optimization algorithms, and reviewed a few such methods from the literature. In Section 2 we focus on some approaches to melody harmonization (or similar) which use statistical models. In Section 3 we discuss the corpus and briefly outline techniques which are particularly relevant to this work, especially *multiple viewpoint systems* and *probability thresholds*. Results and example harmonizations are presented in Section 4, followed by a general discussion and conclusions in Section 5.

2. Statistical methods for harmonization

This section reviews several existing statistical approaches to harmonization, a task highly amenable to statistical modelling and machine learning. A common theme of all methods reviewed is that one observed sequence (i.e. a melody or cantus firmus) is provided, and there is a vast underlying space of hidden sequences (the possible harmonizations of the melody, or counterpoint to the cantus firmus). The various methods differ mainly in the statistical relation between observed and hidden sequences, and how the latter are inferred from the model. A comprehensive survey is presented by Fernandez and Vico (2013, Section 3.3).

2.1 Markov models for counterpoint generation

The task of generating a single counterpoint line to a given *cantus firmus* is very similar to that of melody harmonization. Two approaches which use statistical models for the task are those of Farbood and Schoner (2001) and Herremans et al. (2014).

Farbood and Schoner (2001) demonstrate that Markov models can generate first species counterpoint which obeys rules set out in musical treatises. In the first instance, a set of rules is converted by hand into a corresponding set of probability tables; for example, the probability of a sequential interval conditioned on the previous one, and the probability of a vertical interval (without condition). A probability of zero is assigned to transitions which blatantly break a rule. Overall first-order transition probabilities are calculated by multiplying relevant values from each of nine probability tables (assuming conditional independence). Although some rules would be better captured by a second-order model, a one step look-ahead from the generation point (and a certain amount of optimization) means that the simpler first-order model is sufficient. In generating a countermelody to a cantus firmus, the user has the option of specifying the climax point (i.e. the position of the highest note). The Viterbi algorithm is used to find the solution with the greatest overall probability. Musically satisfactory results are obtained, comparable with examples composed by expert musicians. A corpus of counterpoint generated in this way, along with some human-composed examples, is then used to derive a new set of transition probability tables, for purposes of comparison.

Herremans et al. (2014) take a different approach, which does not treat sequential and vertical attributes as conditionally independent. The method is able to jointly generate the cantus firmus and one counterpoint line, and can be restricted to generate a counterpoint line compatible with a given cantus (resulting in a two-part texture in each case). They use a first-order Markov model constructed from a corpus of first species counterpoint to compare *variable neighbourhood search* (VNS) with iterative random walk and Gibbs sampling. This form of counterpoint comprises a sequence of *dyads* (two simultaneously sounding notes). An abstract representation is derived from the musical surface in the form of a linked *vertical viewpoint* (Conklin, 2002) which encodes dyad progressions as a combination of two melodic intervals (one for each part) and one harmonic interval within the second of the dyads, on a pitch class basis (e.g. an interval of 15 semitones is treated as equivalent to three). This representation enables a dyad-to-dyad transition matrix to be created from the corpus such that the data sparseness problem is ameliorated. Counterpoint generated by sampling from this statistical model is evaluated by calculating its cross-entropy, bearing in mind that low cross-entropy tends to imply higher quality. VNS begins with a random sequence of dyads on which it makes a series of *moves* (small predefined changes), with the goal of minimizing cross-entropy. Three such moves are defined. The set of sequences reachable by the application of a move is called a *neighbourhood*, from which the lowest cross-entropy sequence is chosen as the basis for the next move. On reaching a local optimum, a *perturbation* routine is invoked which randomly changes a predetermined percentage of notes, after which iteration continues. The algorithm terminates when the transition matrix has been accessed a prescribed number of times. In an experiment, the cantus firmus was specified and could not be altered by VNS; therefore only the set of dyads constrained by the cantus note could be considered at any point in the sequence. VNS reached the Viterbi solution 51 times compared with zero times for the other sampling methods.

2.2 Melody harmonization

Having discussed the addition of a countermelody to a cantus firmus by statistical means, we now move on to a review of some statistical approaches to melody harmonization.

2.2.1 Markov models

Chuan and Chew (2011) have developed the *automatic style-specific accompaniment* system, which employs statistical modelling and music theoretic knowledge in tandem. Specifically, they aim to harmonize melodies in a specified musical style, learned from a suitable corpus. A training corpus comprises songs from two albums by a single artist, in the form of melodies and associated basic triad chord labels. Chord progressions are represented in terms of single or compound *neo-Riemannian operations* to overcome the problem of data sparseness. Melody notes are converted to key-independent pitch classes before being represented by subsets of 73 attributes (designed to place notes in their melodic context). Decision trees induce statistics concerning whether or not a melody note should be considered part of a chord. Candidate chord labels can then be put forward based on the selected *chord tones*. *Melody fragments* are subsequences of notes harmonized by a single chord label. The melody is divided into sections beginning and ending with melody fragments called *checkpoints*, in which chord labels of particularly high likelihood are inferred. Each section is harmonized separately in the first instance, using Markov models constructed from the corpus. Chord transition probabilities depend, in part, on the position of chord labels in a phrase.

Raczyński, Fukayama and Vincent (2013) describe an automatic harmonization method which combines, by linear or log-linear interpolation, several statistical sub-models in which harmonic function symbols are conditioned on different musical attributes. Specifically, the task addressed is the addition of harmonic function symbols, comprising the note name of the root and the chord type, to a melody. Inversions are not taken into account. To overcome data sparseness problems, the models are smoothed by combining them with the prior (unconditional) and uniform chord distributions. There are currently first-order sub-models for chord transitions (chord given previous chord), chord given the prevailing key (or *tonality*, to take account of modulation) and chord in relation to melody. The sub-models are constructed from a training corpus, following which interpolation and smoothing coefficients are optimized on a smaller validation corpus. Each position in a musical sequence is a regular time frame (various time frames are investigated). Melody is encoded as the set of pitch classes appearing within a time frame. Multiple tonalities and harmonic function symbols within a time frame are reduced to the symbol having the longest duration. The results show that log-linear interpolation produces lower cross-entropies (and therefore better models) than linear interpolation.

2.2.2 Hidden Markov models

Allan and Williams (2005) present a four-part harmonization system comprising first-order *Hidden Markov Models* (HMMs) constructed from separate major and minor key corpora of J.S. Bach chorale harmonizations. The system adds alto, tenor and bass parts to a given soprano part. The musical data is split into chords of constant duration, equal to that of the beats in the bar (measure). In the HMM for harmonization, chords are considered to be a sequence of hidden states from which soprano notes are emitted to form an observed melody. There are learned transition probabilities between hidden states, and learned probabilities for the emission of visible notes (dependent upon the hidden state). Chords are represented by a list of intervals from the soprano in semitones, in conjunction with a harmonic function symbol (such as T for tonic). There is also a means of distinguishing between repeated notes or chords and those which carry over from beat to beat. The chord sequence which accompanies a melody with the highest likelihood is found by the Viterbi algorithm; but alternative chord sequences can also be sampled. A second HMM is used to ornament the basic harmonization with, for example, passing notes. Modelling this elaboration as a separate subtask considerably reduces data sparseness in the harmonization HMM. Hidden states comprise alto, tenor and bass movement in semitones at intervals of a quarter of a beat, relative to the pitch at the start of the beat. Between them the harmonization and ornamentation HMMs, faced with unseen melodies, produced examples of Viterbi harmonizations that were stylistically convincing.

2.2.3 Hidden semi-Markov models

Groves (2013) examines the potential of the *hidden semi-Markov model* (HSMM, Yu, 2010) for carrying out automatic harmonization in a purely symbolic manner. The aim is to add chords to a melody such that the resulting music is in the style of a 200 song Rock ‘n’ Roll training corpus. Previous approaches to such tasks using hidden Markov models have only made use of melody pitch and harmonic function (chord) symbol (e.g. I and V for tonic and dominant chords respectively) information, whereas HSMMs additionally allow the incorporation of chord duration data. A chord (hidden state) can harmonize from one to sixteen (an arbitrary maximum) observed melody notes, its duration being unusually defined as the number of these notes: note lengths, such as minim (half note), are not taken into account. In the *Explicit Duration* HSMM, the probability of a chord and its duration is dependent upon the previous chord (excluding its duration); and a chord may not transition to the same chord. The *Variable Time* HSMM has two chord transition possibilities, both conditioned on the previous chord and its duration: if the transition is to a different chord, the new chord’s duration is set to 1; and if a chord transitions to itself, its duration is incremented (that is, there is no new chord). The model used to perform the harmonization task is essentially a Variable Time HSMM, but with transitions independent of the previous duration (as in the Explicit Duration HSMM), so as to limit the size of the state set for time complexity reasons.

2.2.4 Bayesian networks

Suzuki and Kitahara (2014) note that most previous research on four-part harmonization has explicitly modelled harmonic function symbols. They have developed Bayesian network models with and without chord nodes to explore the extent to which four-part harmonization can be learned from an unlabelled music corpus. Durations are ignored and there is assumed to be one chord per melody note, here achieved by expanding the harmony in the training corpus. Both networks have nodes for sequence positions $i-1$, i and $i+1$. Given all musical information at $i-1$ and soprano notes at i and $i+1$, the remaining musical information at i and $i+1$ is inferred concurrently such that the total likelihood is maximized. This is intended to elicit a smooth progression. The inferred nodes at $i+1$ are reset to *unknown* before the network moves one position to the right. The *non-chord model* has nodes for soprano, alto, tenor and bass. All lower part nodes are dependent upon the vertically aligned soprano node and the previous node in the same part. The inner part nodes are also dependent upon the vertically aligned bass node. The two types of *chord model* have additional nodes representing chords, which are conditioned on the previous chord. This time, all part nodes are dependent upon the vertically aligned chord node as well as the previous node in the same part. There are no dependencies between vertically aligned note nodes. Chord model 1 makes use of harmonic function symbols indicating the root (e.g. E or G♯) and whether the chord is major or minor; whereas chord model 2 employs symbols which additionally include chord extensions and bass notes.

The training corpus comprised 254 hymn tune harmonizations in a major key, transposed to C major. Thirty-two C major melodies (not hymn tunes) were harmonized once each by the chord and non-chord models, and the results were evaluated using six criteria: dissonant chords, non-diatonic notes, tonic chord endings, chords containing the same pitch class in three or more parts, successive large bass leaps and the mean number of pitch classes appearing in each lower part. Arguably, the non-chord model is best, as its evaluation score is closest to that of the training corpus according to the first, second, fourth and sixth criteria; and it performs reasonably well according to the other two.

2.2.5 Multiple viewpoint systems

Multiple viewpoint systems (Conklin & Witten, 1995) are statistical models of multiple basic and more abstract musical attributes (or viewpoints). Whorley et al. (2013) describe the development of the multiple viewpoint framework for harmony, which is used as the basis for three increasingly complex models. The simplest model (version 1) is very similar to earlier melodic models, with complete chords (tuples of four notes) replacing single melody notes. Obviously the choice of chord is constrained by the soprano note to be harmonized. A more complex model (version 2) allows prediction or generation to be carried out part by part; for example, the entire bass part can be added first, followed by the inner parts (together or separately). So far, it has been assumed that the same viewpoint is used in each part. The most complex model (version 3), however, allows different viewpoints in different parts. Whorley et al. (2013) present an algorithm for the reliable construction of domains (alphabets) for these complex *inter-layer linked* viewpoints. A detailed comparison of the performance of these models is given in Whorley (2013). In summary, version 2 is better than version 1 where fair comparisons are possible (i.e. overall, version 2 predicts existing music with lower cross-entropy), and version 3 is best of all in spite of using a smaller pool of viewpoints for time complexity reasons.

2.3 Discussion

In this section we have presented several different statistical approaches to melody harmonization (or similar), unified by the idea of generating a hidden (usually harmonic) sequence from an observed (usually melodic) sequence. In terms of the limitations of the methods, note durations are completely ignored in some of the models (Groves, 2013; Lo & Lucas, 2006; Suzuki & Kitahara, 2014), while being taken into account to a varying degree in others. Since rhythm is an important component of music, it seems reasonable to suppose that note duration should be modelled in addition to pitch. The multiple viewpoint framework addresses several of these limitations, allowing the modelling of both note duration and pitch, including interaction between them.

In many cases, harmony is modelled using only harmonic function symbols, or similar representation (Chuan & Chew, 2011; Groves, 2011; Raczyński et al., 2013), with differing powers of discrimination. Although this may be acceptable for certain types of music, it is not sufficient for four-part harmonization. The multiple viewpoint framework, on the other hand, is perfectly capable of taking account of soprano, alto, tenor and bass notes, as are the models of Allan and Williams (2005) and Suzuki and Kitahara (2014). In fact, our implementation of the multiple viewpoint framework does not use harmonic function symbols at all, as is the case for the non-chord model of Suzuki and Kitahara (2014) which arguably outperforms their chord model.

Other than the approach of Suzuki and Kitahara (2014), the harmonization methods we describe above are based on first-order chord dependencies; that is, chord prediction probabilities are conditioned on one immediately preceding chord. This is not an unreasonable assumption, considering that treatises on harmony are mostly concerned with progressions from one chord to another taken in isolation. Experienced musicians, however, have an intuitive understanding of the way that longer sequences of chords work together; therefore it is also not unreasonable to consider models higher than first-order. The multiple viewpoint framework is able to do this in such a way that data sparseness is not as problematic as might first be thought. In addition, the framework can handle non-adjacent (and therefore longer term) dependencies.

Some of the models described above use an abstraction of pitch, such as melodic and harmonic intervals (Allan & Williams, 2005; Herremans et al., 2014; Lo & Lucas, 2006), pitch classes (Herremans et al., 2014; Raczyński et al., 2013), key-independent pitch classes (Chuan & Chew, 2011), and (effectively) key-independent pitches by virtue of transposition to the same key (Suzuki & Kitahara, 2014). This brings us to the greatest advantages of multiple viewpoint systems: the fact that many different abstractions of basic musical attributes are available for modelling purposes, that multiple models can be combined in a way that is more sophisticated than methods outlined by Raczyński et al. (2013), and that the set of models to be combined can be automatically optimized to extract the maximum information from the corpus. Details of these and other related matters are given in the next section.

3. Methods

In this section we first examine the corpus used to train the multiple viewpoint statistical model of harmony, in particular discussing the modelling challenge presented by its partially polyphonic nature. We then present a brief outline of multiple viewpoint systems and their application to the modelling of melody and harmony. An introduction to cross-entropy as a measure of sequence likelihood comes next, followed by a description of the viewpoint selection procedure. Finally, the probability threshold sampling method is explained.

3.1 Corpus

The corpus used for machine learning purposes comprises 100 hymn tune harmonizations which form part of a collection edited by Vaughan Williams (1933); see Appendix C for a complete list. These pieces are all in a major key and without rests. The latter restriction is due to a limitation of the current implementation of our harmonization system; but since the vast majority of hymn tune harmonizations contain no rests, this is not a serious problem. The pieces generally follow the harmonization rules of the common practice era; but their texture varies from the absolute homophony of *Tallis' Canon* to the extreme floridity of *Caton* (or *Rockingham*). The first stage in the production of the corpus was to create MIDI files using commercially available music sequencing software. A specially written Java program then converted the MIDI files into text files containing a list of tuples, which describe each note of each part in terms of a set of musical attributes.

Since most of the music is not completely homophonic (e.g. it contains passing notes), the music is automatically expanded (i.e. notes are divided) such that a four-part chord exists everywhere a note is newly sounded. In order to preserve information concerning the (partially) polyphonic nature of the music, attribute `Cont` was introduced, which indicates whether a note is freshly sounded or is a continuation of the preceding note in the same part. `Duration` is a measure of note length where, for example, a crotchet (quarter note) has the value 24; while `Pitch` is encoded as MIDI pitch values. Global attributes `KeySig`, `Mode`, `BarLength` and `Pulses` are made available to the Java program as a handcrafted auxiliary text file. `KeySig` is plus or minus the number of accidentals in the key signature, where $+$ and $-$ represent sharp and flat respectively. `Mode` takes integer values representing major, minor or archaic modal forms such as Dorian. `BarLength` is the length of a bar (measure) in the same units as `Duration`, while `Pulses` is the number of beats in a bar; for example, a bar in $\frac{6}{8}$ time has a `BarLength` value of 72 and a `Pulses` value of 2, while a $\frac{3}{4}$ bar has the same `BarLength` value but a `Pulses` value of 3. It should be noted that the `Pulses` value is fixed for the duration of the piece; therefore the modelling of metre does not take account of the deliberate alternation of bars with two and three pulses, as sometimes occurs in music. `Onset` is a measure of the temporal position of a note in a piece, in the same units as `Duration`, starting at the beginning of the bar containing the first chord; that is, a non-zero value for the first note indicates an anacrusis. `Piece` trivially indicates the beginning and ending of a piece, while the `Phrase` attribute indicates phrase boundaries.

3.2 Modelling of melody

The multiple viewpoint systems used in this work contain both sequential and vertical components. The sequential aspects are explained first, with reference to melodic modelling. We assume that the model is capable of generating attributes `Duration` and `Pitch` for each note in turn, such that a completely defined melody is produced in one pass. Attributes are also known as basic viewpoint types, or more simply as basic viewpoints, from which other viewpoints are derived; for example, `Interval` is derived from basic viewpoint `Pitch`. Attributes other than those to be generated, such as `BarLength` and `KeySig`, are assumed to be given. Threaded viewpoints, such as `ScaleDegree` $\ominus$ `LastInPhrase`, are able to model non-adjacent sequence positions (i.e. longer distance dependencies). Linked viewpoints, such as `Duration` $\otimes$ `Interval`, are conjunctions of *primitive* viewpoints (i.e. single basic or derived, including threaded, viewpoints). A multiple viewpoint system is a set comprising more than one viewpoint (usually many more).

The overall model for melody has a hierarchical structure, with models at different levels of abstraction throughout. The overall model comprises separate subtasks for the generation of `Duration` and `Pitch` attributes (for each note in turn, `Duration` is generated before `Pitch`). Each subtask may consist of a *long-term model* (LTM), which models the entire corpus, and a *short-term model* (STM), which models a single piece of music as it is predicted or generated (thereby capturing intra-opus repetition and aspects of a piece's uniqueness). Each of these models is a multiple viewpoint system comprising *viewpoint models*.

At the base of the hierarchy are simple *n-gram models* (e.g. Jurafsky & Martin, 2000), in which the *prediction* probability depends on a short historical *context* of length $n-1$ (the *order* of the model). For example, the `Interval` trigram -2 2 0 can be used to make a second-order prediction, with prediction 0 following context -2 2. The higher level viewpoint models construct complete prediction probability distributions from the counts of n-gram models from some maximum order $\hbar$ down to the uniform distribution, using *Prediction by Partial Match* (or PPM, Cleary & Witten, 1984). In very general terms, predictions following the $\hbar$th-order context (as seen in the corpus) are assigned relatively high probabilities; but not all predictions will have been seen at this point, therefore back-off to the $(\hbar-1)$th-order context occurs, and so on, with probabilities becoming progressively lower. Completing the probability distribution with reference to the uniform distribution of all possible predictions ensures that the models are *non-exclusive*; that is, that no event has zero probability. The various viewpoint distributions are then converted to either `Duration` or `Pitch` distributions, depending on the subtask.

Ascending the hierarchy a little further, viewpoint model distributions are amalgamated by means of a weighted geometric combination technique (Pearce, Conklin, & Wiggins, 2005) which makes use of a *bias* favouring distributions of low entropy (Conklin & Witten, 1995). It should be noted that not all viewpoints are defined at all points in the musical sequence; for example, in a melody of four phrases, `ScaleDegree` $\ominus$ `LastInPhrase` is only defined in four places. At any point, then, only defined viewpoints are involved in the amalgamation. The same technique is used to combine the LTM and STM distributions, although the optimal *LTM-STM bias* is likely to be different from the viewpoint combination bias.

3.3 Modelling of four-part harmony

Whorley et al. (2013) present a melody harmonization method which extends the melody generation method outlined above. Here, PPM is used in conjunction with escape method C (see Witten & Bell, 1989, for a review of escape methods). Tuples of soprano, alto, tenor and bass viewpoint values are used for modelling purposes (Conklin, 2002), producing, for example, `Interval` tuple trigrams such as $\langle -2, 1, -1, 5\rangle$ $\langle 4, 0, 3, 0\rangle$ $\langle 3, 0, 0, 4\rangle$. This enables alto, tenor and bass lines to be added in a single pass. `Pitch` tuples represent chords in a completely unmodified way; that is, with the notes of a chord distributed precisely as they appear in the music. Derived viewpoints model abstractions of this representation; but since they contribute to the creation of prediction probability distributions over `Pitch` tuples, the representation can be seen to handle both vertical and horizontal (voice leading) aspects of harmony explicitly. Since the melody is given, and cannot be changed, prediction probability distributions are constrained by the attributes of the current soprano note; for example, if this note is G4, all candidate chords (the current `Pitch` tuple domain) must have a G4 in the soprano. The complete `Pitch` tuple domain (or alphabet) comprises all chords seen in the corpus plus all transpositions of those chords which lie entirely within the part ranges seen in the corpus. This is a compromise between the extreme time complexity of using all possible pitch combinations and the over-restrictiveness of using only seen combinations. For more details on how viewpoint domains are constrained and constructed, see Whorley et al. (2013). In this model of harmony, `Duration`, `Cont` (see Section 3.1) and `Pitch` attributes are generated for each chord in the sequence, in that order. In our current research, each of these subtasks is an LTM only for time complexity reasons. Similarly, although more complex (and better performing) models have been implemented (Whorley, 2013), for example one which generates the bass line first followed by the inner parts, we employ the simplest in the work described in this article.

3.4 Cross-entropy

It is possible for different harmonizations of the same melody to contain different numbers of chords (with commensurately different numbers of chord probabilities), which affects the overall probability of the piece. This can occur, for example, when one solution contains more passing notes than another in the lower three parts, since each passing note is part of a separate chord in the expanded harmony required for modelling purposes. Similarly, some melodies are longer than others in the first place. To ensure fair comparisons between solutions, we use the information theoretic measure *cross-entropy* instead of the sequence probability. Note that low cross-entropy indicates a high probability sequence. An approximation to cross-entropy for a sequence $e_1, \ldots, e_n$ is given by

$$H = -\frac{1}{n}\sum_{i=1}^{n}\log_2 P(e_i|c), \tag{1}$$

where n is the number of sequential events and $P(e_i|c)$ is the probability of event e_i, given its context c, according to the statistical model (see e.g. Jurafsky & Martin, 2000, for more details).

3.5 Viewpoint selection

The precise combination of viewpoints in a multiple viewpoint system is determined objectively, by minimizing the cross-entropy in a ten-fold cross-validation of the corpus. The automatic procedure extends that of Pearce (2005). Starting with systems {`Duration`}, {`Cont`}, and {`Pitch`}, single viewpoint additions (such as `Interval` to the latter system) and additions of links with viewpoints already in the system (such as `Pitch` $\otimes$ `Interval`) are tried. The addition which reduces cross-entropy the most is accepted, and then a similar procedure is carried out for all possible deletions. Addition and deletion stages continue until the stage to stage reduction in cross-entropy falls below a prescribed minimum. Biases are automatically optimized after viewpoint selection. Optimal $\hbar$ is determined by carrying out viewpoint selection runs at different $\hbar$ values. The multiple viewpoint systems used in this research (see Appendix A) were selected in relation to a sub-corpus of 50 hymn tune harmonizations (for time complexity reasons); but they are used in conjunction with the full 100 piece corpus when generating novel harmonizations. See Whorley (2013) for more details on viewpoint selection with respect to the modelling of harmony, including algorithms.

3.6 Probability thresholds

The *probability threshold* sampling method (Whorley & Conklin, 2015; Whorley et al., 2013) modifies standard random walk by calculating a minimum allowable probability equal to a specified fraction of the highest probability in the prevailing distribution. Predictions with a probability lower than this threshold are effectively disregarded during sampling. Consider the distribution X constructed for an event in a sequence. A new distribution of events Y is defined by removing all events from X that fall below a specified threshold $t \in [0, 1]$ of the highest probability event in X:

$$P(Y = x) = \begin{cases} 0 & P(X = x)/m < t \\ P(X = x)/z & \text{otherwise} \end{cases} \quad (2)$$

where $m = max_{x \in X} P(X = x)$, and z is a normalizing constant.

As an example, for a hypothetical random variable X with the probability distribution $P(X = a) = 0.5$, $P(X = b) = 0.3$, $P(X = c) = 0.15$, $P(X = d) = 0.05$, let us consider four different thresholds: $t = 0$, $t = 0.25$, $t = 0.5$ and $t = 1$. The original and resulting distributions are shown in Table 1. Note that although the new thresholded distribution Y is used during sampling, cross-entropies are calculated from the original probabilities in X for fair comparison of solutions generated with different t values. At $t = 0$, the minimum allowable probability is zero, hence there is no change to the original distribution and sampling proceeds as standard random walk. At the other extreme, $t = 1$, the minimum probability is equal to 0.5, the highest probability in the distribution; since all other predictions have a lower probability, only prediction a remains in Y, with a revised probability of 1. Assuming that there is not more than one prediction with the highest probability in any distribution (a safe assumption for a long-term multiple viewpoint model), then this t value produces only a single solution. In spite of using the highest probability for each prediction, this deterministic solution does not necessarily have the lowest possible cross-entropy; sometimes a lower probability prediction can lead to the possibility of higher probability predictions later in the sequence. We shall demonstrate this with an example.

Table 1. Table showing the original probability distribution X over four predictions a, b, c and d, and modified distribution Y at four different probability threshold values.

	X	Y ($t = 0.00$)	Y ($t = 0.25$)	Y ($t = 0.50$)	Y ($t = 1.00$)
a	0.50	0.50	0.526	0.625	1
b	0.30	0.30	0.316	0.375	0
c	0.15	0.15	0.158	0	0
d	0.05	0.05	0	0	0

Table 2. Transition matrix for a first-order Markov model, showing conditional probabilities of predictions e, f, g and h.

	prediction			
context	e	f	g	h
e	0.3	0.4	0.2	0.1
f	0.4	0.3	0.1	0.2
g	0.1	0.1	0.1	0.7
h	0.1	0.2	0.6	0.1

Table 2 shows an artificially constructed transition matrix for a first-order Markov model. There are four probability distributions over predictions e, f, g and h: one for each context. Let us assume that we are creating short sequences which must begin with an e (this letter, without context, having a probability of 1). We then add three more letters by sampling from the relevant distributions to complete a sequence. The first sequence will be constructed by always generating the letter with the highest probability. To begin with, the context is e; therefore f is generated next, with a probability of 0.4. This is followed by another e, and a second f completes the sequence (both with a probability of 0.4). The total probability of the sequence is therefore 0.064. Let us now generate a second sequence, again starting with the prescribed e. This time, the second letter is g, with the comparatively small probability of 0.2. This paves the way for h and g, with probabilities of 0.7 and 0.6 respectively. The total probability of this sequence is 0.084, which is higher than that of the sequence constructed by adding the highest probability letters at each stage (recall that high sequence probability correlates with low cross-entropy). We shall see this demonstrated using real statistical models in Section 4.2.

4. Results

The results of two types of evaluation experiment are presented in this section: one relating to harmonic correctness, and the other to various measures of musical diversity.

4.1 Harmonic correctness and cross-entropy

Hymn tunes *Das walt' Gott Vater*, *Innocents*, *Das ist meine Freude*, *Bromsgrove* and *Monte Cassino* (hymn nos. 36, 37,

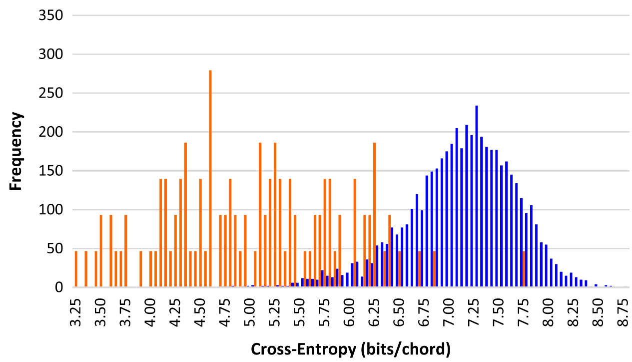

Fig. 1. Histogram showing frequency against cross-entropy (mid-points of 0.05 bits/chord bins). The right-hand distribution is for runs generating 1024 samples of harmony by random walk for each of five melodies not present in the corpus (Vaughan Williams, 1933, hymn nos. 36, 37, 97, 144 and 150). The left-hand distribution is for hymnal harmonizations of these and 105 other tunes, on a not-in-corpus basis, scaled to 5120 samples for direct comparison.

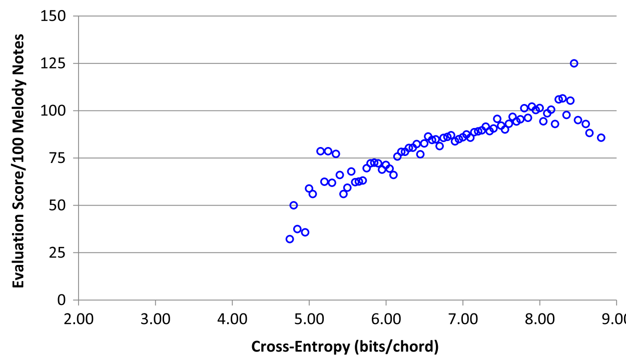

Fig. 2. Plot of evaluation score on a per 100 melody note basis against cross-entropy (mid-points of 0.05 bits/chord bins), for the 5120 samples of generated harmony described in the caption to Figure 1. Lower scores indicate fewer rule violations, and therefore better quality harmony.

97, 144 and 150 respectively, Vaughan Williams, 1933) were harmonized 1024 times each by iterative random walk using the multiple viewpoint method described in Section 3. We can see from Figure 1 that the cross-entropies of actual hymnal harmonizations are generally much lower than those generated by iterative random walk. Samples produced in this way typically contain many chord progressions, note lengths and so on which are atypical of the corpus to the extent that many rules of harmony (e.g. Piston, 1976) are broken. Since we would expect the hymnal harmonizations, as used in the corpus, to contain relatively few rule violations (empirically demonstrated in Section 4.2), we may surmise that the quality of harmony improves with decreasing cross-entropy in the generated samples. The fact that iterative random walk produced samples covering a huge range of cross-entropies means that we were well placed to test this hypothesis in an objective manner.

The test method decided upon was similar to that used by Suzuki and Kitahara (2014). The 5120 samples were computationally checked for violations of some of the more general rules of harmony (note that Evans, Fukayama, Goto, Munekata, and Ono, 2014, also use rules of harmony for evaluation purposes). Specifically, the samples were checked for part overlaps between adjacent chords, parallel fifths and octaves, and unusually large leaps across two or three notes in the same part (i.e. leaps of 11 semitones, or greater than an octave). These few clear and simple general rules were considered preferable to more specific or more complex rules from the point of view of implementation and data analysis. See Appendix B for a more detailed description of the rules and the applicable scoring system. A weighted mean rule violation count was calculated for each bin of 0.05 bits/chord, as shown in Figure 2. The weightings adjusted the counts to a per 100 melody note basis for fair comparison, since we would expect longer melodies to contain proportionately more rule violations, all else being equal. We can see that, as expected, there is a clear trend towards fewer rule violations, suggesting better quality harmony, at lower cross-entropy.

4.2 Cross-entropy reduction and minimization

The establishment of a correlation between quality of harmony (as indicated by the number of rule violations) and cross-entropy elicits the following question: if it were possible to generate samples of harmony at lower cross-entropies, would the trend towards fewer rule violations continue? We would certainly hope this to be the case, considering that Figure 2 still shows a fairly large number of violations at the low cross-entropy end of the sample distribution generated by iterative random walk. One way of achieving lower cross-entropy solutions is to simply generate a very much larger number of samples; but this takes an excessively long time, and relatively few low cross-entropy solutions emerge. Ideally, we would wish to find an optimization procedure which shifts focus to a generally lower cross-entropy search space, such that very low cross-entropy solutions can rapidly be found. We have developed a novel procedure, based on iterative random walk and probability thresholds, which does precisely this.

Whorley et al. (2013) showed that a modified random walk procedure, making use of probability thresholds, could generate individual lower cross-entropy solutions (see Section 3.6). Our improved procedure combines this technique with the iterative method employed in Section 4.1. By using a threshold t of 1, only the highest probability prediction in any distribution is generated, meaning that just one low cross-entropy solution is possible. One of our aims is to demonstrate that even lower cross-entropy solutions can be found by using lower values of t. This we shall do now.

As described in Section 4.1, five hymn tunes have already been harmonized 1024 times each using standard iterative random walk ($t = 0$). The same procedure was followed from $t = 0.05$ to $t = 1$, in increments of 0.05. The results are tabulated and plotted in Table 3 and Figure 3. Mean minimum is the mean of the cross-entropy minima of the five hymns. It

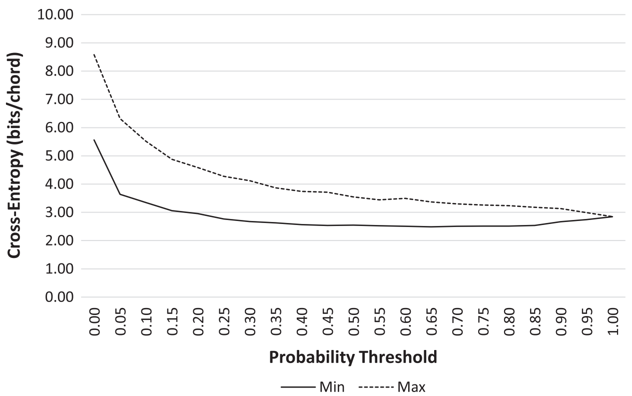

Fig. 3. Plot of mean minimum *(Min)* and mean maximum *(Max)* cross-entropy against probability threshold, for runs generating 1024 samples of harmony at each value of t for each of the tunes to hymns numbered 36, 37, 97, 144 and 150 (Vaughan Williams, 1933). Mean minimum is the mean of the minima of the five hymns.

turns out that all t in the range 0.25 to 0.95 produce a lower mean minimum than $t = 1$, which has a cross-entropy of 2.85 bits/chord. The lowest cross-entropy of all, 2.49 bits/chord, results from a t of 0.65.

So far, the same value of t has been used to generate the attributes `Duration`, `Cont` and `Pitch`; but it is possible to optimize t for each individual attribute such that even lower cross-entropies can be found. Starting with a t of 0.65 for all attributes, t for `Duration` was adjusted first. Values of 0.60 and 0.70 resulted in higher cross-entropies; therefore 0.65 was retained. `Cont` was dealt with next, and in this case a t of 0.70 reduced the cross-entropy to 2.46 bits/chord. Continuing in the same way, $t = 0.65$ was retained for `Pitch`; and checking `Duration` once more also produced no change.

Figure 4 illustrates the effect of using different values of t. The distribution at the high end of the cross-entropy range is the one produced by standard iterative random walk (as shown in Figure 1). A t value of 0.05 shifts the distribution a long way towards the lower end of the range; while thresholds optimized for each attribute produce a distribution containing very low cross-entropies. The rather irregular shape of the latter distribution is due to the way in which the distributions for individual melodies overlap with each other. We are, then, able to focus on solution spaces containing lower cross-entropy harmonizations; but is the harmony of better quality than before?

Figure 5 shows a plot of evaluation score against cross-entropy, extending Figure 2 by adding $t = 0.05$ and $t = 0.65/0.70/0.65$ to the plot for standard iterative random walk ($t = 0$). It is clear that the number of rule violations continues to decrease with cross-entropy all the way down to about 2.5 bits/chord. From there, the number of violations increases again down to 2.0 bits/chord, which is entirely due to anomalous results from *Innocents* (hymn no. 37, Vaughan Williams, 1933). The reason for this is not obvious, but we do not think that, in general, the trend would reverse in that

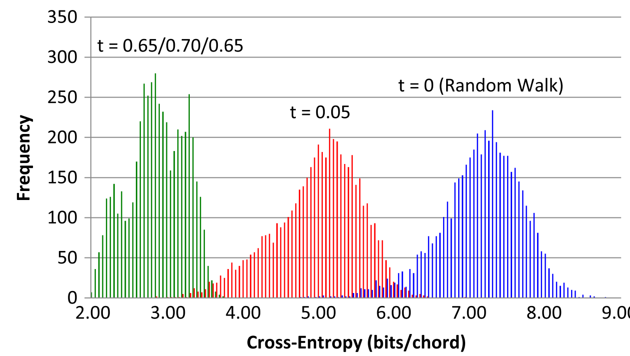

Fig. 4. Histogram (1024 samples per hymn tune per distribution divided into bins of 0.05 bits/chord) showing $t = 0.65/0.70/0.65$ (optimized for `Duration`/`Cont`/`Pitch` respectively), $t = 0.05$ and $t = 0$ (standard random walk) sample distributions. The five tunes (not present in the corpus) are to hymns numbered 36, 37, 97, 144 and 150 (Vaughan Williams, 1933).

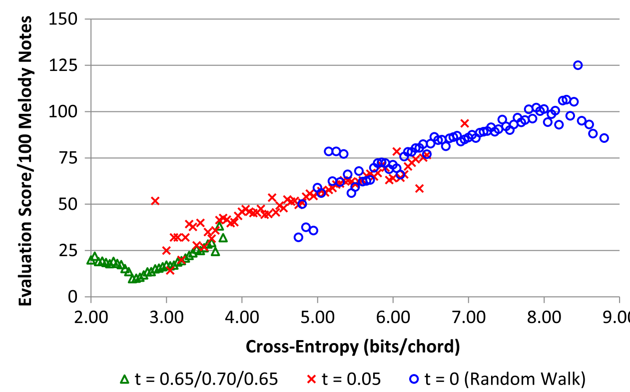

Fig. 5. Plot of evaluation score on a per 100 melody note basis against cross-entropy (mid-points of 0.05 bits/chord bins), for the 15,360 samples of generated harmony described in the caption to Figure 4.

manner; see Figure 6 for the same plot without *Innocents* data. The $t = 0.65/0.70/0.65$ sample distribution restricted to *Innocents* is of particularly low cross-entropy, and overlaps the least with other tunes' distributions (it does not overlap at all with that of *Bromsgrove*). Irrespective of this, there are far fewer rule violations in the samples produced by optimizing the thresholds, compared with $t = 0.05$ and $t = 0$. This low cross-entropy solution space is therefore greatly superior to that produced by standard iterative random walk.

The evaluation plots have made use of bins of 0.05 bits/chord, with a mean evaluation score calculated for each bin, so that trends are easily discernible. To give a better idea of the range of evaluation scores in the sample distributions, a scatter plot is presented in Figure 7 (please note the increased range of the y-axis). In the $t = 0$ case, at worst there are more than 200 rule violations per 100 melody notes, and at best about 20 per 100. For $t = 0.05$ we come down to about 150 at worst, while one sample has no violations at all. Optimized thresholds again produce the best results on this

Table 3. Table of probability threshold *(t)* against mean minimum *(Min)* and mean maximum *(Max)* cross-entropy, using a long-term model in runs generating 1024 samples of harmony at each value of *t* for each of the tunes to hymns numbered 36, 37, 97, 144 and 150 (Vaughan Williams, 1933). Mean minimum is the mean of the minima of the five hymns.

t	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00
Min	5.56	3.64	3.35	3.06	2.95	2.76	2.67	2.63	2.56	2.54	2.55	2.53	2.51	2.49	2.51	2.51	2.51	2.53	2.67	2.74	2.85
Max	8.58	6.32	5.51	4.88	4.58	4.28	4.12	3.87	3.74	3.71	3.55	3.44	3.49	3.37	3.30	3.26	3.23	3.18	3.14	2.99	2.85

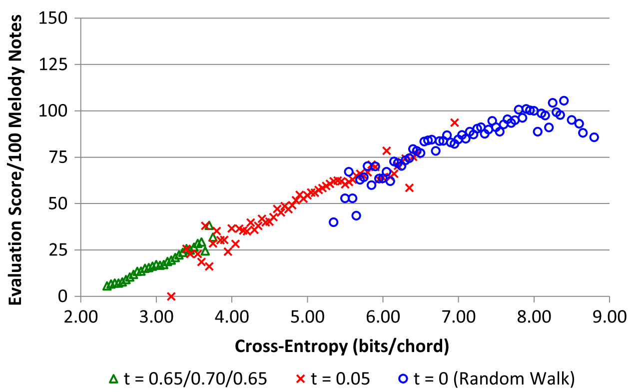

Fig. 6. Plot of evaluation score on a per 100 melody note basis against cross-entropy (mid-points of 0.05 bits/chord bins). As Figure 5, but with the data for *Innocents* (hymn no. 37, Vaughan Williams, 1933) removed (12,288 samples of generated harmony).

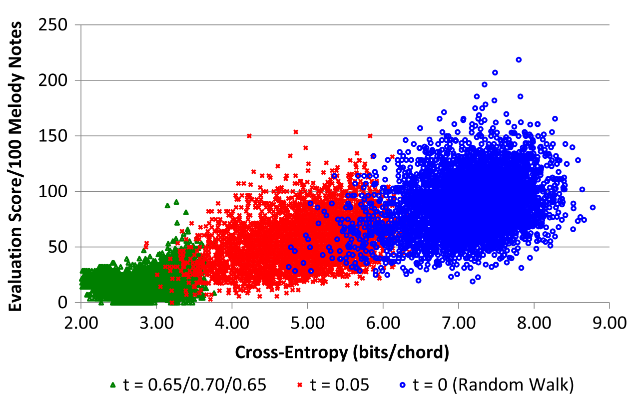

Fig. 7. Scatter plot of evaluation score on a per 100 melody note basis against cross-entropy, for the 15,360 samples of generated harmony described in the caption to Figure 4.

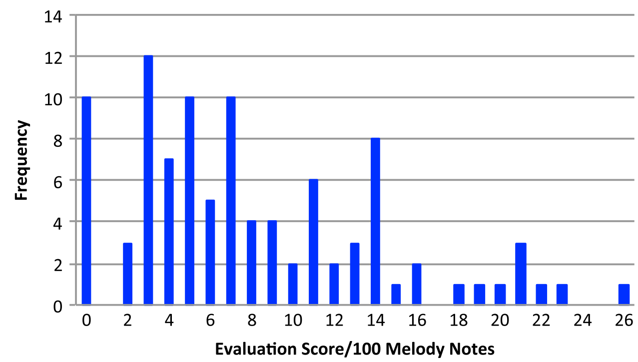

Fig. 8. Histogram of frequency against evaluation score on a per 100 melody note basis (rounded to the nearest whole number) for 98 of the 100 pieces in the corpus (from Vaughan Williams, 1933). The two excluded pieces contain unison passages, which are not expected to comply with the usual rules of harmony (i.e. all of the voices sing the melody line in these passages, with the tenor and bass an octave below the soprano and alto). The vast majority of violations are due to part overlaps.

basis, since a fair number of samples have zero violations and the maximum number is less than 100.

The 100 corpus pieces have been checked for rule violations, which were found to cover a range from zero to 52.94 violations per 100 melody notes, with a mean of 8.81 (ten pieces had no violations). This fits best with the range produced by optimized thresholds. Note that the two pieces with most violations (52.94 and 47.69) were the only ones to contain unison passages, which are not expected to comply with the usual rules of harmony (i.e. all of the voices sing the melody line in these passages, with the tenor and bass an octave below the soprano and alto); since the automated check did not contain all of the required exceptions in these circumstances, an excessive number of violations was recorded. If these pieces are ignored, the range becomes zero to 25.81 violations per 100 melody notes, with a mean of 7.97. A histogram of frequency against evaluation score on a per 100 melody note basis (rounded to the nearest whole number) is presented in Figure 8. The vast majority of violations are due to part overlaps, strongly indicating that this rule is far less rigid than the others. The rule most strictly adhered to is the one prohibiting large leaps between adjacent notes in the same part.

Something is not quite right, however. Recall from Section 4.1 that actual hymnal harmonizations, with few if any rule violations, had cross-entropies ranging from approximately 3.25 to 7.75 bits/chord. This range corresponds best with the $t = 0.05$ sample distribution; and it overlaps only a little with the distribution for optimized t, which has the fewest rule violations. The reason for this lies in the fact that the corpus is not really large enough. The hymnal harmonizations of the test melodies contain some chords and chord progressions which, although perfectly acceptable, have not been seen in the corpus. As a result of this, those chords are assigned extremely low probabilities by PPM, which pushes up the cross-entropy of the harmonizations. To obtain a rough

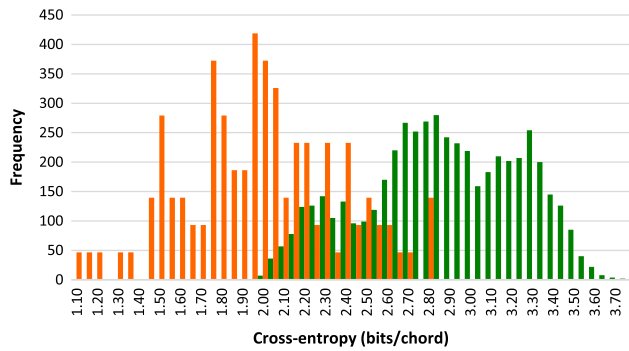

Fig. 9. Histogram showing frequency against cross-entropy (mid-points of 0.05 bits/chord bins). The right-hand distribution is for runs generating 1024 samples of harmony for each of five melodies not present in the corpus, using t = 0.65/0.70/0.65 (optimized for `Duration`/`Cont`/`Pitch` respectively). The five melodies are to hymns numbered 36, 37, 97, 144 and 150 (Vaughan Williams, 1933). The left-hand distribution is for hymnal harmonizations of these and 105 other tunes, scaled to 5120 samples for direct comparison; in this case, the harmonizations were present in the corpus.

indication of the values we might expect from a much larger corpus, we determined the cross-entropy of hymnal harmonizations while they were present in the corpus. Figure 9 shows that the hymnal harmonization distribution, produced in this way, now overlaps to some extent with the low cross-entropy side of the distribution for optimized t. The generally lower cross-entropies for the hymnal harmonizations are probably due to over-fitting of the in-corpus model.

Let us now consider whether optimization of t for individual attributes using multiple melodies is of any real benefit. The motivation for this was to avoid the additional time it would take to perform this optimization for individual melodies, realizing that there would be a compromise with respect to the cross-entropies achieved. In the initial stage of this multiple optimization, however, cross-entropies were recorded across the whole range of global t values. We compare minima achieved during this stage with those produced using optimized t in Table 4. Individual cases are higher or lower; but overall, from this admittedly limited data, the use of different global values of t seems to be as effective as a single set of thresholds optimized for each attribute, without the need to spend time on the final stages of optimization. There is also probably no need to go all the way to t = 0, judging by the global values shown in Table 4; perhaps going from 1.00 to 0.25 would give a reasonable margin for error.

If it is necessary or desirable to push towards the limits of discoverable minima, there is no option but to carry out t optimization for individual melodies. For *Das walt' Gott Vater* (hymn no. 36, Vaughan Williams, 1933), the optimized values of t are 0.45, 0.70 and 0.30 for attributes `Duration`, `Cont` and `Pitch` respectively. We are rewarded by a minimal cross-entropy of 2.04 bits/chord, compared with a previous best of 2.28 bits/chord (see Table 4). Figure 10 for *Das walt' Gott Vater* alone can be directly compared with Figure 5. The overall shape of the plots is similar, although in the case of hymn no. 36 there is less overlap between the distributions, and there is no pronounced reversal in the trend at very low cross-entropy.

Table 4. Table showing, for harmonizations of the five hymn tunes used for multiple optimization, minimum cross-entropy (bits/chord) for optimized t (`Duration`/`Cont`/`Pitch` respectively) and minimum cross-entropy at the relevant global value of t (i.e. a single value of t used for all attributes). Mean cross-entropies are also given.

Hymn	t = 0.65/0.70/0.65	Global t values
36	2.37	2.28 (t = 0.50)
37	2.01	2.00 (t = 0.75 to 0.90)
97	2.42	2.49 (t = 0.65 & 0.70)
144	2.95	2.91 (t = 0.65)
150	2.56	2.60 (t = 0.50)
Mean	2.46	2.46

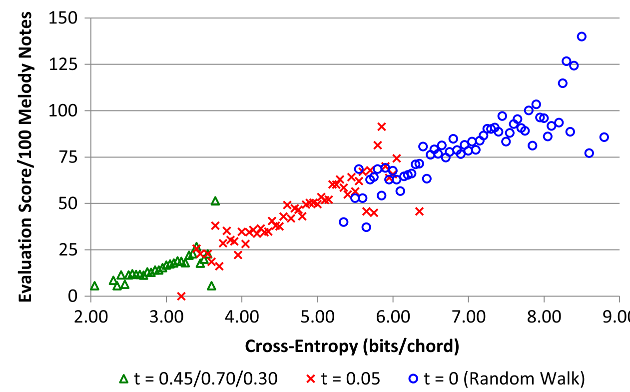

Fig. 10. Plot of evaluation score on a per 100 melody note basis against cross-entropy (mid-points of 0.05 bits/chord bins), for 3072 harmonizations of *Das walt' Gott Vater* (hymn no. 36, Vaughan Williams, 1933) divided equally between t = 0.45/0.70/0.30 (optimized for `Duration`/`Cont`/`Pitch` respectively), t = 0.05 and t = 0 (standard random walk). Derived from Whorley and Conklin (2015), Figure 4.

4.3 Detailed rule violation and other trends

Let us examine trends for individual rule violations (on a per 100 melody note basis), as illustrated in Figure 11. Part overlaps between adjacent chords (e.g. tenor note higher than the previous alto note) account for the majority of rule violations (see plot `a`), and so the trends are very similar to those of the overall evaluation score. The trend for parallel fifths (plot `b`) is not so linear, especially if the very low cross-entropy trend reversal is disregarded: the rate of increase of violations falls as cross-entropy increases. The parallel octaves plot has a much more pronounced very low cross-entropy trend reversal (plot `c`). If this is ignored, there is a high rate of increase of violations at low cross-entropy, which rapidly tails off to essentially a constant number of violations (on average) at high cross-entropy. One could speculate that the flattening of the parallel fifth and octave trends is due to the introduction of low probability chords not having these common intervals within them. There is a mostly linear relationship between cross-entropy and the number of large leaps between adjacent notes in a part (plot d); however, for optimized t the number of leaps is close to or equal to zero. The situation for large leaps across three notes is similar (plot e); but in this case, for optimized t, there appears to be an increase in the number of leaps with decreasing cross-entropy, with a sudden reversal close to 2.00 bits/chord. For 98 of the pieces in the corpus, the mean number of violations corresponding to plots a to e are 6.22, 0.36, 0.89, 0.03 and 0.47 respectively, best matching the plots for optimized t in each case.

We now move on to plots involving diversity measures rather than rule violations. The first set of plots, also on a per 100 melody note basis, is shown in Figure 12. The first observation to make is that the number of notes generally remains close to 400 (on average), which is what we would expect from four notes per chord (see plot f). For 100 melody notes, samples with more than 400 notes must by definition contain non-chordal notes (passing notes, appoggiaturas and auxiliary notes) in the bottom three parts; while samples with fewer than 400 notes have effectively had some soprano notes interpreted as non-chordal. All three t values produce a note range rising from about 400 to 440 with increasing cross-entropy. In each case, the number of notes remains approximately constant on reaching 440. There is no discernible overall trend across the whole cross-entropy range. Regarding the plot of number of events against cross-entropy (plot g), an event is said to occur when a note begins to sound or several notes begin to sound simultaneously. The number of events is likely to be related to the number of notes; indeed, the plots are somewhat similar in shape, although the number of events has a tendency to reduce after peaking. Since the number of notes remains approximately constant as the number of events tails off, there must be more notes starting per event.

The final diversity measure in Figure 12 (plot h) is the number of non-diatonic notes (Suzuki & Kitahara, 2014, criterion 2). For optimized t, this measure is generally close to zero: the increase in the number of non-diatonic notes as cross-entropy reduces to very low values is due entirely to the tune *Innocents*. A t of 0.05 sees a gentle rise in this measure with increasing cross-entropy; but $t = 0$ produces an explosion of non-diatonic notes. We can gather from this that a large number of chords containing non-diatonic notes are removed from consideration by the application of a t as low as 0.05.

The next set of diversity plots, not on a per 100 melody note basis, is shown in Figure 13. The number of distinct pitches, irrespective of the part in which they appear (plot i), is fairly constant at around 20 at low cross-entropy. The number rises to about 30 between 4 and 7 bits/chord, and then remains constant again. These numbers seem reasonable, considering that there are approximately 23 diatonic notes and 39 chromatic notes in a fairly conservative combined SATB range. At low cross-entropy, it is likely that most of the diatonic notes and a few non-diatonic notes are being used, whereas at high cross-entropy many more non-diatonic notes are used (as seen in Figure 12). Moving on, the sum of the number of pitches seen in each part individually (plot j) is clearly going to be higher: after all, some pitches, like middle C, can occur in all four parts. As would be expected, there is a higher rate of increase in this measure with increasing cross-entropy. None of the samples came very close to employing all possible pitches; the maximum number was 65, compared with a possible maximum of about 80.

Continuing with Figure 13, a plot of the entropy of a distribution comprising chords with roots on the tonic, supertonic, mediant, and so on against cross-entropy (plot k) reveals something interesting. At very low cross-entropy, the chord distribution entropy is very low, indicating a reliance on a few common chords such as tonic and dominant. From here the chord entropy rises rapidly, indicating that within a short cross-entropy span the chord usage has become much more diverse. This suggests that extremely low cross-entropies should be avoided if harmonically interesting solutions are desired. The use of $t = 0.05$ and $t = 0$ introduces increasingly greater chord diversity. Note that the automatic chord identification procedure looks for complete triads, triads with fifth missing, complete seventh chords, sevenths with third missing and sevenths with fifth missing. In addition, it looks for the dominant seventh of the three most closely related keys and their triad counterparts; but the chords are identified as tonic, supertonic and mediant of the original key (chromatically altered). Finally (in plot l), we examine the maximum number of consecutive leaps of greater than a perfect fourth in the bass (Suzuki & Kitahara, 2014, criterion 5). Too many consecutive leaps are best avoided to achieve a coherent sounding bass line. There is a definite trend towards more consecutive leaps with increasing cross-entropy. Optimized t produces strange looking results, however, with the trend suddenly reversing just before 3 bits/chord.

4.4 Example harmonizations

Several harmonizations of two hymn tunes are now examined, paying particular attention to non-diatonic notes, structural aspects such as cadences, and the general rules of harmony used above for objective evaluation. The harmonizations have widely varying cross-entropies, having been generated by iterative random walk using different values of t.

Figure 14 shows musical scores of harmonizations of hymn tune *Bromsgrove* (hymn no. 144, Vaughan Williams, 1933). It should be noted that the spelling of MIDI pitch values appearing in the output is sometimes ambiguous (e.g. between F♯ and G♭); therefore in such cases a reasonable spelling (in context) was chosen by the authors. The score at the top, at 6.08 bits/chord, has the lowest cross-entropy from amongst solutions produced for this tune by standard random walk ($t = 0$). This piece contains many non-diatonic notes; for example the B♮ at the beginning of the second bar (measure), which suggests an early modulation from B flat major to the

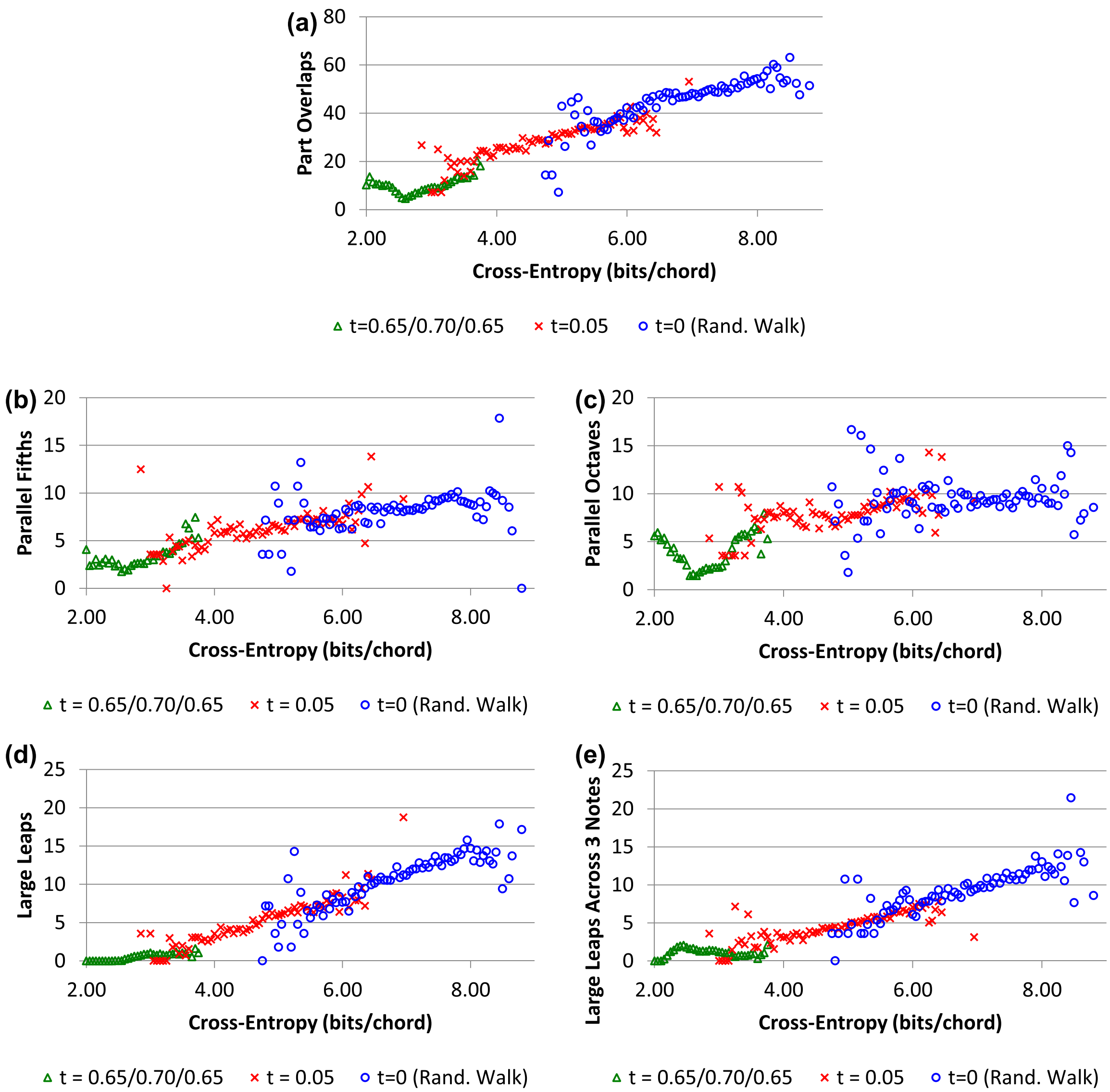

Fig. 11. Plots of cross-entropy (mid-points of 0.05 bits/chord bins) against number of rule violations on a per 100 melody note basis, for the 15,360 samples of generated harmony described in the caption to Figure 4. There are plots for part overlaps between adjacent chords, parallel fifths, parallel octaves, large leaps (11 semitones or greater than an octave) between adjacent notes in a part, and large leaps across three consecutive notes.

relatively distant key of C. The B♭ in the alto of the second quaver (eighth note) of the bar rules this out, however. Other non-diatonic notes are used equally indiscriminately, such as the B♮ at the end of bar 7 and the C♯ at the end of bar 14, which clash with C in the soprano. Four notes are tied across bar lines, which is highly unusual for this corpus. There are obvious part overlaps in bars 3 and 13, and the tenor and bass parts actually cross in bar 6 (had the tenor B♭ been a passing note, the cadence would have matched rare examples in the corpus). The bass line ascends from B♭2 to D4 (a major tenth) within three notes in bars 4 and 5; the use of such intervals is discouraged in text books on harmony. In the first minim (half note) of bar 13, the interval from tenor to alto is unusually large: normally, the interval would be no greater than an octave. There is a parallel fifth between bass and tenor in bars 15 and 16. Although the piece starts strongly enough on a tonic root position chord, it ends weakly on a tonic first inversion. Subjectively, overall, the piece sounds quite chaotic.

The second score in Figure 14 has the lowest cross-entropy (4.31 bits/chord) amongst samples generated using $t = 0.05$. This piece has only two non-diatonic notes: an A♭ in bar 5 and an E♮ in bar 12. This time there are no notes tied across bar lines, which is in accordance with the corpus. There are still part overlaps, for example between tenor and bass in bar 1; but there is no part crossing. Although the lower three parts are generally somewhat smoother than before, there is an interval of a major tenth within three consecutive bass notes in bars 5 and 6. There are parallel unisons between

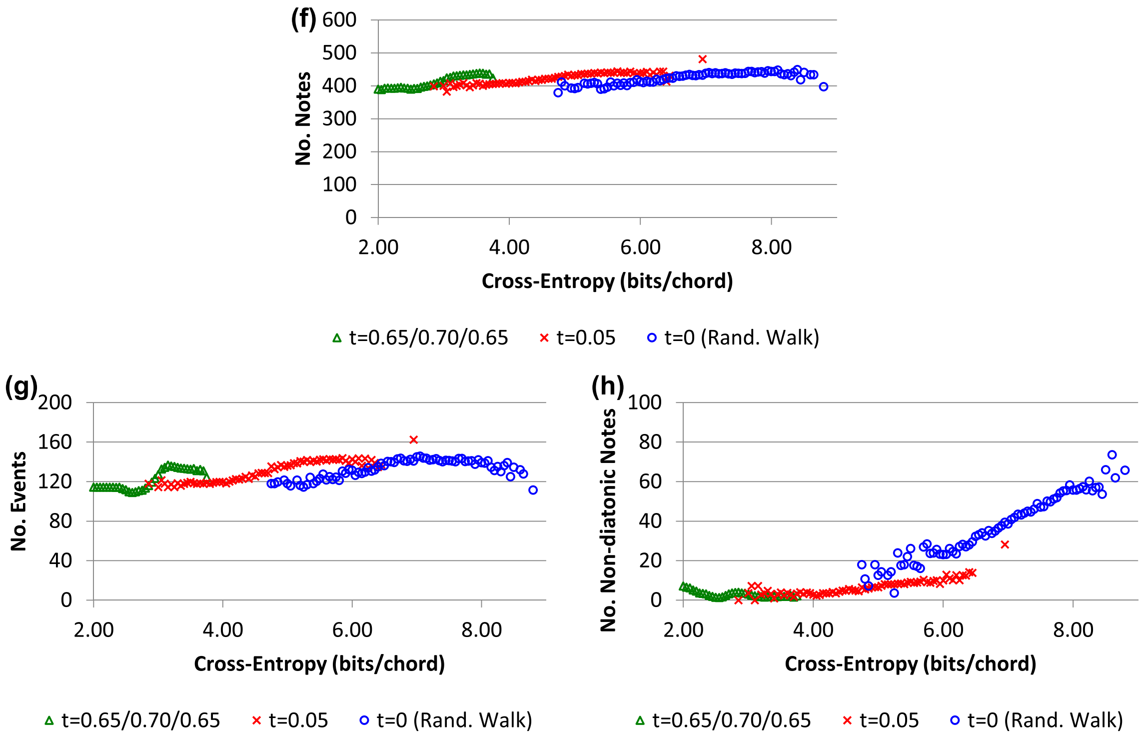

Fig. 12. Plots of cross-entropy (mid-points of 0.05 bits/chord bins) against diversity measures on a per 100 melody note basis, for the 15,360 samples of generated harmony described in the caption to Figure 4. There are plots for the number of notes, the number of events, and the number of non-diatonic notes.

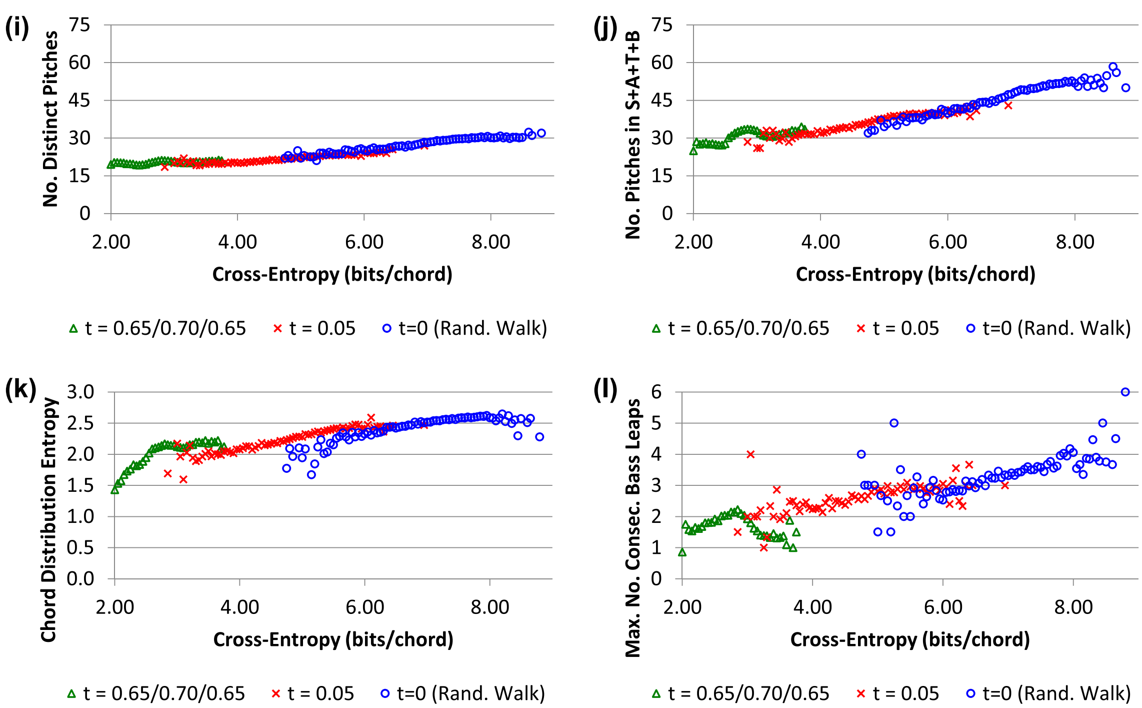

Fig. 13. Plots of cross-entropy (mid-points of 0.05 bits/chord bins) against diversity measures, for the 15,360 samples of generated harmony described in the caption to Figure 4. There are plots for the number of distinct pitches (irrespective of part); the sum of the number of distinct pitches in each part; the entropy of a distribution comprising chords with roots on the tonic, supertonic, mediant, and so on; and the maximum number of consecutive leaps of greater than a perfect fourth in the bass. These plots are not on a per 100 melody note basis.

Fig. 14. Musical scores of the lowest cross-entropy harmonizations of hymn tune *Bromsgrove* (hymn no. 144, Vaughan Williams, 1933) produced by $t = 0$ (standard random walk, top, 6.08 bits/chord), $t = 0.05$ (middle, 4.31 bits/chord) and $t = 0.65/0.70/0.65$ (optimized for `Duration`/`Cont`/`Pitch` respectively, bottom, 2.95 bits/chord).

tenor and bass, and parallel fifths between these parts and the soprano, in bar 6. The piece begins and ends on a tonic root position chord; but the alto and tenor notes are more widely separated than is usual (especially in these structurally important positions). This piece definitely has more overall harmonic coherence.

The last of this set of scores, at 2.95 bits/chord, has the lowest cross-entropy from amongst samples produced using optimized thresholds. This piece also has only two non-diatonic notes, and there are still a few part overlaps, such as between the soprano and alto in bar 11. The lower three parts are now much smoother, without any unusually large leaps (the bass part is the least smooth of the three, as would be expected from the corpus). There is a parallel fifth between tenor and bass in bars 9 and 10. The piece begins on a well distributed tonic root position chord, but concludes very weakly with tonic first inversion chords throughout the last two bars. The highlight of the piece can be found in bar 13, where the alto line might have been considered well thought out if written by a human composer. Notice too the contrary motion between soprano and tenor in bar 3, and the properly prepared and resolved passing $\frac{6}{4}$ at the end of the same bar. This piece is the most harmonically coherent of the three.

Figure 15 shows musical scores of harmonizations of hymn tune *Das walt' Gott Vater* (hymn no. 36, Vaughan Williams, 1933). The first score, at 5.33 bits/chord, has the lowest cross-entropy from amongst solutions produced for this tune by standard random walk. This piece contains surprisingly few

Fig. 15. Musical scores of the lowest cross-entropy harmonizations of hymn tune *Das walt' Gott Vater* (hymn no. 36, Vaughan Williams, 1933) produced by t = 0 (standard random walk, top, 5.33 bits/chord), t = 0.05 (second, 3.21 bits/chord), t = 0.65/0.70/0.65 (optimized for `Duration`/`Cont`/`Pitch` respectively, third, 2.37 bits/chord) and t = 0.45/0.70/0.30 (optimized specifically for this tune, bottom, 2.04 bits/chord). Partially derived from Whorley and Conklin (2015), Figure 5.

non-diatonic notes, with the majority of them occurring in bar 4 where they undermine the cadence. There are three part overlaps in bars 3 and 4: two between tenor and bass, and one between alto and tenor. An interval of a major seventh occurs within three consecutive bass notes in bar 6; but generally the parts are fairly smooth. There is a parallel unison between soprano and alto in bar 2. Although the piece starts with the expected tonic root position chord, the second crotchet (quarter note) briefly changes the chord to a submediant before a return to the tonic at the beginning of bar 1. The tonic ending is good; but the alto passing or auxiliary (changing) note leading up to it is not properly resolved.

The second score has the lowest cross-entropy (3.21 bits/chord) from amongst samples generated using $t = 0.05$. This piece has no non-diatonic notes; although an A♮ would have been appropriate as part of a possible excursion to B flat major in bar 4. It also has no part overlaps, unusually large leaps, parallel fifths or parallel octaves. There is a well executed passing note in the tenor at the end of bar 2. The cadence at the end of the first phrase, however, is very badly constructed, finishing as it does on a second inversion chord. The first two chords of the piece are exactly as they appear in the hymnal harmonization; but ending the piece on a submediant chord is poor.

The third score, at 2.37 bits/chord, has the lowest cross-entropy from amongst samples produced using optimized thresholds. The harmony is generally acceptable, although there is a part overlap between alto and tenor in bar 3, and an interval of a major seventh within three bass notes in bars 3 and 4. There is also an unnecessary repeated crotchet A♭ in the alto in bar 4. On the positive side, the appoggiatura E♭ in the first chord of bar 3 is harmonically interesting, and the passing note in the bass in bar 7 is well conceived as part of a four note step-wise descent. The two perfect cadences, the plagal cadence at the end of the second phrase, and the imperfect cadence ending the third are all well constructed; this is noteworthy, considering that cadences are not specially modelled. The piece begins and ends appropriately on tonic root position chords.

Whereas the third score is generated using thresholds optimized to minimize the overall cross-entropy of harmonizations of five tunes, the final score (2.04 bits/chord) is produced using thresholds optimized for this particular tune. This time, there are two part overlaps in bar 3: between alto and tenor as before, and between tenor and bass. The latter eliminates the large interval previously present within three bass notes, however. In bar 4, the rather odd repeated crotchet A♭ in the alto has disappeared; but on the other hand passing notes now only appear in the soprano. Overall, the piece is harmonically well-formed, albeit with a very restricted palette of chords. Harmonizations of three further hymn tunes, namely *Innocents*, *Das ist meine Freude* and *Monte Cassino* (hymn nos. 37, 97 and 150, Vaughan Williams, 1933), are available for inspection: see Appendix D.

5. Discussion and conclusions

Statistical models of music are powerful and widely applicable, because they can be learned from a corpus and then used for music prediction, classification or generation. In the case of music generation, issues often discussed in relation to the family of context models are their lack of long-range coherence and wandering nature. In this article it was hypothesized that both of these problems could be addressed to a large degree by searching for low cross-entropy solutions. A new approach was developed, which combines iterative random walk with probability thresholds. This method enables multiple low cross-entropy solutions to be found within far fewer samples, and therefore in a much shorter time, than when using standard iterative random walk.

The new method was tested on the task of melody harmonization. Samples of harmony were generated by a statistical model of chords based on the multiple viewpoint framework. To provide an objective motivation for finding low cross-entropy solutions, some general rules of harmony were used to study the relationship between cross-entropy and the quality of the harmony. By first of all using standard iterative random walk on a set of five existing hymn tunes, we confirmed that this sampling method cannot reach into the low cross-entropy space within a practical number of samples, and convincingly demonstrated a correlation between cross-entropy and harmonization quality. With probability thresholds added to the iterative random walk method, we clearly showed that low cross-entropy solutions could quickly and easily be found, and that the correlation between cross-entropy and musical quality extends to the very low cross-entropy regions reached by optimizing the thresholds for each individual musical attribute.

A subjective evaluation of the method was also carried out, by studying in detail harmonizations of two existing melodies which were generated using different threshold values. It was found that low cross-entropy solutions have more harmonic coherence and better cadences; and they seem to be closer in overall quality and style to pieces in the corpus than higher cross-entropy solutions. The lowest cross-entropy solutions tend to be very conservative in their chord progressions due to the generally high probability chord transitions used. Some residual problems with statistical models remain, which could be overcome by the use of simple constraints at certain melodic positions; for example, by specifying phrase ending cadences. On the other hand, it would be interesting to pursue solutions purely within the realm of statistical modelling.

As far as future research is concerned, several interesting lines are suggested. First, a natural consequence of the probability threshold method is that some acceptable chords that do not meet the thresholding requirements are discarded along with unacceptable ones; therefore as thresholds increase, the diversity of reachable harmonizations of a given melody will decrease. It will be interesting to explore whether there exist thresholds that achieve close to optimal diversity, while at the same time maintaining the speed of the probability threshold method (i.e. we would not wish to significantly increase the number of samples taken). Second, in this article some optimization was performed in order to determine good values for the probability thresholds in terms of their ability to reach into low cross-entropy regions: the possibility of inferring good default values for thresholds heuristically for a given statistical model should be investigated. Third, some other ways of applying the threshold method could be explored; for example, by successively reducing the threshold while at each stage generating slightly more samples of harmony. Fourth, the fact that four out of the five rule violation types are due to a mismatch between adjacent chords suggests a more radical approach to sampling. The mismatches must occur because of the influence of viewpoints where back-off has occurred beyond the point of any conditioning context. By keeping track of the degree of back-off required to assign each chord's probability within a viewpoint's distribution, it may be possible to dynamically determine which chords to discard: effectively, the probability thresholds would vary throughout the generation of a harmonization. Fifth, this method can be used in conjunction with models which assume a different given part, such as adding soprano, alto and tenor to a known bass; and even with models which generate all four parts (modelling of melody and harmony jointly). Finally, a logical step would be to compare this method with other sampling and optimization methods, such as Gibbs sampling, Metropolis sampling, Metropolis–Hastings sampling and variable neighbourhood search.

In conclusion, we find that the harmony of music generated by standard iterative random walk from a statistical model improves (in the sense that there are fewer violations of rules of harmony) as cross-entropy reduces. The solution space can be shifted to lower regions of the cross-entropy range, thereby generally improving the harmony in the sample distribution, by applying probability thresholds to iterative random walk. Various diversity measures also show trends with respect to cross-entropy. The most striking of these is for the number of non-diatonic notes per 100 melody notes, where for a probability threshold of zero (standard iterative random walk) there is a rapid increase in this measure with increasing cross-entropy. This underlines the harmonic incoherence found in this sample distribution as measured by the number of rule violations. Minimal cross-entropy solutions can be found by optimizing the probability threshold for the generation of individual musical attributes for single melodies. This method of finding minimal cross-entropy solutions in a time-efficient way is applicable to any type of sequence, and so is likely to be of use in other areas of investigation.

Acknowledgements

Special thanks to Kerstin Neubarth for valuable comments on the manuscript.

Funding

This research is partially supported by the Lrn2Cre8 project 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. & Williams, C.K.I. (2005). Harmonising chorales by probabilistic inference. In L.K. Saul, Y. Weiss & L. Bottou (Eds.), *Advances in neural information processing systems* (Vol. 17, pp. 25–32). Cambridge, MA: MIT Press.

Biyikoğlu, K.M. (2003). A Markov model for chorale harmonization. In R. Kopiez, A.C. Lehmann, I. Wolther & C. Wolf (Eds.), *Proceedings of the 5th Triennial European Society for the Cognitive Sciences of Music (ESCOM) Conference* (pp. 81–84). Osnabrück: epOs-Music.

Cherla, S., Weyde, T., & d'Avila Garcez, A. (2014). Multiple viewpoint melodic prediction with fixed-context neural networks. In *Proceedings of the 15th International Society for Music Information Retrieval (ISMIR) Conference* (pp. 101–106). Canada: International Society for Music Information Retrieval.

Chordia, P., Sastry, A., & Şentürk, S. (2011). Predictive tabla modelling using variable-length Markov and hidden Markov models. *Journal of New Music Research, 40* (2), 105–118.

Chuan, C.-H., & Chew, E. (2011). Generating and evaluating musical harmonizations that emulate style. *Computer Music Journal, 35* (4), 64–82.

Cleary, J.G., & Witten, I.H. (1984). Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications, COM-32* (4), 396–402.

Conklin, D. (2002). Representation and discovery of vertical patterns in music. In C. Anagnostopoulou, M. Ferrand & A. Smaill (Eds.), *Music and Artificial Intelligence: Proceedings of ICMAI 2002* (LNAI Vol. 2445, pp. 32–42). Berlin: Springer-Verlag.

Conklin, D. (2003). Music generation from statistical models. In *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences* (pp. 30–35). London: AISB Society.

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

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

Evans, B., Fukayama, S., Goto, M., Munekata, N., & Ono, T. (2014, September). AutoChorusCreator: Four-part chorus generator with musical feature control, using search spaces constructed fromrules of music theory. Paper presented at the Joint International Computer Music Conference and Sound and Music Conference. Athens, Greece.

Farbood, M. & Schoner, B. (2001, September). Analysis and synthesis of Palestrina-style counterpoint using Markov chains. Paper presented at the International Computer Music Conference. Havana, Cuba.

Fernandez, J.D. & Vico, F.J. (2013). AI methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research, 48*, 513–582.

Groves, R. (2013). Automatic harmonization using a hidden semi-Markov model. In *Musical Metacreation: Papers from the 2013 AIIDE Workshop* (pp. 48-54). Palo Alto, CA: AAAI Press.

Hedges, T., Roy, P., & Pachet, F. (2014). Predicting the composer and style of jazz chord progressions. *Journal of New Music Research, 43* (3), 276–290.

Herremans, D., Sörensen, K., & Conklin, D. (2014, September). Sampling the extrema from statistical models of music with variable neighbourhood search. Paper presented at the Joint International Computer Music Conference and Sound and Music Conference. Athens, Greece.

Jurafsky, D., & Martin, J.H. (2000). *Speech and language processing*. Upper Saddle River, NJ: Prentice-Hall.

Kirkpatrick, S., Gelatt Jr., C.D., & Vecchi, M.P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Lo, M.Y. & Lucas, S.M. (2006). Evolving musical sequences with n-gram based trainable fitness functions. In *Proceedings of the 8th IEEE Congress on Evolutionary Computation* (pp. 601–608). Piscataway, NJ: IEEE.

Pachet, F. & Roy, P. (2011). Markov constraints: Steerable generation of Markov sequences. *Constraints, 16* (2), 148–172.

Pearce, M.T. (2005). *The construction and evaluation of statistical models of melodic structure in music perception and composition* (PhD thesis). Department of Computing, City University, London, UK.

Pearce, M.T., Conklin, D., & Wiggins, G.A. (2005). Methods for combining statistical models of music. In U.K. Wiil (Ed.), *Computer music modelling and retrieval* (pp. 295–312). Berlin: Springer.

Pearce, M.T., Müllensiefen, D., & Wiggins, G.A. (2010). Melodic grouping in music information retrieval: New methods and applications. In Z.W. Ras & A. Wieczorkowska (Eds.), *Advances in music information retrieval* (pp. 364–388). Berlin: Springer.

Pearce, M.T., & Wiggins, G.A. (2006). Expectation in melody: The influence of context and learning. *Music Perception, 23* (5), 377–405.

Pearce, M.T., & Wiggins, G.A. (2007, June). Evaluating cognitive models of musical composition. Paper presented at the 4th International Joint Workshop on Computational Creativity. London, UK.

Piston, W. (1976). *Harmony*. London: Victor Gollancz Ltd.

Ponsford, D., Wiggins, G.A., & Mellish, C. (1999). Statistical learning of harmonic movement. *Journal of New Music Research, 28* (2), 150–177.

Potter, K., Wiggins, G.A., & Pearce, M.T. (2007). Towards greater objectivity in music theory: Information-dynamic analysis of minimalist music. *Musicae Scientiae, 11* (2), 295–324.

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

Raphael, C., & Stoddard, J. (2003). Harmonic analysis with probabilistic graphical models. *Proceedings of the 4th International Society for Music Information Retrieval (ISMIR) Conference* (pp. 177–181). Baltimore, MD: Johns Hopkins University.

Rohrmeier, M. & Graepel, T. (2012). Comparing feature-based models of harmony. In *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval* (pp. 357–370). London: Springer.

Suzuki, S., & 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.

Vaughan Williams, R. (Ed.). (1933). *The English hymnal* (2nd ed.). Oxford: Oxford University Press.

Viterbi, A.J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory, 13*, 260–269.

Whorley, R.P. (2013). *The construction and evaluation of statistical models of melody and harmony*. (PhD thesis). Department of Computing, Goldsmiths, University of London, UK.

Whorley, R.P., & Conklin, D. (2015). Improved iterative random walk for four-part harmonization. In T. Collins, D. Meredith & A. Volk (Eds.), *Mathematics and computation in music* (LNAI 9110, pp. 64–70). Berlin: Springer.

Whorley, R.P., Wiggins, G.A., Rhodes, C., & 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.

Witten, I.H., & Bell, T.C. (1989). *The zero frequency problem: Estimating the probability of novel events in adaptive text compression* (Technical Report, No. 89/347/09). Calgary, Canada: The University of Calgary, Department of Computer Science.

Yu, S.-Z. (2010). Hidden semi-Markov model. *Artificial Intelligence, 174* (2), 215–243.

Appendix A. Specific multiple viewpoint systems used

Attributes are generated using specially selected multiple viewpoint systems, with each viewpoint in a system contributing to the generation of a single attribute. Lower cross-entropies can be achieved in this way than by employing a single system capable of generating all attributes (Whorley, 2013). Recall that $\otimes$ implies linking and $\ominus$ implies threading. The symbol $\hbar$ is the maximum order of the model, and a bias is used to favour low-entropy probability distributions with respect to distribution combination. Basic viewpoints `Cont`, `Duration`, `Phrase`, `Piece` and `Pitch` are described in Section 3.1; and derived viewpoints used in the systems below are briefly defined in Table A1. The attribute `Duration` has the system

{`DurRatio` ⊗ `Phrase`, `Duration` ⊗ `PositionInBar`, `Duration` ⊗ `LastInPhrase`, `DurRatio` ⊗ (`IOI` ⊖ `FirstInBar`), `DurRatio` ⊗ `TactusPositionInBar`, `DurRatio` ⊗ (`ScaleDegree` ⊖ `LastInPhrase`), `Duration` ⊗ `Metre`, `Duration` ⊗ (`ScaleDegree` ⊖

Table A1. Derived viewpoints used in this research, in alphabetical order. Basic viewpoints are described in Section 3.1.

Viewpoint	Meaning
Contour	descending, level or ascending pitch
DurRatio	sequential duration ratio
FirstInBar	first beat of bar, or otherwise
FirstInPhrase	first event in phrase, or otherwise
FirstInPiece	first event in piece, or otherwise
InScale	in scale denoted by KeySig and Mode
Interval	sequential pitch interval (semitones)
IOI	difference in start-time of notes
LastInPhrase	last event in phrase, or otherwise
Metre	metrical strength (3 strongest, 0 weakest)
PositionInBar	number of Duration units from start of bar
ScaleDegree	pitch interval (semitones) above tonic
Tactus	on one of main beats of bar, or otherwise
TactusPositionInBar	number of main beats from start of bar

LastInPhrase), Duration ⊗ (IOI ⊖ FirstInBar), DurRatio ⊗ (Interval ⊖ FirstInBar)}

with an $\hbar$ of 0 and a bias of 5.9; the attribute Cont has the system

{Cont ⊗ Interval, Cont ⊗ Metre, Cont ⊗ TactusPositionInBar, Cont ⊗ (Contour ⊖ Tactus), Cont ⊗ Tactus, Cont ⊗ (ScaleDegree ⊖ FirstInPhrase)}

with an $\hbar$ of 2 and a bias of 0; and the attribute Pitch has the system

{Cont ⊗ ScaleDegree, Cont ⊗ Interval, ScaleDegree ⊗ LastInPhrase, (Pitch ⊖ Tactus) ⊗ InScale, Contour ⊗ InScale, (ScaleDegree ⊖ FirstInPhrase) ⊗ FirstInPiece, ScaleDegree ⊖ Tactus, Interval ⊗ InScale, Cont ⊗ (ScaleDegree ⊖ LastInPhrase), (ScaleDegree ⊖ FirstInPhrase) ⊗ Piece, ScaleDegree ⊗ Metre}

with an $\hbar$ of 4 and a bias of 1.1 (Whorley, 2013). We use this model to examine a new method for generating high probability solutions.

Appendix B. Rules of harmony used for evaluation

The general rules of harmony used for evaluation purposes are described below, along with the scoring system.

part overlap This occurs when, for example, a soprano note is lower than the previous alto note, or an alto note is higher than the previous soprano note. Each such occurrence is given an evaluation score of 1. A part overlapping with two other parts receives a score of 2, on the basis that this indicates even worse harmony. Similarly, in the unlikely event that a part overlaps with three other parts, a score of 3 results. All parts are checked against all other parts.

parallel fifth Two parts, a perfect fifth apart (modulo octave), moving such that they are still a perfect fifth apart (modulo octave). All parts are checked against all other parts, and all such occurrences receive an evaluation score of 1.

parallel octave Two parts, in unison or an integer number of octaves apart, moving such that they are still in unison or an integer number of octaves apart. All parts are checked against all other parts, and all such occurrences receive an evaluation score of 1.

large leap (2 notes) An interval of 11 semitones or greater than an octave between adjacent notes in the same part. All such occurrences receive an evaluation score of 1.

large leap (3 notes) An interval of 11 semitones or greater than an octave across three consecutive notes in the same part. All such occurrences receive an evaluation score of 1.

Appendix C. Corpus hymn tune harmonizations

The corpus comprises the following hymn tune harmonizations from Vaughan Williams (1933): 4, 5, 9, 11, 14 (modern tune), 15, 16, 17 (modern tune), 20, 21, 23, 26, 29, 31 (modern tune), 32, 34, 35, 39, 40, 42, 43, 45, 49 (modern tune), 52 (modern tune), 56 (modern tune), 61 (modern tune), 63, 71, 75 (first tune), 80, 81, 82, 83, 85, 86, 91, 93, 98, 102, 103, 104, 105, 106, 107, 115, 115 (first alternative tune), 118, 119, 120, 126, 128, 131, 132, 133, 133 (original version), 134 (alternative tune), 135, 137, 138, 139, 145, 147, 148, 151 (modern version), 155 (modern tune), 156, 158, 162, 167, 168, 169 (modern tune), 171, 173, 177, 179, 180 (modern tune), 186, 187, 190, 193, 195, 196, 197, 199, 201, 204, 206, 209, 211, 213 (modern tune), 216, 217, 218, 220, 222, 223 (modern tune), 267, 316, 452 and 485.

Appendix D. Additional example harmonizations

Harmonizations of hymn tunes *Innocents*, *Das ist meine Freude* and *Monte Cassino* (hymn nos. 37, 97 and 150, Vaughan Williams, 1933) are available for inspection in Figures D1 to D3.

Fig. D1. Musical scores of the lowest cross-entropy harmonizations of hymn tune *Innocents* (hymn no. 37, Vaughan Williams, 1933) produced by t = 0 (standard random walk, top, 4.75 bits/chord), t = 0.05 (middle, 2.86 bits/chord) and t = 0.65/0.70/0.65 (optimized for `Duration`/`Cont`/`Pitch` respectively, bottom, 2.01 bits/chord).

Fig. D2. Musical scores of the lowest cross-entropy harmonizations of hymn tune *Das ist meine Freude* (hymn no. 97, Vaughan Williams, 1933) produced by $t = 0$ (standard random walk, top, 6.11 bits/chord), $t = 0.05$ (middle, 3.88 bits/chord) and $t = 0.65/0.70/0.65$ (optimized for `Duration`/`Cont`/`Pitch` respectively, bottom, 2.42 bits/chord).

Fig. D3. Musical scores of the lowest cross-entropy harmonizations of hymn tune *Monte Cassino* (hymn no. 150, Vaughan Williams, 1933) produced by t = 0 (standard random walk, top, 5.55 bits/chord), t = 0.05 (middle, 3.94 bits/chord) and t = 0.65/0.70/0.65 (optimized for `Duration`/`Cont`/`Pitch` respectively, bottom, 2.56 bits/chord).