Harmonizing Music the Boltzmann Way

M. I. BELLGARD & C. P. TSANG

Music harmonization has long been recognized as a highly intellectual process. Musicologists have studied music pieces by great composers and formulated a number of symbolic rules. However, these rules usually form only a set of heuristics and may not be absolutely precise. In this paper, we demonstrate how to train a Boltzmann machine to capture these syntactic rules and use it to construct an effective Boltzmann machine (EBM) to harmonize some unseen pieces. We have also incorporated ways to apply absolute constraints to the completion process. Our experiments demonstrate that using an EBM, 'good' harmonies can be non-deterministically synthesized along with a relative measure of their quality.

KEYWORDS: Boltzmann machine, music harmonization, music composition, machine learning.

1. Introduction

The period between the 17th and 18th centuries saw the emergence and development of tonal harmony in western music. The underlying principles of tonal harmony may be found in chorale harmonization, which is also known as four-part writing. Chorale harmonization is considered a highly intellectual, time-consuming task and to produce a notable harmonization requires a wealth of knowledge and experience on the part of the composer (Piston, 1978; Siegmeister, 1966).

The formalization of four-part writing generally involves an analysis of musical pieces by musicologists. They formulate this process in terms of rules, heuristics and symbolic descriptions of the abstraction. However, as with most arts, the original composers did not write down the rules governing their works and the musicological analysis has been performed long after their demise. As a result, these analyses cannot be confirmed as entirely correct. The derived rules are only as good as both the analysis and the expressiveness of the symbolic representation employed. Existing computer harmonization systems have been generally designed by utilizing this human knowledge to search for an appropriate combination of musical notes. As a consequence, symbolic artificial intelligence (AI) systems cannot entirely
capture the musical style of a human composer. The quality of the harmonies synthesized by these systems has fallen short in comparison to those produced by experienced human composers.

Given the advances in machine learning in recent years, it is desirable to apply its learning techniques to music harmony in an attempt to build a system that learns to harmonize unseen melodies as well as to distinguish between different musical styles. However, symbolic rule-learning mechanisms are unsuitable for imprecise and noisy processes, like music harmonization. Alternatively, the choice of artificial neural network (ANN) methodologies appears to be a natural one (Hild et al., 1991; Kohonen, 1989; Mozer, 1991; Todd, 1989). HARMONET (Hild et al., 1991) is an example of a neural system for chorale harmonization. However, a major limitation of this system is its reliance on a feedforward neural architecture. As a result, HARMONET's harmonization process assumes a sequential operation, where it harmonizes a chorale melody from left to right, in a deterministic manner.

With the intention of building a machine-learning music harmonization system, we have designed a learning system based on the Boltzmann machine (BM) (Aarts & Korst, 1988; Hinton & Sejnowski, 1986). We use a BM to learn the local contexts of a set of chorales and use our effective Boltzmann machine (EBM) (Bellgard & Tsang, 1992; Tsang & Bellgard, 1990) to harmonize music via completion (Kohonen, 1978; Hertz et al., 1991). We have demonstrated that our system can learn a particular harmonization style from learned local contexts which are subsequently used to harmonize new melodies. This harmonization process is non-deterministic and may produce different results on different runs due to the EBM's stochastic nature. There is a correspondence between the quality of the harmonization and the energy value of the EBM. The energy value can be used to identify wrongly harmonized notes within a piece.

In this paper, we describe various design issues affecting the EBM and also describe the process of incorporating external constraints in the EBM. These external constraints are vital to the success of the EBM as a harmonization system. We describe and discuss a set of experiments that highlight the appropriateness of the EBM to chorale harmonization.

2. Background

2.1. Chorale Harmonization

A chorale comprises four voices: soprano, alto, tenor and bass. It is performed either by singing voices or by instruments. Chorale harmonization involves writing the alto, tenor and bass parts for a given soprano melody. A chorale may be viewed as a sequence of four-note chords where each note in the chord corresponds to a part. Melodic notes are grouped into phrases and a cadence indicates the end of a phrase.

A musical style may be abstracted as a set of rules, restrictions or guidelines that should be observed by a composer so that the harmony remains within the confines of the style. These syntactic procedures include restrictions on chordal progressions as well as guidelines for appropriate choices of parts. For example, in the Baroque style (Kerman, 1978), parallel fifths and parallel octaves are disallowed, and voice leading is taken into consideration (Piston, 1978). The chorales that are of interest for the current work are those chorales composed in the Baroque period (17th and 18th centuries) (Dorffel, 1950).
Harmonizing Music the Boltzmann Way

The process of harmonization as described by Walter Piston (Piston, 1978 p. 152) states:

True harmonisation, then, means a consideration of the alternatives in available chords, the reasoned selection of one of these alternatives, and the tasteful arrangement of the texture of the added parts with due regard for consistency of style.

A student learning the harmonization process would discover from the many treatises on music harmony (Denny, 1961; Piston, 1978; Siegmeister, 1966; Prout, 1903; Tunley, 1990) that chorales may be harmonized in one or in a combination of three methods. Firstly, a chorale may be viewed as a sequence of chords where chord choices are made on their 'mutual connectibility' from chord to chord (Piston, 1978). The bass part is a consequence of the chord and does not possess its own melodic shape. This is contrasted with the second method, which initially considers the soprano and the bass parts, furnishing the bass part with its own melodic identity (Piston, 1978; Siegmeister, 1966; Tunley, 1990). The inner parts are subsequently added. The final method requires a memorization of groups of chords that are commonly recurring formulae or harmonic words (Piston, 1978; Tunley 1990). Equipped with an understanding of the above methods as well as the general laws of four-part writing procedures, the student would be able not only to analyze the works of great composers, but also harmonize new chorale melodies.

To date, there has been no definitive formalization of the harmonization process. Learning or capturing the salient features of another composer's style is not clearly defined, as the implementation of four-part writing is invariably unique to a composer. For instance, students eventually develop their own unique harmonization styles. Although musical analysis may articulate certain universal principles of well-known western musical styles, proceduralization of the process is a problematical issue. There is also difficulty in prescribing heuristic and less understood elements of a musical style (Gjerdingen, 1991; Loy, 1991). Thus, many informal, yet important aspects of harmonization are neglected. There have been a number of attempts to formalize this process using (Schenkerian) structural analysis (Oswald 1973) and linguistic models such as generative grammars (Lerdahl & Jakendoff, 1983; Winograd, 1968). All these models do involve, however, the human analytical process.

2.2. AllMusic Research and Chorale Harmonization

The difficulties faced by musicologists have not been a deterrent for the development of prescriptive, knowledge-based systems. CHORAL (Ebcioglu, 1988) is a rule-based system for harmonizing chorales in the style of J. S. Bach. The system captures musical knowledge from multiple viewpoints which observe: chord skeleton, individual melodic lines of the different voices and Schenkerian voice leading within the descant and bass. Tsang and Aitken (1991) developed a harmonizing system using constraint logic programming (CLP). The rules of harmonization were defined in terms of numerical constraints.

Apart from the issues outlined above, these systems typically suffer from two other problems. Firstly, the search space is enormous, exponentially increasing depending on the size of the melody (to be harmonized), the number of rules encoded in the system as well as the complexity of the rules. Secondly, it is left to the researcher or a musician to decide when to terminate the search and evaluate the quality of the harmonies synthesized.
2.2.1. Neural network approaches. With the difficulties of articulating harmonization knowledge explicitly, research in ANNs and music is currently receiving a great deal of attention (Bellgard & Tsang, 1992; Hild et al., 1991; Mozer, 1991; Todd, 1989; Todd & Loy, 1991). ANNs do not require the explication of rules as they have the ability to learn internal representations from a given set of examples. It is desirable that these internal representations capture the relevant concepts (Geman et al., 1992). ANNs have been used to generate monophonic melodies in a particular style (Todd, 1989; Mozer, 1991; Kohonen, 1989).

The HARMONET system. The HARMONET system (Hild et al., 1991) is a hybrid system combining both neural networks and symbolic methods to harmonize chorale melodies in the style of J. S. Bach. The music representation employed captures musically relevant symbolic information and also encodes look-ahead knowledge. Because the system operates in a linear fashion, harmonizing a melody (note by note) from start to finish, the look-ahead information is used in order to direct the harmony to an appropriate cadence. A feedforward neural network, employing the back-propagation learning algorithm (Rumelhart et al. 1986), is used to learn the harmonic skeleton (musically relevant information) of a set of example chorales. At the harmonization stage, the net will make predictions of the chord and bass part of a particular note in the melody using a fixed-length local context. Once the harmonic skeleton is determined by the neural network, symbolic algorithms are used to fill in the inner voices and also to insert passing notes. The symbolic algorithms ensure that constraints particular to the style are not violated. Such constraints are the prevention of parallel fifths and parallel octaves.

Analysis of some of the harmonies synthesized by HARMONET revealed that they contained parallel fifths and parallel octaves. For example, there are parallel octaves in the first and fifth bars of HARMONET's harmonization of Christus, der ist mein Leben, and there is a parallel fifth in the third bar of Happy Birthday (Hild et al., 1991). An instance of a parallel octave appearing in the first bar of Christus, der ist mein Leben is reproduced in Figure 1.

However, the symbolic algorithms incorporated in HARMONET were intended to prevent parallel fifths and parallel octaves from occurring in the harmony. This apparent contradiction is due to the hierarchical nature of HARMONET. Once HARMONET has decided on the harmonic skeleton using the neural network, the results from this process are then processed by the symbolic algorithms to fill in the inner parts. It is not possible to revise the harmonic skeleton once it is passed to

![Parallel Octaves](image)

Figure 1. The first few chords of Christus der ist mein Leben, harmonized by the HARMONET system reproduced from Hild et al. (1991). The arrows point to the parallel octave from the first chord to the second: C in the 4th octave moves to E in the 4th octave (soprano part) and at the same time C in the 2nd octave moves to E in the 2nd octave (bass part).
Harmonizing the Boltzmann Way

the next stage. Thus, an inappropriate selection of notes for the harmonic skeleton will lead to violations or inappropriate chords as the symbolic algorithms fill in the inner parts. A good example of this is the piece, Nicht so traurig, Nicht so sehr (Hild et al., 1991), where the inner parts progress abruptly through the harmonization. That is, the interval between consecutive pitches for both inner parts changes by relatively large amounts. This is unusual for chorales in the style of J. S. Bach. The output of the neural network in HARMONET is deterministic, thus only one harmonic skeleton is produced for a given melody.

2.3. Motivations for Current Work

2.3.1. Completion, chorale harmonization and constraint satisfaction. Pattern completion is the process by which a partial specification of information is completed (Hertz et al., 1991; Kohonen, 1987). Although feedforward networks are unable to perform this task, Hopfield and Boltzmann machine networks (Hinton & Sejnowski, 1986) can. Typically, if the information is binary encoded, then a partial specification of a binary pattern is clamped (Hinton & Sejnowski, 1986) to the completion device, which is then expected to ‘fill-in’ the missing bits of the pattern. The completion is constrained by both the partial input as well as what has been learnt by the completion device.

The chorale harmonization process may be viewed as the satisfaction of interdependent constraints via some completion process. The melody may be viewed as a partial input that must be completed, and the completion device (the composer) will ‘fill-in’ the other parts. During the completion of the melody, there are stylistic constraints as well as constraints imposed by the chorale melody. For example, for a given melody, a composer’s choice of a particular cadence would constrain the choice of harmonies for notes immediately preceding the cadence. An experienced human composer would observe all the harmony constraints, those which are well defined and can be articulated into precise, left-to-right syntactic rules (descriptive rules which can be easily implemented by knowledge-based systems) and those which cannot. As mentioned in Section 2.1, the completion process is unique to a composer.

2.3.2. Learning to harmonize using an effective Boltzmann machine. We propose a constraint satisfaction, completion-based learning system named an EBM for music harmony. A BM is trained on the local contexts (training set) of a set of chorales and an EBM is constructed from the BM to harmonize new melodies in a similar style. The local contexts learnt by the BM will resemble the constraints of the harmonization process. By using the EBM construction, a melody of any length may be harmonized. Unlike HARMONET, the EBM’s completion process is not directed. Thus, implicational constraints may hold in any direction. Because the system is non-deterministic, more than one harmony may be synthesized for a given melody. An energy measure is associated with each synthesized completion (each harmony). This measure may be used to indicate the quality of the harmony.

In summary, if it is required to complete a soprano melody, the EBM behaves as a harmonizing system. If no partial input is provided, the EBM behaves as a compositional system and, finally, if the complete piece is given as input the EBM may behave as an analysis system.
3. The Effective Boltzmann Machine

A description of the EBM may be found elsewhere (Bellgard & Tsang, 1992; Tsang & Bellgard, 1990) and hence only an overview will be given here. In the following sections, we briefly describe the music representation employed, how to learn the local contexts of the chorales and how to construct the EBM to synthesize harmonies for given chorale melodies.

3.1. Music Representation

In this music application, we use the pitch-height model (Barucha, 1991; Todd, 1989) to represent musical pitch. Each pitch is represented by a binary input/output (IO) unit. IO units are the visible units in the BM and EBM architecture. These IO units may be clamped or unclamped. Clamped IO units will act as inputs to the system as their values do not change during completion. A scale is a collection of all possible pitches. Each pitch is represented in the BM and the EBM by one IO unit. When the IO unit is activated to '1', the pitch associated with the unit is sounded, and vice versa. An event is an instance of a scale with some pitches sounded. In our particular application, each scale consists of 35 pitch units, two phrase control units and one spacer unit: a total of 38 IO units. As shown in Figure 2(a), the first 35 units correspond to a range of pitches from G in the second octave (G2) to F in the fifth octave (F5). The next two units in a scale represent start and end phrases respectively. Start/End phrase units are necessary to ensure that the system will recognize cadences at the end of a phrase, otherwise a cadence chordal progression could be placed anywhere in the phrase. The last unit indicates a spacer event (a '1' in this unit for an event implies that all other units in this event are set to '0'). The need for this unit will be detailed below. Figure 2(b) is a shorthand representation of a scale. All chorales are normalized to the same key before learning or completion commences.

The chorales used in the training set are taken from Choralbuch (Dorffel, 1950) which is used by musicians to study tonal harmony. As input to the system, each event describes a chord in a chorale. Hence, each event will have at most four pitch units clamped to '1' and the rest of the units clamped to '0'. Note that this is a very strong constraint, as in any four-part chorale there simply cannot be more than four voices sounding at any one time. We also make an assumption that each event is of the same duration. This length has the same length as the minimum length chord in all the chorales. Thus, chords held for twice this length in the chorale will be treated as two events in the system. Passing notes are omitted but there is no reason why they cannot be learned by this system at a later stage.

![Figure 2. (a) A diagrammatical representation of a scale. There are 38 units corresponding to IO units in both the BM and EBM formalism. The first 35 units correspond to pitches from G2 to F5. The next two units correspond to the start/end phrase units. The last unit indicates spacer events which are placed between chorales. (b) A shorthand representation of a scale to be used in subsequent diagrams.](image-url)
3.1.1. Error control codes and the pitch-height representation. It was discovered that a serious problem of conflicting contexts can arise in using the EBM for this music application (Bellgard & Tsang, 1992). To highlight the problem, let us assume that an EBM must harmonize a melody (a partial sequence) and that the partial sequence contains local contexts that are not contained in the training set. That is, the melody is not similar to any of the chorale melodies used in the training set. Completion of this sequence, using the EBM, will result in conflicts by the learned local contexts (the trained BM) as to how the partial sequence should be completed. As a result, the EBM synthesizes ill-formed sequences. Ill-formed sequences are those which contain events that do not have the right number of voices. For example, if the EBM produced a harmony for a given melody but there were voices missing from some chords or there were more than four voices in each chord, then this would constitute an ill-formed sequence.

To overcome this problem, two pieces of information are required. We need to know when an event is ill-formed, and when it happens, the EBM must attempt to correct the ill-formed event. The first piece of information relates to the representation of the events (discussed in this section) and the second relates to the incorporation of external constraints on the EBM to ensure well-formed events are synthesized (discussed later). The correction ability will depend on the information acquired during the learning stage. Is the pitch-height representation a sufficient encoding in order to convey this information?

Results from coding theory may be utilized to determine the appropriate representation. If the events in sequences are encoded using an error-control code (Farrell, 1979), then it will be possible to detect ill-formed events. The error-control code used in the EBM is a simple redundant code (Farrell, 1979). The redundant code encodes sequences by a fixed-length binary code (tuple) containing a fixed number of bits with the value ‘1’ in each code (parity). For four-part writing, the pitch-height model may be viewed as a redundant code. The tuple length in this case is 35 (the number of bits to represent an event) with a 4-bit parity (four voices for each chord). The information for determining ill-formed and well-formed events is readily available. We describe how ill-formed events are resolved in Section 4.2.1.

3.2. Learning the Local Contexts in Chorales

At this stage, a single BM is used to learn the local contexts of the chorales using the BM learning algorithm (Aarts & Korst, 1988; Hinton & Sejnowski, 1986). The BM topology consists of two sets of units: visible (IO) units and hidden units. There are bidirectional connections between the two sets of units and there are no lateral connections between units in the same set. Figure 3 describes a BM made up of three scales in the IO layer and a hidden layer. Each hidden unit is connected via a symmetric weight (represented collectively by \( W \)) to each unit in the IO layer.

The BM is governed by an energy function \( E \). The energy \( E \) of the BM is defined to be the summation of the weighted connections of pairs of units that are activated:

\[
E = - \sum_{i} \sum_{j} W_{ij} s_i s_j
\]

where \( s_i \) and \( s_j \) represent the state of the units in the BM and the \( W_{ij} \) corresponds to the set of symmetric weights, where \( W_{ij} \) connects unit \( i \) to unit \( j \).

Unit updates are made in relation to this energy function. The energy of a particular unit (known as the energy gap \( \Delta E \)) is determined by a difference in the
global energy of the system when that unit is either '1' or '0', and is given by

$$\Delta E_u = \sum_{j \neq i} W_{ij} s_j$$

In conjunction with the simulated annealing algorithm (Aarts & Korst, 1988; Kirkpatrick et al., 1983) the unit is stochastically updated according to the following probability acceptance criterion:

$$P_k = \frac{1}{1 + e^{-\Delta E_k/kT}}$$

The BM attempts to minimize global energy where the larger the energy gap for a particular unit, the more likely it will be activated to '1' (Aarts & Korst, 1988; Hinton & Sejnowski, 1986). There is a positive correlation between the energy gap of a unit and its probability of being activated. For this reason, the energy of the system and the energy gap of all units may be used as a relative measure of the degree of confidence the BM places in its completions.

It is important to note that the definition of the energy gap is derived from the attempt to minimize the energy of the system. Thus, the larger the unit's energy gap, the more likely it will be activated in order to minimize energy of the entire system.

To obtain the local contexts of a set of chorales, it is helpful to view the chorale as a sequence of events. The training patterns from the chorales are obtained from a sliding window consisting of $M$ scales. The window is initially set to the first $M$ events of the chorale. What the window can see is taken as a training pattern. The window then slides to the right, one event at a time. This process continues until all local contexts are obtained from the chorales.

When learning more than one chorale, the sliding window does not overlap over chorales. This is achieved by placing $M-1$ spacer events (spacer events are those with all units clamped to '0' except the last bit in the event, which is set to '1') between each chorale. This set of binary patterns constitutes the training set. The standard BM learning algorithm (Hinton & Sejnowski, 1986) is used to obtain a set of weights that characterizes the patterns in the training set. At the completion stage, the BM relaxes to one of the stored patterns non-deterministically, using simulated annealing (Aarts & Korst, 1988; Kirkpatrick et al., 1983). In other words, it is performing completion of a partially specified input. This trained BM is used to construct an EBM of arbitrary length.
3.3. Completion of an Arbitrary-length Melody

Given a BM which has learnt the local contexts of some chorales, a melody $S$ of length $L$ may be harmonized using the EBM. This melody will contain $L$ events with one unit in each event, set to '1', corresponding to the pitch in the melody for that event. The units in $S$ represent the IO layer of the EBM. An example melody $S$ is shown in Figure 4(a). The units in the events of $S$ may be clamped or unclamped. If they are unclamped, they are initially assigned '0' or '1' randomly.

An EBM may be constructed as shown schematically in Figure 4(b) to harmonize the melody in Figure 4(a). Assuming that the BM with a window size of $M = 3$, shown in Figure 3, is used to learn the local contexts of the chorales, then this machine is replicated and placed over $S$. There will be $L - M + 1$ copies of the BM. In Figure 4(b), each machine is enclosed by the dotted lines. An event in $S$ will correspond to a specific event in one or more individual BMs.

We enforce the constraint that the events in the IO layer of an individual BM are no longer independent of the events in the IO layer of another BM. The events in the IO layer of each individual BM are constrained to take on the values of the events in the IO layer of the EBM (i.e. the events in $S$). This implies that during the completion process, the individual BMs cannot be individually annealed. The

**Figure 4.** (a) A melody to be harmonized, consisting of five pitches and represented as five scales with one unit in each scale set to '1' corresponding to the pitch. (b) An example of an EBM constructed from the BM in Figure 3, completing a harmony for the melody in (a). The BM in Figure 3 is replicated three times. Each $B'$ has identical weights $W'$ and the same number of hidden units. All events appear in at least one copy of the BM $B$. For example, the 3rd event, which is shaded, appears in each $B'$ in a different location in the IO layer of each $B'$.
completion process is performed with respect to \( S \). That is, the entire system (not
the individual BMs) is subjected to the relaxation process to produce a solution (to
synthesize a harmony). As may be seen in Figure 4(b), unit updates in \( S \) (as the
system is relaxed) take into account the information arriving from neighbouring
contexts (from more than one individual BM) and the whole sequence \( S \) is
completed at the same time. As the system is slowly relaxed to a solution, the
unclamped units along with the hidden units will be updated such that the energy
function governing the system is minimized and a harmony produced. The energy
gap of a unit in \( S \) is the summation of the energy gaps of units in the respective
individual machines and its value is positively related to the consensus of the
respective individual machines. Further details may be obtained from our earlier
papers (Bellgard & Tsang, 1992; Tsang & Bellgard, 1990).

4. Experimentation

In this section, we present four different experiments. Five chorales with a total of
11 phrases, taken from *Choralbuch* (Dorfé, 1950), were chosen for the training set.
The BM was chosen with a window size of \( M = 5 \) and there were 80 training
patterns and 70 hidden units. It took about 25 minutes for each learning sweep of
the training set on a Sun SPARC 2 processor. Our results are based on a training
of approximately 400 sweeps. The experiment in Section 4.3 details a harmony that
is produced with a window size of \( M = 3 \) using the same five chorales.

4.1. Completion of a Chorale in Training Set

In this experiment, the soprano melody of one of the chorales in the training set
was clamped in the IO layer of the EBM. Hence, the soprano melody is the partial
input into the EBM. Figure 5 shows the score produced after the EBM is annealed.
This score is identical to the original score used in the training set. This
demonstrates that a completely correct recognition can be achieved by the EBM.
Completion took approximately 2 minutes on a Sun SPARC 2 processor.

This completion involved selecting 12 correct patterns (local contexts) from the
80 stored in the BM. Other experiments were also conducted. For example, the
bass line of the above piece was clamped to the IO layer of the EBM, which was
then annealed. The same chorale was synthesized. That is, given the bass part, the
EBM filled in the tenor, alto and soprano parts. In another experiment, the first
three and the last three chords of the above chorale were clamped and the EBM
filled in the rest of the piece.

These experiments demonstrate long-range effects by the EBM. For instance,
Harmonizing Music the Boltzmann Way

Figure 6. (a) The soprano melody of the first phrase of the chorale, Lobe den Herren, den machtigen König. (b) Harmony to the melody in Figure 6(a), produced by EBM. It can be seen that some events are not harmonized, while other events do not have exactly four parts.

the soprano melody provided sufficient context to recall the entire chorale. On the other hand, clamping the first and last three chords also provided sufficient context to complete the entire chorale.

4.2. Completion of Chorale Melody that Is Not in the Training Set

Using a chorale which is not in the training set, Lobe den Herren, den machtigen König, we clamped the soprano part of the first phrase of this chorale, shown in Figure 6(a), to the IO layer in the EBM. Figure 6(b) shows the score produced after the EBM is annealed. It can be seen that the EBM did not produce a correct harmony to every note in the melody. In fact, it is apparent that when given a new melody not in the training set, the EBM has no concept of how many notes to produce for each event. It also does not have any information telling it not to provide notes above the melody line. In addition to this, from the chorales provided to it in the training set, the EBM is unable to ensure that there should not be notes within two semitones below the melody. It is evident that we need to overcome these problems if we wish to obtain harmonies that are correct for this musical style.

4.2.1. Effects of absolute constraints. There are certain implicit rules that must be enforced by any composer (both human and artificial) to ensure a syntactically correct harmony. We have called these rules absolute constraints (ACs). Given a soprano melody, the system must not produce a harmony with notes above this melody line. Secondly, there must only be at most four notes for each event corresponding to the four different parts (voices). Finally, for this style of music, there must not be a note in the harmony within two semitones of the note in the melody.

Implicit in the nature of the system, the first and the third ACs can be easily satisfied by clamping the respective units to '0' in the IO layer before EBM is annealed. That is, all units above the soprano melody note and the two units below the melody note in each event are turned off.
For the second AC, where we must enforce at most four units on in each event, we introduce an encouragement/discouragement function. The function makes use of the pitch height (or error-control encoding, see Section 3.1). This function, at each temperature before the last few annealing temperatures, encourages the best $T$ units in each event (we have taken $T$ to be 10 in our experiments). This encouragement takes the form of an increment of energy $E$. The best $T$ units have their energy gap incremented by $E$ before they are annealed again. Towards the final annealing temperatures, we encourage only the best three units by $E$, and discourage the other seven units by a multiple of $E$ before annealing the units again. Figure 7 is a harmony of the same melody shown in Figure 6(a) but this time it is annealed with the encouragement function. It can be seen that the harmony produced is almost entirely correct except for the third and ninth events which, for this style, are incorrect chords. Apart from these two chords, as judged by a trained musician, this harmony is of comparable quality to the style it is mimicking.

4.2.2. Start/end phrase units. From our experimentation we discovered that the cadence point was not placed correctly when completing simply the events from a given melody. This is because there is no way to determine the start and the end of a phrase. Thus, start/end phrase units were included in the representation to ensure that a cadence chordal progression was completed at the end of a phrase. For example, when considering a phrase melody consisting of 15 events, each of the events would have one pitch unit activated (and clamped) to '1'. The rest of the pitch units in all events would be set to '0'. The first/last event would have the start/end phrase unit activated to '1' respectively. For events 2–14, the start and end phrase units would be clamped to '0'.

Although the cadence chordal progressions are prevented from appearing within the phrase, they cannot be forced to appear at the end of the phrase. Considering the above example, if the window size was $M = 5$, the context of event 1 and event 15 are only constrained by one BM window, while the contexts from event 5 to 11 are constrained by five staggered BM windows.

Because each of the events containing the activated start/end phrase units are constrained by one BM window, the events containing the activated start/end phrase units do not contribute sufficient information to their respective context. This implies that a cadence chordal progression may not necessarily be completed at the end of the phrase.

The above problem can be resolved by adding (if the window size was $M = 5$) four spacer events (see Section 3.1) before and after the phrase to be completed. In this way, all events in the phrase melody will appear in five BM windows each
Figure 8. The chorale melody shown in Figure 6(a) harmonized by an EBM constructed from a BM using a window size of $M = 3$. The BM was trained on the same set of chorales from *Choralbuch* (Dorffel, 1950) which were used to train the BM with a window size of $M = 5$.

(spacer events are added at both the learning and the completion stages). The harmony in Figure 7 was completed with spacer events.

4.3. Effects of a Different Window Size

In this experiment, we construct an EBM using a different BM, which used a window size of $M = 3$ to learn the local contexts of the same set of chorales taken from *Choralbuch* (Dorffel, 1950). This EBM (which incorporates absolute constraints) is used to harmonize the chorale melody in Figure 6(a). One of the harmonies produced by this EBM is shown in Figure 8. As is evident upon listening to this harmony, its quality is not as high as the harmony produced from a window $M = 5$, shown in Figure 7. Like the harmony in Figure 7, the third and ninth events are incorrect for this style of music. In addition, however, the bass part from the 4th to the 6th chords jumps abruptly, the fifth chord is not resolved properly to the 6th chord and there is an overlapping part from the 14th to the 15th chord. The harmony is of low quality as judged by a trained musician.

In comparison with the harmony in Figure 7, the poor quality of the harmony in Figure 8 would indicate that a window size of $M = 3$ is not a sufficiently large enough context to ensure a smooth progression from one chord to the next. Other experiments that we have conducted seem to confirm this.

4.4. Using the Energy Gap to Determine the Validity of the Harmony Produced by EBM

Despite the incorporation of ACs, it is still possible that the harmony synthesized for a particular event is not entirely appropriate. This is acceptable, since we have based our results on just five chorales and there are many contexts, as well as many chords, not present in the training set which have not been learnt. The system is currently unable to produce a correct harmony when it encounters these particular contexts.

We would like to have some indication of the validity of the harmony produced. We can achieve this by reporting the energy gap of each event of the completion that the EBM settled to. This is given by the sum of the energy gaps for the activated units in each event. Figure 9 shows five different harmonies produced by the EBM. The soprano part was taken from the chorale, *Nun lob, mein Seel, den Herren*, which is not part of the training set. Along with the score is the energy gap of each event. The BM trained with a window size of $M = 5$ was used in the construction.
As shown by Figure 9, events which the EBM is confident of, will have higher-energy gap values. For example, the first event and the last few events of each phrase in Figure 9 have a high energy. The harmony for Figure 9(d) is almost correct except for chord five. The corresponding energy graph reflects an error by the reduction in energy for this chord. The harmony in Figure 9(e) is totally correct if we are not too strict on overlapping parts. Once again, because of the nature of the system, the problem of overlapping parts may be overcome by clamping the relevant units in each event to '0' before the system is relaxed.

However, for events which the EBM is not so confident about, lower energy gaps are produced. In Figure 9(a), event 2 has the wrong note doubled, if we assume that the tenor and bass part both sound a B in the third octave, and event three is a wrong chord. This is reflected in the low energy gaps of events 2 and 3. In Figure 9(b), there are parallel octaves from event 5 to 6, which is reflected in the low energy gap of event 5. In Figure 9(c), the second event uses the chord I in its second inversion. This chord was never used in the training set, hence the low energy value for this event.

5. Conclusion

We have presented an application of the EBM to four-part chorale harmonization. We have demonstrated that an EBM can synthesize good harmonies after being trained on only a few chorales. We describe the harmonization process as a non-deterministic completion process where a melody, presented to the EBM for harmonization, is the partial input and may be of any length. The completion process is not sequential, completing the harmony, step by step, from left to right, and it is not solely dependent on local information of the previous step. The EBM will attempt to satisfy all constraints on the system simultaneously from the start of the completion process. In this way the completion process takes on a global perspective of the entire melody, as the EBM attempts to minimize the energy of the entire system.

The pitch-height representation behaves as an error-control code which enables the detection of ill-formed events and the correction ability is obtained from the learning stage. If appropriate local contexts are learnt by the BM, then the EBM is equipped with the necessary information to correct ill-formed events. The encouragement/discouragement function will bias the EBM (in its search for a solution) to a more correct one.

It was shown that as a consequence of the musical representation employed in the EBM, when harmonizing a soprano melody, the system is unaware that pitches should not be sounded above a melody pitch and that there should only be four pitches sounded in each chord. We have termed these constraints on four-part writing as absolute because it is imperative that they be observed during harmonization. To overcome these deficiencies in the EBM, we introduced methods for incorporating external (absolute) constraints (ACs) on the system. They are introduced into the stochastic approximation of the system as a selection bias and, depending on the particular AC incorporated, they do not necessarily possess the rigidity of a symbolic rule. For instance, the encouragement/discouragement function, which encourages the activation of four pitches in each chord of the chorale, is flexible enough to encourage only three pitches in a particular chord. This is the situation when two parts in a particular chord sound the same pitch.

It was demonstrated that the EBM approximates some long-range dependencies. Given just the melody of a chorale used in the training set, the EBM is able to
Harmonizing Music the Boltzmann Way

Figure 9. Different harmonies to the same melody produced by the EBM. Plotted beside each harmony is a plot of the energy gap of each event in that harmony. High energy values correspond to EBM's certainty of the harmony for that event, while low values of energy correspond to EBM's uncertainty of the harmony to produce for that event.
complete the rest of the chorale. In another experiment, only the first three and the last three chords of the same chorale were clamped, and the EBM completed the rest of the chorale. There are, of course, other long-range dependencies in music harmony that are not captured with the current system. However, it is envisaged they can be ascertained via a variation on the training method as well as with the incorporation of other ACs. We are currently investigating this important area.

We have shown that the EBM is able to relate the energy gap of events to the quality of the harmony. In the above experiments, a BM was trained with only five chorales. Thus, the EBM's judgements on synthesized harmonies are restricted to patterns found within the training set. Events completed by the EBM with relatively low energy levels correspond to harmonies that did not appear in any of the learned contexts obtained from the training set. In the experiments in this paper, only five chorales were used in the training set; however, a larger training set would equip the EBM with a greater repertoire. We are currently testing this situation and the initial results confirm this. While the computational requirement for this simulation is still too expensive for the harmonization of large musical works, there remains substantial scope for improvement. In particular, we are currently investigating the use of a deterministic BM (Hinton, 1989) to speed up this simulation.

From a music standpoint, this research is of great significance, because we have demonstrated that complex harmonization rules can be learnt by our EBM solely from examples of local contexts. It is significant that we can capture these contexts without requiring any musicological analysis and using only a simple pitch representation. The inherent nature of the EBM formalism to behave as a completion device is complementary to the simple musical representation employed. These properties of the EBM make it well suited to music harmonization. There are numerous avenues for this research. We are currently investigating how the EBM may be used to compare musical styles, and we are interested in listening to harmonies synthesized by an EBM which uses a BM trained on chorales in more than one key. Of course, training a BM on chorales from more than one musical style may enable the EBM to harmonize melodies in ways never before imagined.

This paper has concentrated on learning the harmonies for the chorales and has ignored meter and rhythm. Although meter and rhythm may be omitted from analysis for the Baroque style of music, it cannot be ignored in other styles, such as music from the Romantic Period (Kerman, 1978). Ways of incorporating rhythm and meter into the EBM formalism would broaden its potential application in this area.

Although we are applying the EBM to music harmony, its application is certainly not restricted to music. We believe that this method can be used for any complex sequences. We are currently applying the system to learning vowel harmony in natural language with success. The EBM cannot only inductively capture contexts, but can also incorporate known absolute constraints in the induction process. This opens up the possibility that domain knowledge can be incorporated within the neural network framework.

Acknowledgements

We would like to acknowledge the assistance of the following people: Professor David Tunley and Ralph Wilson from the School of Music Department at The University of Western Australia, and Peter Dodd, Leigh Smith and Mark Messenger of the Department of Computer Science, UWA.
References