Automatic Melodic Harmonization:
An overview, challenges and future directions

Dimos Makris, Ioannis Karydis and Spyros Sioutas
Department of Informatics, Ionian University, Greece
{c12makr, karydis, sioutas}@ionio.gr

ABSTRACT

Automatic melodic harmonization tackles the assignment of harmony content (musical chords) over a given melody. Probabilistic approaches to melodic harmonization utilize statistical information derived from a training dataset, producing harmonies that encapsulate some harmonic characteristics of the training dataset. Training data is usually annotated symbolic musical notation. In addition to the obvious musicological interest, different machine learning approaches and algorithms have been proposed for such a task, strengthening thus the challenge of efficient & effective music information utilisation using probabilistic systems. Consequently, the aim of this chapter is to provide an overview of the specific research domain as well as to shed light on the subtasks that have arisen and since evolved. Finally, new trends and future directions are discussed along with the challenges which still remain unsolved.

Keywords: Probabilistic Harmonization, Machine Learning, Algorithmic Composition, Survey, Idiom representations

INTRODUCTION

In music, harmony is the use of simultaneous pitches (tones, notes), or chords accompanying a given melody (Benward & Saker, 2003). However, in order, to understand harmony, it is first necessary to grasp what melody is. Melody is a group of notes played one after the other, the tune that is often the easiest part of music to remember, the part that one may hum. Harmony is also a group of notes, except that these notes are played in the background, beneath and around the melody. The role of harmony role is to accompany the melody and is usually expressed as a sequence of different voices or instruments that play musical chords. A chord, in music, is any harmonic set of three or more notes that is heard as if sounding simultaneously (Benward & Saker, 2003). Chords are typically consisting of four voices ranging from a higher to lower pitch: Soprano, Alto, Tenor, and Bass. The assignment of musical chords in a given melody is called melodic harmonisation, which is the object of study for this chapter.

The task of melody harmonization incorporates the preservation of balanced relations between the melody and all chord-composing sequences. This is achieved by a set of musical "rules" which defines a certain music style, such as classical, rock, jazz etc. The analysis of harmony is normally performed manually by music experts; however with the advent of computers, research has investigated whether all these rules can be analyzed and simulated by computerized frameworks. The practice of abstracting rules of harmony and placing them in a linguistic framework has been a part of computer science at least since the 1960s (Winograd, 1968; Jackson, 1967). Automatic melodic harmonization is a natural extension of harmonic analysis, and an important component in music information research. Its function is to clarify principles used by composers and musicians, and to capture these rules in an artificial intelligence (AI) framework (Koops, 2012).

Automated melodic harmonization has so far been approached from two different angles: with either the purpose of finding a satisfactory chord sequence for a given melody (performed by the soprano voice) or with the purpose of finding the remaining three voices that complete the harmony for a given bass line (Figure 1). The four-part harmonization is a traditional part of the theoretical education of Western classical musicians and therefore numerous researchers have attempted to generate automatically the four-part harmonization.

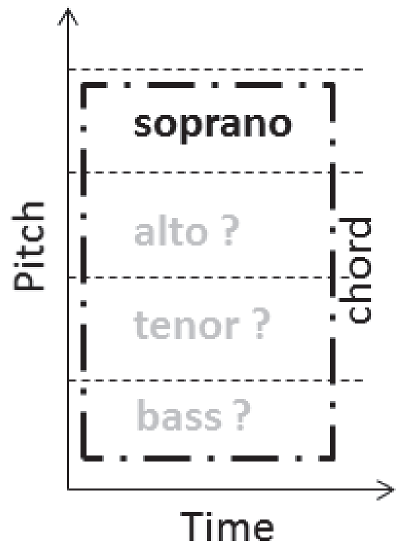

Figure 1: Automated melodic harmonization approach: finding the remaining three voices that complete the harmony for a given soprano line.

The task of automated melodic harmonization can be considered as a branch of algorithmic musical composition (Jacob, 1996) which is the application of a strict, well-defined artificial intelligent algorithm for the process of composing music. In the current context, music can be considered as a single instrument/voice or a combination of voices and instruments, as clearly shown by both the music industry and common practice. As far as the computational part of the harmonization is concerned, machine learning algorithms and techniques have been used widely in the field. Nevertheless, most existing methods use a context generic approach (HMM), which makes little use of domain specific information (Eddy, 1998).

The aim and key contribution of this chapter is to provide an introduction to the importance and the requirements of automated melodic harmonization research, as well as to present a concise literature review of the main conceptual approaches in this area. The chapter presents newly emerging research directions concerning idiom representations and discusses the need for further research which stimulates a number of open challenges in the field of automated melodic harmonization.

The rest of the chapter is organized as follows: The section on *Background* offers an introduction to automated melodic harmonization and justifies this area as an important research direction in music information research. The section on *Automated Melodic Harmonization* provides an overview of earlier automatic harmonization efforts and considers the methodological variations from a musicological point of view, detailing the numerous machine learning techniques that have been adopted. The section on

Integrated Systems for Harmonizing presents integrated musical systems contributing to automatic harmonization with a special interest on systems with a graphical interface. The section on *New Directions on Harmonizing* discusses new musical idiom representations, especially as far as chords are concerned, and, finally, the chapter on *Conclusions* offers a brief and concise summary of existing research directions in automatic melodic harmonization.

BACKGROUND

Music information research is a young multi-disciplinary field, which started developing after the turn of the century, and has since managed to provide a valuable insight into the interaction of computer science and musical domains. For example, Lavranos et al. (2015) provide a comprehensive and extensive survey of music information research and its application to musical creativity and in particular, creative activities such as composition, performance and improvisation, and listening and analysis.

Algorithmic Composition

Musical intelligence may be defined as the simulation of thinking and creativity during the musical content creation/composition process (Cope, 1992). When it comes to musical composition, creativity often arrives in sudden and unexpected bursts, which only the same creative artist can evaluate and take advantage of. Thus, the mental state of an artist, during the process, is more than crucial for the musical composition. On the other hand, it is not just the inspiration of the artist that is important, but also the ability to utilize it, a characteristic of professional musicians' compositions requiring extensive experience.

All in all, creativity mostly comes into two steps: Musical intelligence thinking (that is commonly referred to as a 'genius' element) and hard work. While the former may produce 'inspired' music - it is not fully understood, and therefore it is difficult to repeat; the latter can be more easily realizable as a computer program that attempts to construct a set of rules that help musicians learn the process of creating a musical piece (Jacob, 1996). The ability of such systems to specify a series of deterministic steps for later execution is the state of the art in algorithmic composition. Algorithmic composition is, thus, the application of a rigid, well-defined algorithm for the process of composing music.

An algorithm by definition is rigid, whereas creativity often breaks rules. This is why it is difficult to reproduce the musical intelligence of a human composer as they usually trade musical rules and bend them for creativity. It is thus common to consider algorithmic composition as a cheat when the composer is out of material. But it can also be thought of as a tool that simply makes the composer's work advance faster by helping them to learn how to mimic musical rules that then lead to truly performing the act of being creative in musical terms. Therefore the goal of algorithmic composition as it is described in Jacob (1996) is to reproduce the creativity and the methodological steps for a composer, when he/she is on a hard work mode.

Computers are not new to musical composition (Laske, 1981; Park, 2009). The notion of computer musical composition has been attributed to two different meanings that are complementary. The first revolves around computer-composed music that involves algorithmic composition (and is the subject of chapter), while the second is related to computer-realized music that involves conversions of musical scores or sounds to produce new electronic sounds (Hiller, 1981). The evolution of digital music is the object of study of the latter category and addresses the rise of electronic music with digital synthesizers and new types of sound syntheses, such as Physical Modeling Frequency Modulation. Generally, algorithmic composition is used for purposes such as generating a tonal melody, creating valid alternative chord progressions and suggesting possible harmonization over a melody.

First Attempts

For decades, computers have been used to compose music, typically via probabilistic or stochastic methods. Hiller (1981) proposed the creation of a computer-based framework used as a composing machine with facilities for composition algorithms, analog to digital conversions, sound synthesis algorithms and high level score language interpreters. He also claimed “that computer-assisted composition is difficult to define, difficult to limit, and difficult to systematize”. The publicity generated by Hiller and Isaacson's *Illiac Suite* (Hiller & Isaacson, 1979), a string quarter, attracted a number of individuals interested in merging new music with new technology. In their work, they utilized two basic approaches: a) random selection constrained by lists of rules and b) random Markov chains in which the relative likelihood of each option was conditioned by one or more immediately preceding choices.

In algorithmic music composition, the intention and the expectations are far from modest, even though some comprehensive systems, such us Experiments in Musical Intelligence (EMI) developed by Cope (1989), can construct complex and complete pieces. EMI is a project focused on the understanding of musical style and stylistic replication of various composers. As an input, EMI uses musical works of the composers and outputs some unique characteristics of the works by using computational techniques to separate aspects of music, such as style, which have traditionally been investigated qualitatively or with exhaustive manual effort. Figure 2 shows the block diagram of the underlying structure.

Xenakis (1992) is perhaps the best-known composer on algorithmic composition. He was the first to introduce statistical methods of composing for live ensembles: his work, *Pithoprakta,* exploits Gaussian distribution while *Achorripsis* uses Poisson's distribution of rare events to organize “parts” of sound. Tenney (1969), proposes the direct visual examination of the score. Note detection, statistics about periods between attacks, ranges of pitches, dynamics and several aspects of timbre are thus revealed. In addition to that, a significant feature for notating the rhythm is proposed.

Until the mid-1980s, much of the attention was focused on timbre, rather than the structure of the music composed. As computers got more powerful, handling increasingly complex tasks that required more computer resources became an option. Nowadays, not only computing resources are readily and widely available for researchers but also more complex mathematical models have been tested. Genetic algorithms for composing tasks have been proposed by Jacob (1995). Therein, a composition process is described that combines traditional stochastic methods to complex rule-based systems, such as EMI, in order to achieve the simplicity of a stochastic process and the determinism of a rule-based system.

Uses of Algorithmic Composition

As seen from the indicative initial attempts aforementioned, the applications of algorithmic composition can be quite diverse. Besides the notion of an entire system capable to compose a new musical piece from scratch, which is considered unachievable even today as far as reliable results are concerned, the process of composition can be divided into sub parts which are to be examined separately (Brown, 2007).

Applications of algorithmic processes are to enable the composer to specify music in a partial way. Traditionally, composing music has involved a series of activities, such as the definition of melody and rhythm, harmonization, writing counterpoint or voice-leading, arrangement or orchestration, and engraving (notation) (Fernández & Vico, 2013). Therefore, when the composer does not have the acquired skills to specify all the arrangement of the song, this can be seen as an extension of interpretation of users’ improvisation practices. In addition, ethical and intellectual property issues are taken into consideration. Except for the obvious belief that algorithmic composition is an overall cheat, music produced by algorithmic composition is considered somehow inferior, not because it has been produced by an algorithm, but because it belongs to the designer of the algorithm, not the user (Jacob, 1996).

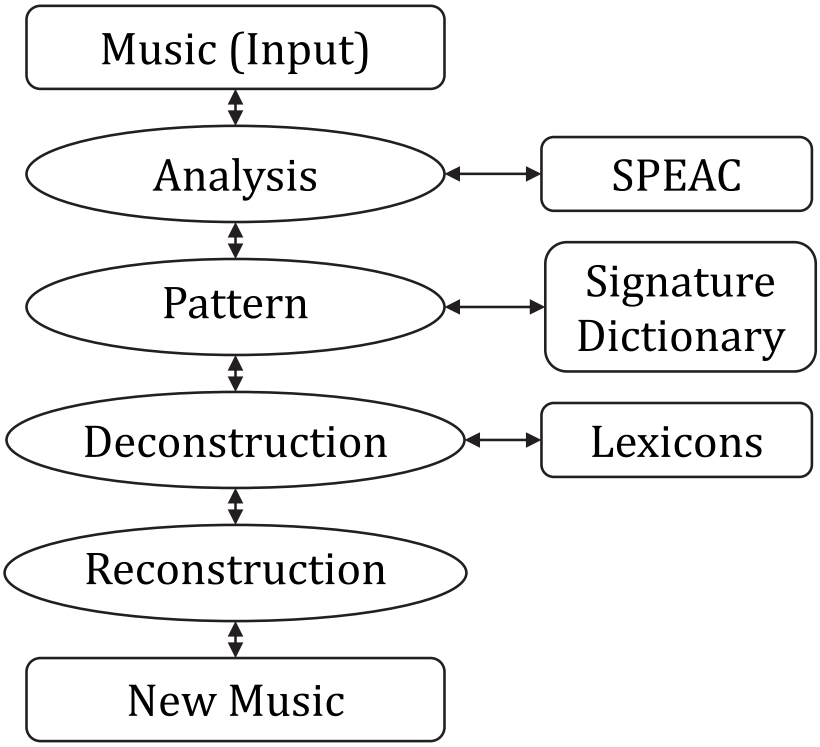

Figure 2: Diagram for the general algorithm of EMI (Cope, 1989).

However, algorithmic systems can be used to provide new ideas for the composer. The probabilistic nature of the systems generating alternative possibilities (or paths) motivates the users to modify their compositions or stimulate further compositional ideas (Englert et al., 1988; Hiller, 1981; Laske, 1981). From this perspective, because of their design, those systems can be considered part of the creative process and act like an autonomous musical instrument. In addition to being applied during the stages of composition in order to create variations or extensions to existing material, algorithmic systems can be designed to create early pre-compositional material.

AUTOMATED MELODIC HARMONIZATION

Melodic harmonization is the assignment of musical chords over a given melody. Automated melodic harmonization can be considered as an application of algorithmic composition. It deals with basic musical information (notes, pitches, intervals, scales etc.), provided by annotated scores. Its aim is not to build a musical piece from scratch but to represent new or alternative harmonies as they were written on musical scores. Probabilistic and stochastic algorithms are used to study the separate voices of harmony, trying to find and learn patterns or rules for given musical pieces. The goal in itself is not only to produce new harmonies (which is considered as a new trend, see Section 5) but to understand and create probabilistic systems with machine learning algorithms that are capable of simulating the harmonization process.

Statistics and information concerning the harmonic structures of each composer can be extracted and subsequently used for creating new harmonies or accompanying the creation process (see Section 4) or for combining different probabilistic systems to create new harmonies and structures (see Section 5). It is thus obvious that the use of melody harmonization is to provide new harmonies (pieces) and act like an autonomous process for creativity or accompaniment for the user creating new variations and pre-compositional material.

Chorale Harmonization

The task of automated melodic harmonization has been approached from two different perspectives: either as a means of finding a satisfactory chord sequence for a given melody (performed by the soprano voice), or as a means of identifying the remaining three voices that complete the harmony for a given melodic or bass line. The typical form in the latter type of harmonization is referred to as the "four-part harmony" task, which examines the proper combination of the soprano, alto, tenor, and bass voices.

The four-part harmonization (Chorale harmonization) is a traditional part of the theoretical education of Western classical musicians. Given a melody, the task is to create three further lines of music which will sound pleasant when played simultaneously with the original melody. A good chorale harmonization will show an understanding of the basic 'rules' of harmony which characterize the composer and his style. A plethora of researchers (Ebcioğlu, 1988; Horner & Ayers, 1995; Allan & Williams, 2005; Yi & Goldsmith, 2007) have dealt with this task which covers all the aspects of melodic harmonization systems. The most important will be reviewed below.

Types of Automated Harmonization Systems

Research into computational analysis of harmony has a history of over four decades. Papadopoulos and Wiggins (1999), in a survey of algorithmic composition, review some representative examples of systems which employ different AI techniques, categorized based on their most prominent features, as follows:

- Mathematical Models
- Knowledge Based Systems
- Grammars
- Evolutionary Methods
- Learning Systems
- Hybrid Systems

Since melodic harmonization is an application of algorithmic composition, herein, all efforts focusing on harmonizing given melodies, rather than composing entire pieces of music are addressed. Some key mathematical models featuring stochastic processes and techniques have already been presented earlier in this chapter as the first attempts of algorithmic composition (Section 1). As far as Knowledge Based Systems are concerned, these refer to AI systems which use symbolic information as well as rules and constraints. These are better known as Rule-based systems and have received a lot of attention prior to Machine Learning techniques. A grammar representation of music is a symbolic representation, referring to a generic class of composition by means of a list of grammatical production rules, or by means of a "parse tree" which graphically depicts the syntactic structure of a composition (Roads & Wieneke, 1979). Evolutionary Methods usually refer to Genetic Algorithms which have been shown to be very efficient in searching methods especially when dealing with very large search space problems (Goldberg, 1981). Learning Systems are systems which, in general, do not have a priori knowledge but instead learn features by examples from datasets. Finally, Hybrid Systems refer to the systems using combinations of the above AI techniques.

However, the focus of this chapter lies on Learning Systems which are the most modern approach and widely used technique. Special mention will be given below to Markov models, which are of particular interest due to their simplicity of operation, although some machine learning algorithms of hybrid nature will also be presented. In addition, extensive references are also provided for interested readers for most of the early attempts.

Grammars, Rule-based & Genetic Algorithms

Due the number and complexity of musical harmony rules, the solution of creating grammars has received wide acceptance by researchers. One of the first grammars was proposed by Winograd (1968) where the user was required to manually remove any non-harmonic notes (passing notes, suspensions and ornaments) before the algorithm processed the remaining chord sequence. Steedman (1984) devised a set of chord substitution rules, in the form of a context-free grammar, for chord progressions in jazz twelve-bar blues. Again, some years later, Steedman (1996) extended the previous work by using categorial grammars, by putting into the lexicon most of the information that is standardly captured in context-free phrase-structure rules. It should be noted that a flexible categorical grammar typically simulates more closely the listener's perception and interpretation of the chord progressions (Steedman, 1993).

However, music grammars solutions suffer mainly from ambiguity problems, making parsing symbolic information, in some cases, very expensive. In addition, the capability of such solutions to generate large number of musical strings is of a questionable quality, and thus an important issue too. Closely related to grammar-based approaches are rule-based approaches, which were used widely in early artificial intelligence systems. An early attempt of Higgins & Steedman (1971) used an elimination process combined with heuristic rules in order to infer the tonality given a fugue melody from Bach's Well-Tempered Clavier. Two "parsing" solutions were proposed: one to determine metrical units and another one to determine harmonic relationships between notes. The results were evaluated as promising for the future.

Another notable effort was achieved by Ebcioglu (1988) who designed an expert system called CHORAL, for harmonizing four-part chorales in the style of J.S. Bach. The proposal contains 270 rules, which were found from empirical observation of Bach Chorales, for representing the knowledge required for harmonizing a given melody. The CHORAL system was written in BSL, a logic program language, and used back track-able process techniques to implement multiple viewpoints like the chord-skeleton (which produces one chord per step), individual melodic lines of each voice, and a voice leading module (which is used for the bass and descant melody lines of the chorale separately).The results demonstrated that tonal music of some competence can indeed be produced through the rule-based approach.

Genetic Algorithms (GAs) are considered promising for music composition as these combine a form of 'creativity' (the ability to explore a large search space) with constraints (creativity is controlled by the fitness function). Two cases are examined where the fitness function is used and evaluated either as an objective computable function or by a human. Papadopoulos and Wiggins (1998) used GAs with problem dependent genetic operators, variable length chromosomes and a fitness function which evaluated eight different characteristics of the melody (intervals, note durations, contour etc.) in order to evolve jazz melodies based on a given chord progression.

Towsey, et al. (2001) addressed the difficulty to define fitness functions which capture the aesthetic qualities of the wide range of successful melodies. They described 21 melodic features used as the basis for a GA fitness function and for mutation procedures for 36 melodies. In their results claimed that, was not possible to draw strong conclusions from the cluster analysis of their dataset.

Horner & Ayers (1995) introduced a genetic algorithm method for harmonizing four-part complex musical progressions. They addressed a problem that has fairly strict constrains, the voice leading between the voices. Therein, the harmonization problem was separated into the sub problems of finding all possible chord progressions and then identifying chords satisfying the voice leading problem. Finding the set of individual chords proved to be easy since there were few constraints. Subsequently, they used GAs based to natural selection, to solve the chord sequencing problem. Their work was successful, but on a very constrained problem, because all the chords were given as an input.

Phon-Amnuaisuk & Wiggins (1999) reported a comparison between Genetic Algorithms (GAs) and a rule-based system using four-part harmonization of chorale melodies. The results indicated better performance for the rule-based system. However the conclusion from this experiment was that the quality of the output of any system is fundamentally dependent on the overall knowledge that the system (explicitly and implicitly) possesses.

Markov models on Automated Melodic Harmonization

Markov models are widely used on melodic harmonization. There are three commonly used Markov models in different situations of harmonization, depending on whether every sequential state is observable or not, and whether the system is to be adjusted on the basis of observations made: Markov Chain, Markov Decision Process and Hidden Markov Models. In the sequel, the most important research for harmonization using Markov models and their variations are presented.

Hidden Markov Models (HMMs) are statistical Markov models in which the system that is being modeled is assumed to be a Markov process with unobserved (hidden) states (Eddy, 1998). In simpler Markov models it is assumed that future states depend only on the present state and not on the sequence of events that have preceded it (Markov assumption), and therefore the state transition probabilities are the only parameters. In a HMM model, the state is not directly visible, in contrast to the output which is dependent on the state, and thus visible.

According to the HMM methodology, a sequence of observed elements is given and a sequence of (hidden) states is produced as output. The training process of an HMM incorporates the extraction of statistics about the probabilities that a certain state follows another state, given the current observation element. Concerning the training process, statistics are extracted from a training dataset that incorporate four aspects:

- The probability for each state to be a beginning state.
- The probability for each state to be an ending state.
- The probability that each state follows another state.
- The probability of a state being present over an observation.

Hidden Markov Models are generative models and widely used on melodic harmonization. Their formalization describes the targeted task very well: given a sequence of observed notes (melody), find the most probable (hidden) sequence of chords that is compatible with the observations. Figure 3 shows a diagram of a HMM for melodic harmonization purposes.

Raphael and Stoddard (2003) proposed a probabilistic approach to functional harmonic analysis, using a hidden Markov model (HMM). Since the model was based on rhythm and pitch for training data they used a collection of MIDI files as a dataset. The harmonic analysis was performed on a fixed musical period and their system outputted the current key and the scale degree of the current chord. In order to make the computation tractable, a number of simplifying assumptions were made, such as the symmetry of all musical keys, etc. Although this reduced the number of parameters, the training algorithm was only successful on a subset of the parameters.

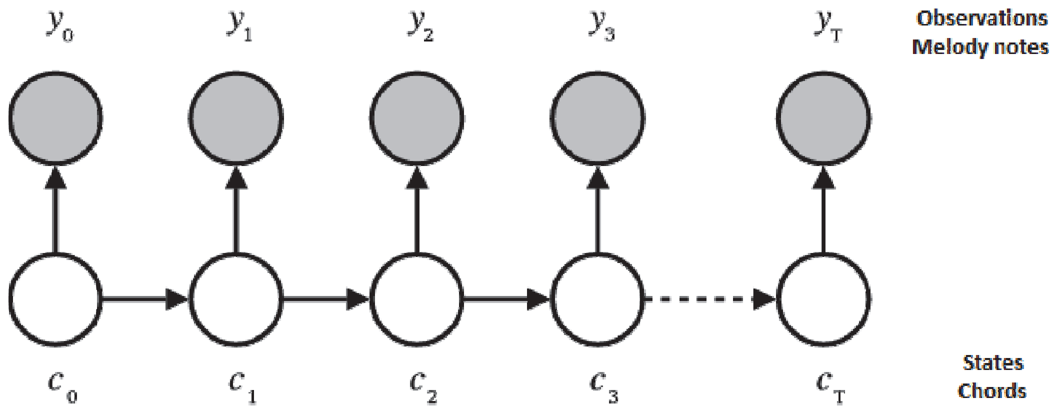

Figure 3: An HMM diagram harmonizing a given melody

Allan and Williams (2005) proposed a four-part harmonization method based on HMMs. Therein, two HMMs were utilized to generate chorales in the style of J.S. Bach. The first HMM was employed to yield a sequence of note intervals that accompanied each melody beat, while the second produced finer-scale ornamentations. In the Hidden Markov Models used there, the ‘hidden’ states of chords and harmonic symbols were in fact visible in the data during training. That means that they learned transition and emission probabilities directly from observations using their training data set of harmonizations. Their motivation was to create a model which can be used for the prediction of notes for filling three voices corresponding to the remaining harmonic lines at each time step. This was made possible using the Viterbi algorithm (Forney, 1973), which, given a new melody line, identifies the most likely state sequence, and thus a harmonization.

Hanlon & Ledlie (2002) presented an automated system for the harmonization of four part chorales in the style of J.S. Bach called CPU Bach. Their solution divided the problem at hand into two subtasks. First, they created a chord sequence in which the melody was consonant and, secondly, they produced that harmony by filling in the other voice parts to be consistent with that progression. The first part computed the several chords to which the note of a given melody might belong and then it selected one sequence of chords by transitioning among these according to the HMM structure. The second module contained musical rules and solved the constraint satisfaction problem to assign actual notes, and therefore melody, to the voice parts. The output of the CPU Bach system was considered to be qualitatively quite good.

Markov processes are based on the “Markov hypothesis” which states that the future state of a sequence depends only on the last state. Simple Markov Models are often estimated by counting occurrences and transitions in a corpus of training sequences. Once the model is learned, sequences can be generated simply by random walk. Markov Decision Processes (MDPs) are thus considered to model stochastic systems. In this case, chord progressions can be viewed as a stochastic process, while the choice of a chord is similar to the choice of an action in MDP planning.

Eigenfeldt & Pasquier (2010) presented a Markov model based method for generating harmonic progressions using a case-based analysis of existing material. The case-based system described by Spector & Alpern (1996) is able to generate Markov conditional probability distributions, using either first, second, or third-order chains. Therein, the user specifies a three-dimensional vector suggesting bass-line movement, harmonic complexity, and voice-leading tension in order to stochastically choose from the best matching solutions. Their proposed system is motivated by the requirement of offering a balance between user requested material and coherence within the database.

Finally, Yi & Goldsmith (2007) introduced decision-theoretic planning techniques for automatic music generation. They used Markov Decision Processes for chord progressions. A state was represented as a 10-tuple (S1, A1, T1, B1, S2, A2, T2, B2, S3, P), where Si, Ai, Ti, Bi were respectively the soprano, alto, tenor, and bass notes at time i, and P was a temporal position. Figure 4 shows an example of a single state. The results showed that the harmonies that were produced were not very sophisticated but that it was possible to apply decision theoretic planning techniques to automate music generation.

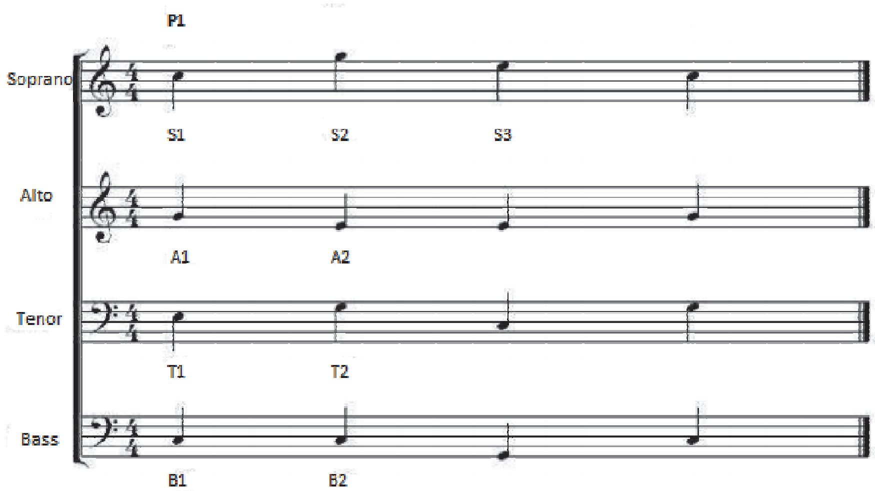

Figure 4: A single state represented as a 10-tuple using Markov Decision Processes (Yi & Goldsmith, 2007).

Hybrid Approaches and other Machine Learning Algorithms

Most systems use the generative Hidden Markov Model (HMM), in which the chords are the hidden states and the melody is the observed output. Relations to other variables, such as the tonality and scale or the metric structure are usually ignored (Chuan & Chew, 2007; Raczyński et al., 2013). These approaches have been reported to fail in capturing essential aspects of the high-level musical structure and context, and thus limiting their usefulness, particularly for musically informed users. In the sequel, some new machine learning algorithms (Dynamic Bayes Networks, SVMs, Neural Networks) and some hybrid approaches, combining rule based systems with probabilistic or multiple probabilistic systems, are presented.

Raczynski, et al. (2013) proposed a flexible way of developing discriminative probabilistic harmonization models, in which the time-varying tonality, as well as other musical variables can be explicitly taken into account. This was done by joining multiple simpler sub-models by means of linear or log-linear interpolation. To test their system, they trained and combined in this way the three sub-models that were created: the tonality, the melody and the chord bigram model. The evaluation was successful in terms of cross-entropy. They observed that log-linear interpolation yielded a model the cross-entropy of which was lower than the best of the component models and, also, better than the one achieved by linear interpolation (Jurafsky & Martin, 2014).

Tree structure approaches were also examined by. Paiement et al. (2005) who presented a graphical model that captures the chord structures in a given musical style using as evidence a limited amount of symbolic MIDI data. Every chord in a chord progression depends both on its position (global dependencies) in the chord structure as well as on the surrounding chords (local dependencies). Parameters in the graphical models were learnt with the EM algorithm and the classical Junction Tree algorithm was used for inference. They showed that chord progressions exhibited global dependencies that could be better captured with a tree structure related to the meter than with a simple dynamical HMM that concentrated on local dependencies.

Chuan & Chew (2007) proposed a hybrid system for generating style-specific accompaniment from a given melody in 3 steps. The first step concerned the determination of chord tones through utilizing Support Vector Machines (SVMs) while at a next step the system determined which notes in a given melody needed to be harmonized. According to these notes, triads were assigned, first at checkpoints (where the bars with all the possible chord solutions were available). The third step was the construction of possible chord progressions using neo-Riemanian transforms.

The utilization of neural networks has also received attention. Hild, et al. (1992) utilized three kinds of neural networks. The first one generated harmonic tree structures from a soprano melody, the second one allocated concrete notes from these skeletons, while the third one was used for ornamentation. Cunha & Ramalho (1999) proposed a hybrid model for chord prediction showing that combining a neural network with a rule-based sequence tracker, improved the system's performance. They defined their problem at hand as one that neural networks cannot adapt the unique characteristics for harmonizing a particular idiom, since it is impossible to have prior knowledge on neural networks. Their reported results were very promising.

Suzuki and Kitahara (2013) developed two kinds of computational models, one that contained chord nodes (in the Bayesian networks) and another that did not. Both were capable of generating four-part harmonies using Bayesian networks. They wanted to investigate to what extent the model without chord nodes affected the harmonization in terms of voice leading compared to the model with chord nodes. Another approach, but this time with Dynamic Bayesian Networks (DBN), was proposed by Dixon, et al. (2005). DBNs are graphical models representing a succession of simple Bayesian networks in time. These are assumed to be Markovian and time-invariant, so the models can be expressed recursively in two time slices: the initial slice and the recursive slice. They proposed two approaches for modeling musical harmony: using a probabilistic and a logic-based framework, respectively. The first was a chord transcription system which used a high-level model of musical context (chord, key, metrical position, bass note, chroma features and repetition structure), integrated in a Bayesian framework. The second approach used inductive logic programming to learn logical descriptions of harmonic sequences which characterized particular styles or genres. They showed that the combination of high-level harmony features with low-level features (audio features) could lead to harmonization accuracy improvements as well as to genre classification.

INTEGRATED SYSTEMS FOR HARMONIZING

So far, autonomous probabilistic systems for harmonizing given melodies by creating chord progressions have been examined. In the context of this work, we can divide these approaches into two categories: interactive and imitation systems. Imitation systems aim at representing stylistic information, i.e. to generate music in various styles, but they require symbolic information (transcripted musical scores) with manual human involvement and of course setting supervised learning algorithms. Most works with four-part harmonization, that have been examined so far, belong in this category.

Interactive systems, on the other hand, have been popular both in the research field as well as in commercial applications. Both share the common drawback of not being able to learn. Consequently, the music generated is strongly correlated to the musical input. This section addresses integrated musical systems contributing to automatic harmonization with combinations of imitation and interactive systems. Interactive systems which are irrespective of a graphical interface but are nevertheless intended for creating music by accompanying a voice (due to the ease of interactivity required by potential users) are also presented.

The Continuator (Pachet, 2003) bridges the gap between two classes of traditionally incompatible musical systems, interactive musical systems and music imitation systems. Therein, the authors proposed a system in which musical styles are learned automatically in an agnostic manner allowing musicians to extend their technical ability with stylistically consistent, automatically learnt material. A MIDI system linked to an arbitrary MIDI controller was used to send the notes to a synthesizer which generated the instruments' sounds. In reaction to the played musical phrase, the system generated a new phrase, built as a continuation of the input phrase, according to the database of patterns already learnt. An efficient implementation of a complete variable-order Markov model of input sequences was designed by building a prefix tree. Then the generation was performed using a traversal of the trees built from input sequences.

Hyperscore (Farbood et al., 2004) is a graphical computer-assisted composition system for novice composers. The Hyperscore software tool addresses the combination of the development of musical algorithms for automating the compositional process and the design of an appropriate interface for humans to interact with the machine. It allows users to construct musical "motifs" out of notes; the volume and pitch of these motifs could then be manipulated using a graphical sketchpad. Finally, it should be noted that the Hyperscore is suited for use as both an educational tool and as a way to explore musical creativity.

MySong (Simon et al., 2008) is a system that automatically chooses chords to accompany a vocal melody. It trains a HMM using a music database and uses that model to select chords for new melodies. Model parameters are intuitively exposed to the user. Creating music with MySong requires the user to record a vocal melody singing along with a computer-generated beat at a user specified tempo. Then the system transcripts the melody and chooses the best chord sequence, over 62 available unique chords from training data, by use of the Viterbi algorithm. Figure 5 shows the graphical user interface of MySong. The evaluation was done by means of comparing it with a commercially-available system for generating chord sequences, "Band-in-a-Box" (BIAB) (PG Music Inc, n.d.), which is primarily a system for generating accompaniment audio from chords, but includes a module for determining chords from a melody. All user-evaluation participants gave positive subjective ratings to MySong, and indicated an interest in the continued use of MySong.

Finally, interaction through the user can be also based in emotions. Robin (Morreale et al., 2013) is an algorithmic composer specifically designed for interactive situations based on Valence & Arousal. It adopts a rule-based approach to compose original tonal music in classical piano style. Users can direct the composition in real time, conveying emotions that are translated into matching music in classical piano style. Its Harmony Generation Module is controlled by a first-order Markov process which states that chords correlation does not depend on previous states of the system. Robin's evaluation was done with user-questionnaires and indicated that most of the users greatly appreciated the installation and in particular the quality of the music.

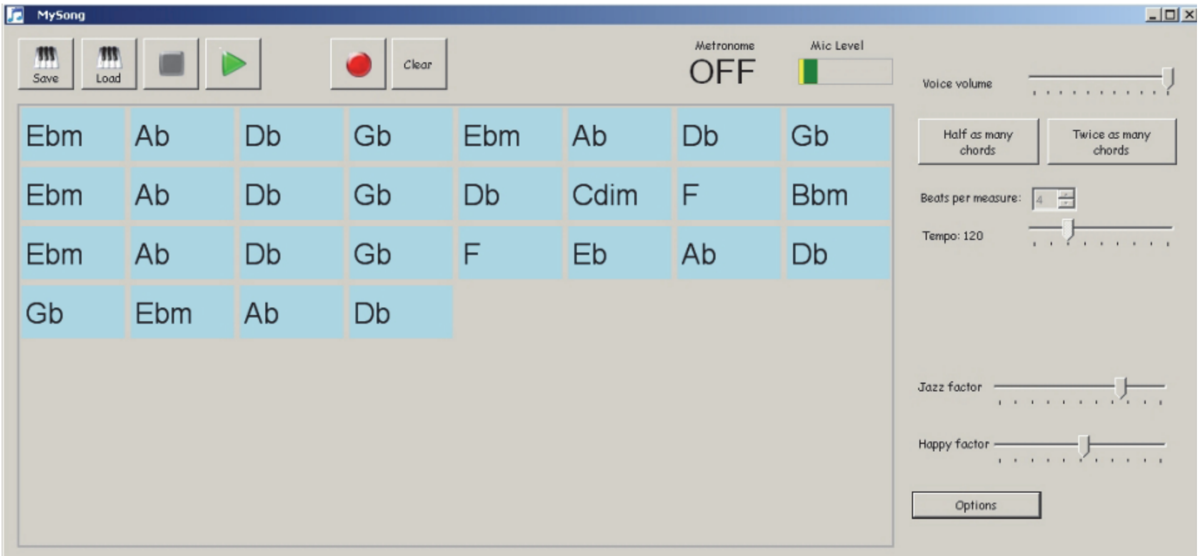

Figure 5: MySong's interface. A user can create an accompaniment using the record and playback controls and change the harmonization style by adjusting the "jazz factor" and "happy factor" sliders (Simon et al., 2008).

NEW DIRECTIONS ON HARMONIZING

So far our work has examined many melodic harmonization systems which aim to produce harmonizations of melodies that reflect the style of the input training examples (idiom). This is commonly pursued by utilizing chords and chord annotations that are characteristic of the idiom. Most of the examined research has focused on four-part harmonization (chorale). However, different idioms may present different representations. For example the chord representation for studies in the Bach chorales usually includes standard Roman numeral symbols, while jazz approaches encompass additional information. In addition, when designing chord detection algorithms, the lack of annotated databases makes evaluation and comparison of results difficult.

Harte et al. (2005) address that very same problem of creating a music idiom that is independent of symbolic representations. They proposed a text representation for musical chord symbols that was simple to write and understand by musically trained researchers, yet highly structured and unambiguous to parse by computer programs. In that work, the authors verify that hand labeling of chords in music files is a long and arduous task and there is no standard annotation methodology, which causes difficulties sharing with existing annotations. Their proposed representation has a tree structure form and it is provided in an extendible format for future additions.

Cambouropoulos et al. (2014) proposed a probabilistic new idiom independent representation for chords which is appropriate for encoding tone simultaneities in any harmonic context. The General Chord Type (GCT) representation allows the re-arrangement of the notes of a harmonic simultaneity and is adaptable to diverse harmonic idioms. At the heart of the GCT representation is the idea that the 'base' of a note, simultaneity should be consonant. Thus, the GCT algorithm tries to find a maximal subset that is consonant. The algorithm gets for input a consonance vector, a Pitch Scale Hierarchy and the input chord (in terms of MIDI pitch numbers), and tries to give the voice intervals and types of the possible chords describing the simultaneity.

Systems based on only one method have been shown to be less effective (Raczyński et al., 2013). This is true as it is quite difficult to embody all the musical rules that are required to do a proper harmonization. In addition, grouping musical rules in different systems is something which is also currently examined. For example, Makris et al. (2015) proposed a probabilistic approach for the automatic generation of voice leading for the bass note on a set of given chords. The proposed bass voice leading probabilistic model is based on the HMM and determines the bass voice contour by observing the contour of the melodic line. The motivation for developing a separate model for voice leading addresses the requirement of performing conceptual blending on melodic harmonization, by using chord progressions from a musical idiom and a characteristic voice leading features from another.

Finally, except for creating new idiom representations and examining the rules of harmony separately for better results, research on the traditional four-part harmonization is still ongoing with new variations. An example is the work of Buys & Merwe (2012) who proposed chorale harmonization with Weighted Finite-state Transducers. Their framework makes possible to place domain-specific regular constraints on the generated sequences. The process of the harmonization is thus achieved in different steps, similar to those followed by human composers, in a full probabilistic setting. Each step is performed by inference through a weighted finite state transducer cascade. The evaluation was measured through entropy, with promising results in comparison to existing approaches.

Another example is the work of Kaliakatsos–Papakostas & Cambouropoulos (2014), who proposed the hidden Markov model (HMM), in combination with additional constraints that incorporate intermediate fixed–chord constraints for melodic harmonization. Harmonization with fixed checkpoints is considered a crucial amelioration of previously used methods, since it enables the prior definition of important chords in intermediate positions of the melody to be harmonized (Chuan & Chew, 2007). These intermediate chords could either be specified by an algorithmic process that determines music structure on a higher hierarchical level, or directly inserted by a human annotator. Figure 6 shows an example of harmonization with constraints. The evaluation of the algorithm utilized a comparison between the proposed model and a typical HMM model which indicated completely different harmonizations between the two models.

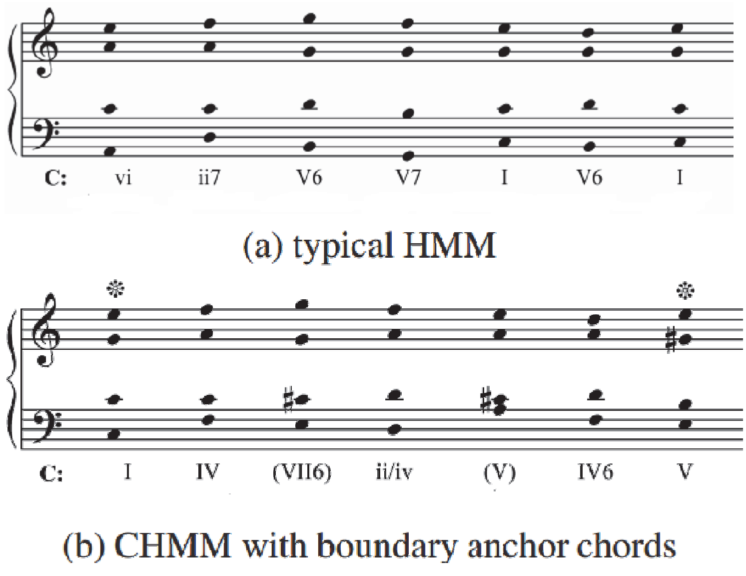

Figure 6: (a) The harmonization of a Bach chorale melody with the typical HMM methodology and (b) with constraints on the first and final chords (indicated with an asterisk). (Kaliakatsos–Papakostas & Cambouropoulos, 2014)

CONCLUSIONS

This chapter examined the task of automatic melodic harmonization which tackles the assignment of harmony (musical chords) over a given melody. Automated melodic harmonization can be considered as a subtask of algorithmic composition. Algorithmic musical composition is the application of a strict, well-defined artificial intelligent algorithm for the process of composing music. An overview of the reasons that led us to the need of algorithmic composition have been provided in Section 1 in addition to the corresponding uses on applications and the first early attempts in this area. However, the aim of melodic harmonization aim is not to build a musical piece from scratch but to represent new or alternative harmonies.

This harmony is usually expressed as a sequence of musical chords that typically consist of four voices, known as four-part harmonization which, in turn, refers to chorale harmonization. Although different types of harmonization systems were reviewed, the current chapter focused on Machine Learning Systems which are the state-of-the-art and widely used techniques. Special mention was given to Markov models which are of particular interest due to their simplicity of operation and widely used on chorale harmonizations. One of the reasons that chorale harmonization is so popular, excluding any musical analysis, is the availability of transcripted musical data in digital form and ready for parsing with machine learning algorithms. The lack of available datasets affects and restricts the researches in developing other musical directions as the transcriptions of musical sheets in a symbolic digital form are usually done manually by music experts, thus disturbing the process.

The above discussion was followed by an overview of, integrated musical systems contributing to automatic harmonization. Interactive systems with the ability to learn and with graphical interfaces have been popular both in the research field as well as in commercial applications. These are intended for creating music by means of accompanying a voice and, due to the ease of interactivity required by potential users, they make a positive contribution to both users and composers. In addition, it is important to note that, all the systems that were presented in section 2 were imitating the stylistic information of the harmonization from the training data. Integrated systems bridge this gap, combining interactive and imitating musical systems.

Finally, the chapter addressed new musical idiom representations, as far as chords are concerned. Latest approaches in chord encoding representation have been shown to allow the development of generic harmonic systems that may be adaptable to diverse harmonic idioms. In addition, new trends of examining and including separate musical rules on the traditional four-part harmonization were reviewed. This discussion, overall, demonstrated the main approaches that can be used for the construction of computationally feasible models for conceptual blending using harmonic content from different music idioms.

REFERENCES

Benward, B., & Saker, M. N. (2003). *Music in Theory and Practice. vol. 1*. McGraw-Hill Higher Education.

Brown, A. R. (2007). *Computers in music education: Amplifying musicality*. Routledge.

Buys, J., & van der Merwe, B. (2012). Chorale harmonization with weighted finite-state transducers. In *Twenty-Third Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)* (pp. 95-101).

Cambouropoulos, E., Kaliakatsos-Papakostas, M., & Tsougras, C. (2014). An idiom-independent representation of chords for computational music analysis and generation. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)*.

Chuan, C. H., & Chew, E. (2007, June). A hybrid system for automatic generation of style-specific accompaniment. In *4th Intl Joint Workshop on Computational Creativity*.

Chuan, C. H., & Chew, E. (2007, June). A hybrid system for automatic generation of style-specific accompaniment. In *4th Intl Joint Workshop on Computational Creativity*.
Cope, D. (1989). Experiments in musical intelligence (EMI): Non-linear linguistic-based composition. *Journal of New Music Research*, *18*(1-2), 117-139.

Cope, D. (1992). Computer modeling of musical intelligence in EMI. *Computer Music Journal*, 69-83.

Cunha, U. S., & Ramalho, G. (1999). An intelligent hybrid model for chord prediction. *Organised Sound*, *4*(02), 115-119.

Dixon, S., Mauch, M., & Anglade, A. (2011). Probabilistic and logic-based modelling of harmony. In *Exploring Music Contents* (pp. 1-19). Springer Berlin Heidelberg.

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

Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics*, *14*(9), 755-763.

Eigenfeldt, A., & Pasquier, P. (2010). Realtime generation of harmonic progressions using controlled Markov selection. In *Proceedings of ICCC-X-Computational Creativity Conference*.

Englert, G., Laske, O., & Roads, C. (1981). Automated composition and composed automation. *Computer Music Journal*, 30-35.

Farbood, M. M., Pasztor, E., & Jennings, K. (2004). Hyperscore: a graphical sketchpad for novice composers. *Computer Graphics and Applications, IEEE*, *24*(1), 50-54.

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

Forney Jr, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, *61*(3), 268-278.

Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. *Addison Wesley*, *1989*.

Hanlon, M., & Ledlie, T. (2002). Cpu bach: An automatic chorale harmonization system.

Harte, C., Sandler, M. B., Abdallah, S. A., & Gómez, E. (2005, September). Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations. In *ISMIR* (Vol. 5, pp. 66-71).

Hild, H., Feulner, J., & Menzel, W. (1992). HARMONET: A neural net for harmonizing chorales in the style of JS Bach. In *Advances in Neural Information Processing Systems* (pp. 267-274).

Hiller, L. (1981). Composing with computers: A progress report. *Computer Music Journal*, 7-21.

Hiller, L. A., & Isaacson, L. M. (1979). *Experimental Music; Composition with an electronic computer*. Greenwood Publishing Group Inc.

Horner, A., & Ayers, L. (1995). Harmonization of musical progressions with genetic algorithms.

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

Jackson, R. (1967). The computer as a'student'of harmony. In *Tenth Congress of the International Musicological Society*.

Jacob, B. (1995). Composing with genetic algorithms.

Jacob, B. L. (1996). Algorithmic composition as a model of creativity. *Organised Sound*, *1*(03), 157-165.

Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing*. Pearson

Kaliakatsos-Papakostas, M., & Cambouropoulos, E. (2014). Probabilistic harmonisation with fixed intermediate chord constraints. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)*.

Koops, H. V. (2012). A model based approach to automatic harmonization of a melody.

Laske, O. (1981). Composition theory in Koenig's project one and project two.*Computer Music Journal*, 54-65.

Lavranos, C., Kostagiolas, P., Korfiatis, N., & Papadatos, J. (2015). Information seeking for musical creativity: A systematic literature review. Journal of the Association for Information Science and Technology.

Longuet-Higgins, H. C., & Steedman, M. J. (1971). On interpreting bach. *Machine intelligence*, *6*, 221-241.

Makris, D., Kaliakatsos-Papakostas, M., & Cambouropoulos, E. A probabilistic approach to determining bass voice leading in melodic harmonisation.

Morreale, F., Masu, R., & De Angeli, A. (2013). Robin: an algorithmic composer for interactive scenarios. *Proceedings of SMC*, *2013*, 10th.

Pachet, F. (2003). The continuator: Musical interaction with style. *Journal of New Music Research*, *32*(3), 333-341.

Paiement, J. F., Eck, D., & Bengio, S. (2005). A probabilistic model for chord progressions.

Papadopoulos, G., & Wiggins, G. (1998). A genetic algorithm for the generation of jazz melodies. *Proceedings of STEP*, *98*.

Papadopoulos, G., & Wiggins, G. (1999, April). AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposium on Musical Creativity* (pp. 110-117). Edinburgh, UK.

Park, T. H. (2009). An interview with max mathews. *Computer Music Journal*,*33*(3), 9-22.

PG Music Inc: Band-in-a-Box. http://www.pgmusic.com

Phon-Amnuaisuk, S., & Wiggins, G. (1999, April). The four-part harmonisation problem: a comparison between genetic algorithms and a rule-based system. In *Proceedings of the AISB'99 Symposium on Musical Creativity* (pp. 28-34).

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, October). Harmonic analysis with probabilistic graphical models. In *ISMIR*.

Roads, C., & Wieneke, P. (1979). Grammars as representations for music. *Computer Music Journal*, 48-55.

Simon, I., Morris, D., & Basu, S. (2008, April). MySong: automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 725-734). ACM.

Spector, L., & Alpern, A. (1994, October). Criticism, culture, and the automatic generation of artworks. In *AAAI* (pp. 3-8).

Steedman, M. (1993). Categorial grammar. *Lingua*, *90*(3), 221-258.

Steedman, M. (1996). The blues and the abstract truth: Music and mental models. *Mental models in cognitive science*, 305-318.

Steedman, M. J. (1984). A generative grammar for jazz chord sequences. *Music Perception*, 52-77.

Suzuki, S., Kitahara, T., & Univercity, N. (2013). Four-part harmonization using probabilistic models: Comparison of models with and without chord nodes. *Stockholm, Sweden*, 628-633.

Tenney, J. (1969). Computer music experiences, 1961–1964. *Electronic Music Reports*, *1*, 23-60.

Towsey, M. W., Brown, A. R., Wright, S. K., & Diederich, J. (2001). Towards melodic extension using genetic algorithms. *Educational Technology & Society*, *4*(2), 54-65.

Winograd, T. (1968). Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, 2-49.

Xenakis, I. (1992). *Formalized music: thought and mathematics in composition* (No. 6). Pendragon Press.

Yi, L., & Goldsmith, J. (2007). Automatic Generation of Four-part Harmony. In *BMA*.