



HAL
open science

A Self-adaptive Learning Music Composition Algorithm as Virtual Tutor

Michele Della Ventura

► **To cite this version:**

Michele Della Ventura. A Self-adaptive Learning Music Composition Algorithm as Virtual Tutor. 18th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2022, Hersonissos, Greece. pp.16-26, 10.1007/978-3-031-08333-4_2 . hal-04317186

HAL Id: hal-04317186

<https://inria.hal.science/hal-04317186v1>

Submitted on 1 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

A Self-Adaptive Learning Music Composition Algorithm as Virtual Tutor

Michele Della Ventura

Music Academy “Studio Musica”, Department of Music Technology
micheledellaventura.mdv@gmail.com

Abstract. The rapid development of digital technologies has made it possible to enhance teaching and learning spaces even in a complex discipline such as music composition, where however the right attention is still lacking regarding the learning style of students. Existing software, in most cases, propose methods for harmonizing a melody, but they rarely deal with the problems of the harmonization of music bassline. This article presents an intelligent adaptive algorithm to support the learning process of (dyslexic and non-dyslexic) students in the context of the harmonization of music basslines. The algorithm allows students to harmonize a music bassline, giving them advices (Virtual Tutor) in case of mistakes in the concatenation of the chords. Experiments have shown that this form of adaptive learning of the algorithm can improve students' ability to find solutions for the harmonization of music bassline. Future improvements of the method are discussed briefly at the end of the paper.

Keywords: artificial intelligence, computer music composition, dyslexia, e-tutoring, music education.

1 Introduction

The transition to the new millennium highlights the decline of the old industrial society and the emergence of the information society [1-2] with the new challenges proposed by artificial intelligence (AI). Even music is not immune to this change, on the contrary it was one of the first areas of experimentation and research [3]: from this perspective it is therefore evident the existence of a broad sharing of strategies and methods between musical composition and intelligence artificial. The hardware and software developments, capable of opening new possibilities to musical verifications and experiments, can be seen as multipliers of creativity (for a musician and not only: it is not strange today to find non-musicians who compose music) but at the same time they can lead to an impoverishment of musical skills and musical experience [4]. Musical composition is a complex discipline based on a series of musical grammar rules whose knowledge is essential. The acquisition and mastery of the musical grammar rules represents an obstacle for students in general but especially for dyslexic students who often find themselves in a situation of lack of autonomy and therefore of uncertainty in the carry out an exercise that leads them not to continue with the task. Therefore, the starting point of the challenges related to AI cannot and must not be technology, but a careful analysis and

identification of the problems that can be faced with AI. Otherwise, any intelligent system designed without adequate problem analysis becomes useless and ineffective: the ability to carry out procedures without understanding the general concepts and connections is very limiting. There are two main dimensions that can be explored, as regards AI and teaching activity [5]:

1. how AI can improve teaching by helping to solve some problems in the school world,
2. how it is possible to educate people (teachers and students) about AI, so that they can benefit from it.

Starting from these two reflections, this research article intends to present a useful algorithm to support (dyslexic and non-dyslexic) students in the individual study, during the harmonization of a music bassline: this is the first step of the study of music composition, and aims to allow the student to know, learn and consolidate the musical grammar rules. The algorithm, through a training phase, is able to self-learn the fundamental musical grammar rules and therefore provide the student with possible solutions in case of errors made during the harmonization of the chords. The algorithm presents itself as a virtual tutor, to facilitate the learning process: the virtual tutor must support the student [6] in the creation and manipulation of musical objects [7], without replacing it by making the composition.

This paper is organized as follows.

Section 2 provides an overview of earlier automatic music composition researches. Section 3 describes the musical grammar rules as far as chords are concerned. This is followed by a description of the method used to formalize the musical grammar rules. Section 4 shows some experimental tests that illustrate the effectiveness of the proposed method. Finally, in Section 5 the paper ends with concluding remarks.

2 Related Works

Computer-assisted musical composition is a synthesis of the evolution of computer-music and computational-musicology. In other words, it is a phenomenon characterized by the existence of software tools that help the composer in the creation and manipulation of musical objects such as a melody, a rhythm or a musical accompaniment. Many algorithms have been designed and developed for these purposes, using different techniques: only those taken into consideration for the development of this research are illustrated below.

A Markov chain is a stochastic process, that is, a random process in which the transition probability that determines the transition to a system state depends only on the immediately preceding system state [8-9]. Due to this characteristic, Markov chains allow the creation of a musical melody, seen as a sequence of notes [10-11]. From the point of view of the creative process, Markov chains could create very similar melodies when the order of the chain is very high, but they can provide new combinations of smaller sections (such as a musical motif) that can be reworked with other techniques [12].

A "generative grammar" is a system of explicit and formal rules with an associated lexicon. These rules are formulated in such a way as to be able to generate a set of sentences, that is, combinations of lexical elements, which represent a language

[13]. The language defined by a set of rules is made up of all the combinations of the elements that the rules can produce and only those combinations [14]. This approach can be used to chords generation but also to create variation on melodies [15].

A rules-based system uses rules to represent knowledge. Music theory traditionally describes rules that help to guide the compositional process. One way to implement rules in compositional systems is to use Constraint Programming, whose declarative nature is well fit to describe music theory rules [16]. This method can be seen as a guide for the composer and allows you to create new melodies, harmonies and rhythms, but at the same time it represents a limitation because it reduces the variety of the output [17-18].

Most computer-assisted musical composition systems deal with the generation of a melody or the harmonization of a melody: there are few studies involving the harmonization of a bass line. [19-20].

The focus of the present research is the harmonization of a musical bassline. In this work, we introduce an algorithm able to support the student during the didactic activity in order to enhance the learning process. That's why the primary goal of the algorithm is not to autonomously harmonizing a bassline to obtain an error-free solution, but to support (as a virtual tutor) the student during the harmonization of a bassline giving him/her advices in case of errors in the concatenation of the chords.

3 Designing the virtual tutor

The algorithm proposed in this article is inspired by the techniques described in the previous paragraph in order to represent and reason ~~on~~ about the compositional rules. The composition rules are modelled so that the algorithm defines the requirements for an advice to be musically valid. The next paragraphs present an overview of the main musical grammar rules, which are fundamental for the design of the proposed algorithm, and the methods used to formalize them.

3.1 Musical composition guidelines

Harmonizing a bass line is an important element of the musical language. It is a traditional system of organizing chords for 4 voices (soprano, alto, tenor and bass) characterized by two aspects [21]:

- 1) there are 4 voices and each one is singing a melody: this is called melodic movement and represents the horizontal aspect of the harmonization (see figure 1, green notes);
- 2) the 4 voices sing simultaneously producing 4-note chords: this is the harmonic, or "vertical", aspect of the harmonization (see figure 1, orange notes). When the three upper voices (Soprano, Alto and Tenor) are within the span of one octave, they are in what is called "close position" (see figure 1, red notes). When the tenor and the soprano are further apart than an octave, the chord is in "open position" (see figure 1, red notes).

The figure shows a musical score in 4/4 time with four staves: Soprano, Alto, Tenor, and Bass. A yellow arrow labeled 'The harmonic aspect' points upwards from the bass line to the soprano line, indicating simultaneous notes in different voices. A green arrow labeled 'The melodic aspect' points horizontally from left to right across the staves, indicating a sequence of notes in the same voice. Red annotations include 'Close position' and 'Open position' with arrows pointing to specific intervals between notes in the soprano line.

Fig. 1. The harmonic aspect and the melodic aspect of a musical composition.

In harmonizing a bassline, these two aspects must be considered at the same time and therefore two distinct movements can be identified: the melodic motion, when there is a passage from one sound to another in the same voice (see figure 1, green notes); the harmonic motion, when two or more melodic motions occur at the same time (see figure 1, red notes). In this context it is necessary to respect some fundamental guidelines. In the melodic motion must be avoided (see figure 2) [21]:

- augmented intervals (intervals that are one half-step larger than a perfect or major interval),
- diminished intervals (Intervals that are one half-step smaller than a perfect or minor interval),
- the intervals of the seventh,
- intervals larger than an octave.

The figure shows a single musical staff with five intervals marked. A red bracket above the staff is labeled 'Forbidden melodic intervals'. Below the staff, the intervals are labeled: Perfect interval, Augmented interval, Diminished interval, Seventh interval, and Interval larger than an octave.

Fig. 2. Examples of forbidden melodic intervals.

In the harmonic motion it is forbidden to proceed with two same voices at a distance of fifths and octaves (see figure 3) [21].

The figure shows a musical score in 4/4 time with two staves: Treble and Bass. Red annotations include 'Parallel fifths not allowed' and 'Parallel octave not allowed' with arrows pointing to specific intervals between notes in the same voice.

Fig. 3. Examples of parallel fifths and parallel octaves.

3.2 Error recognition

In order to be able to identify one of the prohibited movements described above, it is necessary to take into account the concept of musical interval, that means, the distance in pitch between any two notes [22]. For the purpose of calculating this

distance, the musical score is seen as a list of numbers each of which corresponds to a sound, based on its pitch [23]. The list can be represented in the form of a sequence S_m of N notes n_i indexed according to the order of appearance i :

$$S_m = (n_i)_{i \in [0, N-1]} \quad (1)$$

The pitch of a note can be measured by following two criteria [24]. In the first case (called “real distance”), semitones are used as the unit of measurement (regardless of the key of the piece of music):

C=0 D=2 E=4 F=5 G=7 A=9 B=11 etc.

In the second case (called (“tonal distance”), the key of the musical piece is considered and the pitch of a sound is measured by referring to the intervals of the musical scale of the key in question, thus measuring it in degrees:

C=0 D=1 E=2 F=3 G=4 A=5 B=6 ecc.

With reference to the first criteria, it is important to consider simultaneously both the distance of the interval and the type of interval (the interval quality).

The distance is calculated by counting how many names of letters of the musical alphabet there are between the two notes. For example, the notes C and E are three letter notes apart and so is an interval of a 3rd.

The type of the interval is defined by the number of tones and semi-tones between the two notes. It may be (see figure 4): perfect (P), major (M), minor (m), augmented (A), diminished (d).

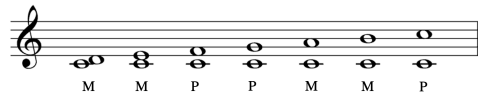


Fig. 4. Types of intervals on the C major scale.

If any of the major intervals are reduced by a semitone, then they become minor intervals (see figure 5). By flattening any of the major or perfect interval by a semitone, they become augmented intervals (see figure 5). By flattening any of the three perfect intervals or a minor interval by a semitone, these become diminished intervals (see figure 5).

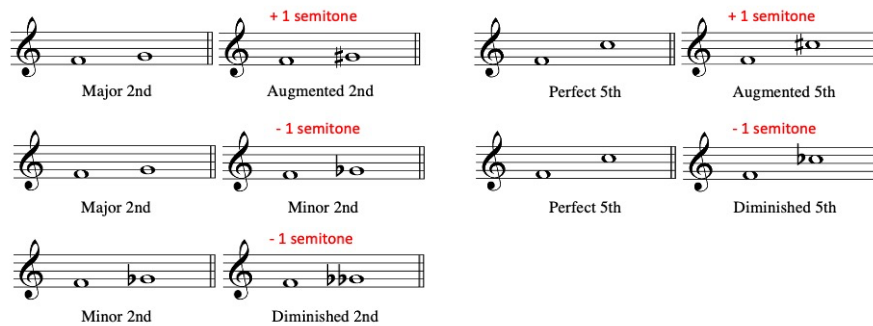


Fig. 5. Types of intervals on the C major scale.

In order to define the interval correctly (thus excluding forbidden intervals), and check a melodic error, it is essential to consider both the tonal distance and the real

distance: considering only the number of semitones between two sounds, it is not possible to distinguish between different intervals [23]. In the example of figure 6 it is possible to see how between the notes F-G# and F-Ab there is an interval of 3 semitones: in the first case it is an increased interval, while in the second case a smaller interval. Considering the tonal distance first, it is possible to obtain the denomination of the interval (F-G# = second interval; F-Ab = third interval); subsequently, with the real distance the interval can be qualified (F-G# = augmented second; F-Ab = minor third).

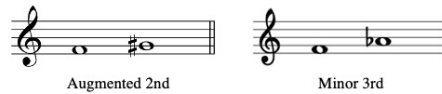


Fig. 6. Example of intervals.

In the harmonic motion it is necessary to verify that in the concatenation of two chords, there is no fifth or octave interval between the same pair of voices (see figure 3). To do so the *Markov process* [24] is used: the choice was made to describe the passage from one state of the system (number of semitones of the same pair of voices in the first chord) to the next (number of semitones of the same pair of voices in the second chord) uniquely from the immediately preceding state [25-26].

Figure 7 shows the matrix of transitions, created by the algorithm during the training phase (see paragraph 4), where in column 1 and row 1 the number of semitones between two voices is indicated, while in column 2 and row 2 it is indicated whether it is a new chord (n) or a repeated chord (r). In this way, if two voices repeat the same sounds in two consecutive chords (see figure 8), the algorithm does not report an error.

| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | | |
|----|---|---|---|---|---|---|---|---|---|---|----|----|----|---|---|
| | | n | r | n | r | n | r | n | r | n | r | n | r | n | r |
| 1 | n | | | | | | | | | | | | | | |
| 1 | r | | | | | | | | | | | | | | |
| 2 | n | | | | | * | * | | * | | | | | | |
| 2 | r | | | | | * | | | | | | | | | |
| 3 | n | | * | * | * | * | * | | * | * | * | | | * | |
| 3 | r | | * | * | * | * | * | | * | * | * | | | * | * |
| 4 | n | | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 4 | r | | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 5 | n | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 5 | r | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 6 | n | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 6 | r | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 7 | n | | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 7 | r | | * | * | * | * | * | * | * | * | * | * | * | * | * |
| 8 | n | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 8 | r | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 9 | n | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 9 | r | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 10 | n | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 10 | r | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 11 | n | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 11 | r | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 12 | n | | | * | * | * | * | * | * | * | * | * | * | * | * |
| 12 | r | | | * | * | * | * | * | * | * | * | * | * | * | * |

Fig. 7. Transitions matrix related to the harmonic motion.



Fig. 8. First 2 bars from Piano Sonata op.53 n.21 by Beethoven: example of repeated fifth (between bass and tenor) and octave (between bass and alto).

From the analysis of the transition matrix (figure 7) it is possible to note how the progressions of perfect fifths (7 semitones) and perfect octaves (12 semitones) have not been identified (orange cells).

4 Obtained results

The proposed algorithm differs from most of the algorithms already developed for the automatic harmonization of the melody or bass, due to the fact that it takes into account the error as a working tool. Through a self-learning procedure, the prohibited melodic and harmonic motions are formalized, in order to identify any errors made by the student in harmonizing the sounds of the bassline, and suggest possible solutions.

To test the effectiveness of the algorithm, the following steps were followed.

1) Data collection

In order to train and test the algorithm, around 1300 musical pieces (written in choral form) were collected (Dataset): the pieces could be both in close position or in open position. The musical scores were in MIDI format [27] or in XML format [28], and this made it possible to take advantage of the numerous archives of musical pieces (of public domain) written in choral form, present on the web.

2) Data cleaning and preparation

The musical scores were analyzed before the dataset was used for the training phase of the algorithm. The use of incorrect data can produce misleading results. This phase took a long time (due to the fact that every single score was analyzed to identify any transcription errors) but it was essential for the veracity of the results.

3) Training phase of the algorithm

The algorithm read the musical scores in order to create the table representing the correct melodic motions (see table 1) and the matrix of transitions representing the correct harmonic motions (see figure 12). Both the table representing the correct melodic motions and the matrix of transitions representing the correct harmonic motions were automatically dimensioned on the basis of the characteristics of the musical piece read during this phase.

4) Model test

Two experimental tests were carried out to verify the integrity of the algorithm. In the first test, 7 music composition teachers used the software to harmonize a musical bassline, first in close positions and then in open positions. This allowed to verify if the algorithm updated the libraries in the case of solutions other than those recognized during the training phase, or in the event of an error if the solutions

proposed by the algorithm could satisfy the musical grammar rules. Subsequently, to verify the soundness of the algorithm, teachers were asked to create a choral (8 bars) without having any musical inspiration but using their own personal creativity. In the second experiment, a group of 40 students (with different levels of preparation and different learning styles) from 2 Music High Schools and 2 Conservatories of music, was asked to harmonize a bassline under the supervision of a Composition teacher who had to note the students' difficulties and errors and the answers of the algorithm.

After this testing phase, 3 teachers (2 from Music High Schools and 1 from Conservatory of music) were asked to permit students the use of the software for a period of 3 months (for their individual study) and to carry out a verification test every month in order to check the learning process.

The results obtained at the end of this period were satisfactory.

On the one hand, the students declared that they found no difficulty in using the software (because it was simple and intuitive), and the advices proposed in the event of an error forced them: to reflect and recognize the type of error; to consult the school book to understand the proposed solution, thus consolidating knowledge.

On the other hand, the teachers expressed their satisfaction: as regards the solutions proposed by the algorithm in the event of an error; regarding the behavior of different students, with particular attention to dyslexic students; as regards the results achieved by the students who, even if the melodic lines obtained by harmonizing the bass were less appreciable, the students had reduced the number of errors in carrying out exercises proposed and carried out independently.

Table 1. Excerpt of the allowed interval identified by the algorithm during the training phase.

| Interval | Tonal distance | Real distance |
|----------|----------------|---------------|
| 2 | 1 | 2 |
| | 1 | 1 |
| 3 | 2 | 4 |
| | 2 | 3 |
| 4 | 3 | 5 |
| 5 | 4 | 7 |
| 6 | 5 | 9 |
| | 5 | 8 |
| 7 | 6 | 11 |
| | 6 | 10 |
| 8 | 7 | 12 |
| ... | ... | ... |

5 Discussion and Conclusions

This article presented a self-adaptive learning music composition algorithm, specifically designed to enhance the educational activity, and support the student during the learning process. The algorithm presents itself as a virtual tutor able to support the student during the harmonization of the music bassline, helping him/her to consolidate the musical grammar rules. The experiments conducted show that the use of the algorithm can be beneficial: it can improve the student's ability to

recognize which musical grammar rules is necessary to use during the concatenation of the chords and can help him/her to consolidate acquired skills.

An advantage of the algorithm is that there are no control parameters to be set manually. It is based on a self-learning model able to acquire on the one hand the rules relating to the movement of two sounds within the same voice (using the principles of information theory) and on the other hand the rules for the movement of the sounds of the various voices between two consecutive chords (by means of the Markov process).

Artificial intelligence is able to provide teachers with an insightful virtual tutor who is able to help students to solve the challenges and obstacles encountered during their learning path. AI can offer students explanations for a better understanding of disciplinary topics and enrich their knowledge without making them feel inferior. The virtual tutor allows learners, even when they fail, to remedy by trial and error and also to learn from mistakes. Often the fear of making a mistake, for many students (especially dyslexic students), is really limiting. If a virtual tutor is effective, this increases the probability that the student will use an application for self-study more often, improving their skills and competences.

Future research should be oriented towards the implementation of the algorithm to support the student (dyslexic and non-dyslexic) during the harmonization of a bassline in which key changes are present: these are the cause of many harmonization errors due to the change of accidentals (sharps, flats).

References

1. Feenberg, A. (1991). *Critical theory of technology*, Oxford University Press, Oxford.
2. Bel, B., Vecchione, B. (1993). Computational musicology. *Comput Hum* **27**, 1–5 (1993). DOI: <https://doi.org/10.1007/BF01830711>
3. Hamman, M. (2002). From technical to technological: The Imperative of Technology in Experimental composition, in *Perspectives in new music* 40-1, 2002.
4. Coutinho, E., Gimenes, M., Martins, J., Miranda, E. (2005). *Computational Musicology: An Artificial Life Approach*. DOI: 10.1109/EPIA.2005.341270
5. Casini, L., Rocchetti, M. (2018). The impact of AI on the musical world: will musicians be obsolete? *Mimesis edizioni* 4(3), pp. 119-134. DOI: 10.7413/18258646064
6. Chen, N.S., Cheng, I.L., & Chew, S.W. (2016). Evolution is not enough: Revolutionizing current learning environments to smart learning environments. *International Journal of Artificial Intelligence in Education*, 26(2), pp. 561-581.
7. Collins, A., & Halverson, R. (2018). *Rethinking education in the age of technology: The digital revolution and schooling in America*. Teachers College Press.
8. Brèmaud, P. (2013). *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, Vol 31. New York, NY: Springer Science & Business Media.
9. Della Ventura, M. (2015), *The Influence of the Rhythm with the Pitch on Melodic Segmentation*, In *Proceedings of the Second Euro-China Conference on Intelligent Data Analysis and Applications (ECC 2015)*, Ostrava, Czech Republic, Springer. DOI: https://doi.org/10.1007/978-3-319-21206-7_17
10. Pachet, F. (2002). "Interacting with a musical learning system: the continuator," in *Music and Artificial Intelligence* (Berlin: Springer), 119–132.

11. Shapiro, I., Huber, M. (2021). Markov Chains for Computer Music Generation. *Journal of Humanistic Mathematics*, Volume 11 Issue 2 (July 2021), pages 167-195. DOI: 10.5642/jhummath.202102.08
12. Papadopoulos, A., Roy, P., and Pachet, F. (2014). Avoiding plagiarism in Markov sequence generation, in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14* (Quebec City, QC: AAAI Press), 2731–2737.
13. Chomsky, N. (1957). *Syntactic Structures*. The Hague: Janua Linguarum. Mouton & Co.
14. Hamanaka, M., Hirata, K., and Tojo, S. (2008). “Melody morphing method based on GTTM,” in *ICMC* (Ann Arbor, MI: Michigan Publishing), 155–158.
15. Wiggins, G. A. (2019). “A framework for description, analysis and comparison of creative systems,” in *Computational Creativity: The Philosophy and Engineering of Autonomously Creative Systems, Computational Synthesis and Creative Systems*, eds T. Veale and F. A. Cardoso (Cham: Springer International Publishing), 21–47.
16. Anders, T., and Miranda, E. R. (2011). Constraint programming systems for modeling music theories and composition. *ACM Comput. Surv.* 43, 1–38. DOI: 10.1145/1978802.1978809
17. Cunha, N. d. S., Subramanian, A., and Herremans, D. (2018). Generating guitar solos by integer programming. *J. Operat. Res. Soc.* 69, 971–985. DOI: 10.1080/01605682.2017.1390528
18. de Mántaras, R.L. (2006). Making music with AI: some examples. In *Rob Milne: A tribute to a Pioneering AI Scientist, Entrepreneur and Mountaineer*, A. Bundy and S. Wilson, Eds. IOS Press, 2006, pp. 90–100.
19. Rothgeb, J. (1969). Simulating musical skills by digital computer. Reprinted in *Schwanauer, S.M., and Levitt, D.A., ed. Machine Models of Music*. 157-164. Cambridge, Mass.: The MIT Press.1993.
20. Wassermann, G., Glickman, M. (2020). Automated Harmonization of Bass Lines from Bach Chorales: A Hybrid Approach. *Computer Music Journal* 2020; 43 (2-3): 142–157. DOI: https://doi.org/10.1162/comj_a_00523
21. Coltro, B. (1997). *Lezioni di armonia complementare*. Ed. Zanibon.
22. Della Ventura, M. (2011). Analysis of algorithms’ implementation for melodical operators in symbolical textual segmentation and connected evaluation of musical entropy, in *Proceeding of The International Conference on Models and Methods in Applied Sciences, Drobeta Turnu Severin*, 2011, pp. 66-73.
23. Della Ventura, M. (2018). *DNA Musicale: matematicamente suono*, ABEditore, Milano.
24. Della Ventura, M. (2021). Automatic Recognition of Key Modulations in Symbolic Musical Pieces Using Information Theory. In: Arai K. (eds) *Intelligent Systems and Applications. IntelliSys 2021. Lecture Notes in Networks and Systems*, vol 294. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-82193-7_56
25. Weaver, W., Shannon, C. (1964). *The mathematical theory of information*, Urbana, Illinois Press.
26. Della Ventura, M. (2018). Voice Separation in Polyphonic Music: Information Theory Approach, In *Proceedings of the 14th International Conference on Artificial Intelligence Applications and Innovations (AIAI 2018)*, Rhodes, Greece, Springer. DOI: https://doi.org/10.1007/978-3-319-92007-8_54
27. Cooper, L. (1986). Mind over MIDI: Information sources and system-exclusive data formats. *Keyboard* October, 110-111.
28. DuCharme, B. (1999). *XML: The Annotated Specification*. Upper Saddle River, NJ: Prentice Hall PTR.