# A System for Automatic Audio Harmonization
## *(Ein System für automatische Audio-Harmonisierung)*

*Noam Yogev*, Alexander Lerch ***

* Freie Universität Berlin, yogev@inf.fu-berlin.de
** zplane.development, lerch@zplane.de

## Abstract

A rule-based system for automatic melody harmonization is presented. It models the cognitive process a human arranger undergoes when confronted with the same task, namely: segmenting the melody into phrases, tagging melody notes with harmonic functions, establishing a palette of possible chords for each note, and finding the most agreeable voicing through these chords.

The system is designed to be embedded in an audio framework, which synthesizes a four-voiced audio output using pitch-shifting techniques.

Principles of classical counterpoint as well as common voice-leading conventions are utilized by the system. We shall outline the various phases of computation, describe the rules applied in each phase, and present perspectives regarding the stylistic flexibility suggested by the system's design.

## 1. Introduction

Automatic harmonization – the task of automatically complementing a given melody with one or more additional voices – has been studied frequently in the past. The increase in computational power enabled the commercial availability of software systems offering such functionality in recent years. The crucial criterion of systems for automatic harmonization is the musical quality of their results. A harmonization should enhance and underline tonal and formal characteristics of the melody and at the same time meet musical stylistic standards. The elements of harmonization, which are responsible for these characteristics, are the choice of chords comprised by the additional voices, as well as the transitions between the chords along the melody.

When approaching the task of harmonizing a given melody, human composers may rely on their formal education in the rules of harmony, and have certain training in musical style at their disposal. It is reasonable to assume that an automatic system confronted with the same task has to use the same knowledge to be successful.

Common conventions of harmony are generally defined in collections of rules, applicable to specific musical situations. These may be straightforwardly implemented as weighted constraints within the context of an algorithm. A more sophisticated strategy is required when trying to extract tonal and formal context out of a given melody. These aspects of music are perception oriented, and in order to successfully describe and utilize them a perceptual knowledge to their extent must be available to an automatic harmonization system.

The system presented here harmonizes a given melody with three additional voices, delivering a four-voice piece. It makes use of perception-oriented techniques in order to find phrases in the given monophonic melody. The functional-harmonic content of each phrase is then determined using heuristic and stochastic methods. This knowledge is in its turn used to find potential chords for the harmonization of each note. The best sequence of transitions between incarnations of these chords is then computed utilizing traditional rules of harmony and voice leading as evaluation functions, and is output as the fully harmonized piece.

The system is both flexible and expandable in regard to different musical styles, and poses opportunities for further development in the fields of music education and research.

In Section 2 we shall present a short overview of some existing solutions for the task of automatic harmonization, pointing out differences in their underlying concepts. A description of the presented system is given in Section 3, including the general concept, details concerning significant stages of the algorithm and an example of the system's results. An overview of the system's capabilities and perspectives on possible future work are given in Section 4.

## 2. Related Work

Various approaches for solving the problem of automatic harmonization have been published in the past, reflecting different conceptual views on the problem.

Pachet and Roy regard automatic harmonization as a pure constraint-solving problem (CSP) [1]. Working in two steps, their system first computes all possible chords for the harmonization of each melody note, and then traces the optimal chord sequence along the melody. Specially tailored object representation for notes and chords, which implicitly fulfill some of the constraints, help in keeping the overall number of constraints for the CSP small.

Ebcioğlu's system CHORAL uses CSP as well [2]. Different perspectives of the harmonization – such as the relations between consecutive chords or the relations between consecutive notes within the same voice – are regarded as complementary sub-problems. A subproblem is solved as a CSP, utilizing the available input and pre-computed data. When the amount of information becomes insufficient, the system continues to solve the general problem along the line of the next sub-problem. With the CHORAL system, Ebcioğlu successfully modeled J. S. Bach's style of choral composition using 350 constraints. The number of constraints and the complexity of the interrelations between the different perspectives are described by Pachet and Roy as a major drawback of this system [3].

Phon-Amnuaisuk and Wiggins describe the use of a genetic algorithm (GA) in the context of automatic harmonization [4]. They conclude that this approach cannot outperform a CSP based solution for the same problem without practically abandoning the principles of GA with regard to the amount of domain knowledge represented in the system, thus arguing in favor of information-rich strategies.

Two possibilities for the encapsulation of domain knowledge within a system for automatic harmonization can be observed in the GYMEL system proposed by Sabater et al. [5] and in the ASSA-system described by Chuan and Chew [6]. Both solutions reduce the input melody to a core group of significant notes within each bar, which are likely to belong to a harmonizing chord. This analysis is performed in GYMEL according to attributes of the melody notes, including the position of a note within the bar. ASSA utilizes for this task a Support Vector Machine trained with fully harmonized pieces of a certain style. In addition, a representation of the melody notes is used which includes the position of a note within the bar, its position within a phrase as well as the number of pitch classes within a bar. GYMEL then attempts to harmonize the melody according to matches with an internal case base, and turns to a CSP solving module if no matches are found. The progression of GYMEL is linear along the time line of the melody, whereas ASSA assigns additional chord tones first to the core groups and then to the rest of the melody notes. Possible chord progressions are computed using a decision tree with rules of voice leading ("Neo-Riemanian Transforms", [6, Page 60] as transition functions. Finally, a Markov model is used to determine the best progression.

Two examples for systems that rely on highly detailed representation of musical data are given by Biyikoglu in [7] and by Paiement et al. in [8]. Biyikoglu's proposed solution is provided with information about metre, harmonic rhythm and the location of phrase boundaries within the melody. All the possible chords for the harmonization of each phrase are then computed. The harmonization is generated as an emission sequence of a 2$^{nd}$-order Markov model, in which the possible chords together with information about the position in the phrase are the process states. Biyikoglu trained the Markov model with J. S. Bach's chorals.

Paiement et al. present with HARMONET a system that incorporates neural networks (NN) in two stages of its computation. The first NN is used in order to determine the harmonic function of each note of the input melody. With these harmonic functions available, actual chord notes as well as the best progressions of chords and voicings are computed, with rules of harmony and voice leading serving as constraints in a CSP. Embellishments in form of passing and transition notes are added in a final stage of the process utilizing a second NN. Paiement et al. trained both networks with highly detailed representations of J. S. Bach's chorals including information about the position of each melody note within the bar and within the phrase. The system's design allows training with different collections of musical works, thus offering a possibility for automatic harmonization in different musical styles.

# 3. Method

## 3.1. Approach

As mentioned above, the conception of the presented system has the traditional harmonization process as its guideline. A harmonization that is correct in regard to the traditional rules of harmony follows inevitably diatonic and harmonic constraints inferred by the melody notes in the context of the key of the melody. Further, it is conceivable that a harmonization that is perceived as musically good or convincing follows the harmonic and formal structures of the melody.

Insights on these information levels are available for a human composer or arranger due to hers or his musical education and knowledge in the rules and traditions of harmony and voice leading, as well as hers or his musical experience and aesthetic training. Providing an automatic harmonization system with comparable information concerning the input melody as well as with knowledge about the practices of harmonization may enable the simulation of the human workflow, and offer comprehensive possibilities for improvement and refinement of such a system.

Figure 1 gives an overview of the processing stages of the system.

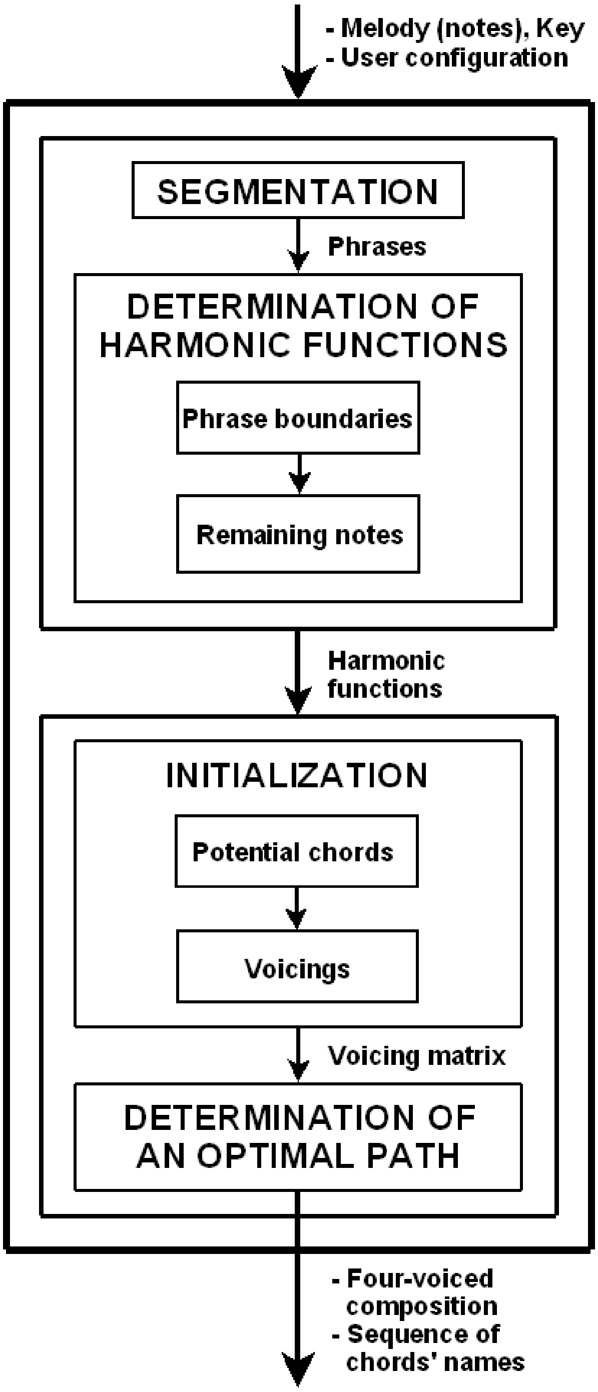

*Figure 1: System workflow*

The approach implemented here distinguishes between an analytical and a constructive stage in the harmonization process. During the analytical stage, phrases are detected in the melody, and their harmonic infrastructure is computed. During the constructive stage, possible chords and voicings are computed. An optimal sequence of voicings is determined, and returned to the user as the system's output.

Both processing stages will be described in detail in the following sections.

### 3.2. Analytical stage

Tenney and Polansky propose a method for perception-oriented segmentation of music according to principles of the Gestalt theory [9]. According to this method, segment boundaries are drawn at locations along a melody where local maxima are observed in the absolute value of a difference function. The difference function implemented in the system presented here takes attributes of the melody notes such as pitch differences and note durations as parameter. Owing to the hierarchic nature of musical phrases this operation is recursively repeated once – phrases detected by the system are on the second hierarchic level of segmentation. On the second level of the recursion the attributes for the difference function are computed for those sub-segments found in the first level. In addition, local differences at the locations of the boundaries are also taken into consideration.

In addition to Gestalt principles, the segmentation algorithm takes also pattern recurrences into account. Melody segments are compared with an equally long segment starting at the beginning of the melody. For this comparison a melody contour function is introduced, describing the tendency of change in a parameter over time. A pattern match is registered when both segments display the same contour in their pitch and duration values. The boundaries of the longest recurring patterns are marked as segment boundaries on the first segmentation level, and are more likely to achieve higher scores during the difference computation on the second level.

The harmonic analysis distinguishes between the following four main harmonic functions: *Tonic*, *Subdominant*, *Dominant*, and *Parallel Tonic*. Other, secondary harmonic functions as well as chords on scale degrees other than the 1st, the 4th, the 5th and the 6th are ascribed each to one of these main functions.

Each of the detected phrases is considered to represent a conclusive sequence of basic harmonic functions. Hence it is postulated that the end of each phrase represents a common cadence. Consequently, the melodic pattern at the end of each phrase (and at the very end of the melody) is compared with pre-defined template voice movements in various cadences, and the melody notes are tagged with the harmonic functions of the matching cadence.

The harmonic function of the first note of the melody is determined heuristically using a lookup table in which dependencies between scale degrees and harmonic functions are registered.

At this point, each phrase starts and ends with notes which were already assigned a harmonic function. Following our assumption in regard to the harmonic content of a phrase, the harmonic functions of the remaining notes – comprising the body of the phrase – describe a coherent harmonic sequence. This implies certain basic relationships between the melodic and the harmonic progress, in addition to the common relationships between harmonic functions within the sole context of harmonic progress. These relationships, expressed as transition probabilities between harmonic functions and occurrence probabilities of scale degrees – i.e. the degree of the melody note within the key – for any given harmonic function, were set heuristically, which in turn made the stochastic determination of the functional-harmonic progress possible.

A hidden Markov model (HMM) was implemented for this task; the harmonic functions represent process states while the scale degrees represent emission symbols. An adapted variant of the Viterbi-algorithm is used to determine the most probable hidden sequence of harmonic functions to the given sequence of melody notes along the phrase. The use of a $2^{nd}$-order HMM allows the consideration of more complex harmonic relationships.

At the end of the analytical stage each melody note is assigned a harmonic function..

### 3.3. Constructive stage

The system queries a data bank for potential chords for the harmonization of each melody note. The parameters for the query are the key of the melody, the note's scale degree and the harmonic function of the note. A list of chord descriptions is returned, each containing the queried scale degree, ascribed to the queried harmonic function, and with an initial weight depending on the probability of its occurrence in the context of this harmonic function.

A pool of all chord notes within the pre-defined voice range is established per chord. Permutations of three notes out of this pool are formed, and each is combined with the melody note. Such four-voiced combinations, created for all queried chords, form all the potential voicings of each melody note. These potential voicings of each melody note are stored along with this melody note.

The weight of each voicing, initially identical with the weight of its originating chord description, is now updated according to the following considerations:

1. **Chord tone doubling** – in order to exploit the full harmonic potential of each chord, the weight of triad-voicings in which more than one chord note is doubled is reduced, as well as the weight of four voice chord voicings in which a chord note is doubled.
2. **Chord inversion** – depending on the specific chord in the context of the harmonic function, some inversions may be encouraged while others may be discouraged.
3. **Voice Distance** – the user may configure the output harmonization to be in close or in open position. Voicings are encouraged or discouraged according to this configuration.
4. **Number of Voices** – favours four voiced chords or triads dependent on the configuration.

Whereas rule no. 1 is a classic harmonization rule and is generally independent of the musical style, the specific weight-modifications applied as penalties or as bonuses by rules Nos. 2, 4 and partly 3 may be used as parameters when adjusting the system to meet a certain harmonization style.

As the global goal is to find an optimal harmonization – i.e. an optimal sequence of voicing-transitions – all the optimal local solutions must be computed, following the principle of dynamic programming. In order to find the optimal local transition to a certain voicing from all its potential predecessors an optimization function is applied, consisting of a weighted combination of the following weight-modifying considerations:

1. **Voice crossing** – transitions with a voice crossing are rejected. This is the case when a voice crosses the melodic line of its neighbour.
2. **Voice Doubling** – sequences of voice doubling in consecutive voicings are discouraged.
3. **Voice Progression** – transitions displaying small melodic steps of the individual voices are favoured.
4. **Parallel movement** – is present when the interval between two voices in consecutive voicings remains constant. Depending on the interval such movement is either discouraged – as in the case of parallel fifths or parallel octaves – or favoured – as in the case of diatonic parallel thirds.
5. **Active Bass** – an "active" bass line is favoured to a "lazy" one. This rule encourages indirectly a livelier harmonization. A potential conflict with rule no. 3 has to be adjusted.
6. **Contrary motion** – encouragement of contrary directions in the voice progression between the bass and the soprano voice
7. **Pre-defined patterns** – specific cadence patterns and chord progressions are detected and monitored, allowing the discouragement of chord inversions that do not comply with traditional voice leading in these cases. For example, the weight of a voicing of a Neapolitan triad will be reduced if it does not realize the first inversion of the chord.

In a manner similar to the initial weighting of the voicings discussed above, rules Nos. 1, 2, 3 and partly 4 are straightforward implementations of traditional voice leading principles and are generally independent of the musical style, whereas the favouring of certain parallel movements in rule 4, as well as the rules 5 - 7 are used to realize a certain musical style.

The weight of each voicing is updated with the combined weight of its best predecessor and the weight of the transition.

After determining the best predecessor for each voicing at each time point along the voicing matrix, the optimal path through the matrix is backtracked, beginning with a voicing of the last melody note with the highest weight.

The voicings along the optimal path comprise the complete harmonization of the input melody. The individual voices, together with a representation of the chords realized by the voicings, make up the output of the harmonization system.

## 3.4. Results

The system output has been verified to comply with all the implemented rules of harmony and voice leading. A systematic musical evaluation of the system's results is yet to be carried out; however, informal musical appreciation of delivered test-harmonizations is satisfactory.

An impression of the musical quality of the delivered results may be gained through their musical analysis in relation to comparable harmonizations. The system presented here is a prototype for a new version of the audio plugin *vielklang*[1] which already offers similar basic functionality. We may therefore demonstrate the musical advantages of the system presented

---

[1] http://vielklang.zplane.de

here in comparison with the system it is designed to replace. A comparison may also be made with an example of the results of the harmonization system described by Allan and Williams [10], which works on a very similar representation of the input melody.

Figure 2 shows the harmonization of the melody of J. S. Bach's choral BWV 438 as delivered by these three systems. It also includes a harmonic-functional analysis and chord names.

While the choice of chords made by the old *vielklang*-version in this case is extremely conservative – the system uses only chords of the 1st, the 4th and the 5th scale degrees in different inversions – a wide variety of chords can be observed in the solutions of Allan's and Williams' system and of the system presented here. Allan and Williams' system demonstrates a fairly melodic bass-line in comparison to the bass line delivered by the system presented here (e.g. bars 3, 5). On the other hand, our system delivers more "daring" chord sequences than those delivered by Allan's and Williams' system, expanding the tonal context while expressing similar harmonic changes (e.g. bars 2, 6).

The example demonstrates the ability of Allan and Williams' system to create rhythmically independent voices (e.g. bars 4, 7). Such a feature is supported neither by the current *vielklang*-version nor by the presented system . This is due to their integration within an audio framework, which in turn synthesizes the additional voices computed by the harmonization system by pitch-shifting the original input audio signal. Rhythmically asynchronous additional voices pose a difficulty for the audio framework which is yet to be addressed.

## 4. Conclusion

The presented system creates a full four-voiced harmonization to an input melody, defined solely through its key and the pitches and durations of its notes. The system delivers correct results with regard to rules of harmony and to conventions of voice leading, and displays a reasonable musical sophistication. A user may define the desired voice range, the position of the melody voice in the harmonization and the desired relative distance of the voices. In addition, a user may ask the system to insert a certain chord at any point along the melody – the system finds the voicing with the optimal fit at that location and writes it into its result.

Refinement or expansion of the musical style implemented in the system as well as the definition of other, different styles may be easily achieved through:

- Enhancements or changes of the chord data-base
- Different configuration of the weights used within the implemented rules
- Enhancements or changes of the rule-sets

Thus, an expansion or specialization of the system may be achieved without the need to alter the input format for the melody – all further necessary information concerning the melody is gained during the analytic stage, and is independent of the implemented musical style.

Future work may include:

- Training of the HMM with harmonically analyzed melodies. The performance of the harmonic prediction may thus be improved. In addition, it may be possible to characterize functional-harmonic structures within a training set of melodies. Such functionality could be useful for musicological research.

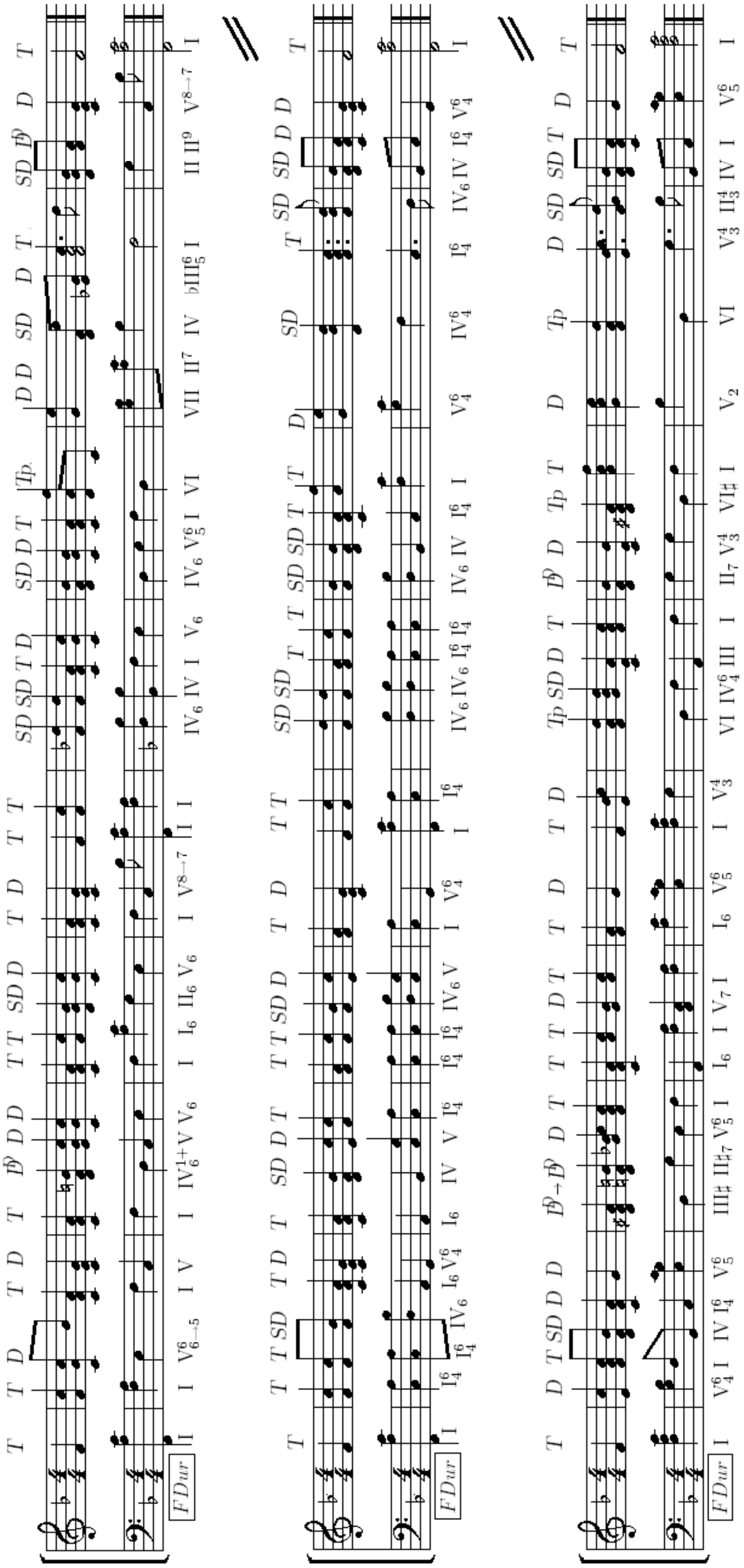

*Figure 2: Harmonization of the melody of J. S. Bach's choral BWV 438 as delivered by (top down) Moray Allan and Christopher K. I. Williams' harmonization system [10, Page 29], version 1.2 of the vielklang-plugin and the system presented here.*

- Style-specific configuration kits with complete chord collections and dedicated sets of rules and weights.
- An evaluation tool utilizing the system's rule sets to test voicings and transitions within input harmonizations. Such an implementation might be useful for students and teachers of harmony and counterpoint.
- Further musical features such as modulations, polyphony etc.

Owing to its conceptual design and its analytical proficiency, the presented harmonhization algrotihm may be successfully used in a large variety of different applications and fields of research. Still, its architecture allows the user to interact with the system in a creative and playful manner.

## 5. Bibliography

[1] François Pachet and Pierre Roy. Mixing Constraints and Objects: a Case Study in Automatic Harmonization. Prentice-Hall, TOOLS Europe: 119-126, 1995.

[2] Kemal Ebcioğlu. An Expert System for Harmonizing Four-Part Chorales. Computer Music Journal, 12(3): 43-51, 1988.

[3] François Pachet and Pierre Roy. Musical Harmonization with Constraints: A Survey. Kluwer Publisher, 6(1): 7-19, 2001.

[4] Somnuk Phon-Amnuaisuk and Geraint A. Wiggins. The Four-part Harmonisation Problem: A comparison between Genetic Algorithms and a Rule-Based System. Proceedings of the AISB '99 Symposium on Musical Creativity: 28-34, 1999.

[5] J. Sabater, J. L. Arcos and R. López de Mántaras. Using Rules to support Case-Based Reasoning for harmonizing melodies. Multimodal Reasoning: Papers from the 1998 AAAI Spring Symposium (Technical Report, WS-98-04). Menlo Park, CA, AAAI Press: 147-151, 1998.

[6] Ching-Hua Chuan and Elaine Chew. A Hybride System for Automatic Generation of Style-Specific Accomaniment. Preceedings of the 4th International Joint Workshop on Computational Creativity, London, UK, 2007: 57-64

[7] Kaan M. Biyikoglu. A MARKOV MODEL FOR CHORALE HARMONIZATION. Preceedings of the 5th Triennial ESCOM Conference: 81-84, 2003

[8] Jean-François Paiement, Douglas Eck and Samy Bengio. Probabilistic Melodic Harmonization. Lecture Notes in Computer Science, http://www.springerlink.com/content/95nx\\16v730344222/fulltext.pdf, 4013: 218-229, 2006

[9] James Tenny and Larry Polansky. Temporal Gestalt Perception in Music. Journal of Music Theory, 24(2): 205-241, 1980.

[10] Moray Allan and Christopher K. I. Williams. Harmonising Chorals by Probabilistic Inference. Advances in Neural Information Processing Systems, 17: 25-32, 2005.