https://doi.org/10.7494/978-83-67427-06-7_2

Robust, hybrid algorithms in AI-aided automatic music production

Paweł Skrzyński[1], Joanna Kwiecień[1], Marek Pluta[1*], Andrzej Dąbrowski[2], Bartłomiej Szadkowski[2]

[1] AGH University of Science and Technology in Krakow
[2] Independent Digital Sp. z o.o.
[*] pluta@agh.edu.pl

Abstract

Contemporary music market is vastly dominated by streaming services. Growing popularity of streamed music consumption enables each music genre to find its niche. Some genres are defined well enough in terms of musical form and structure to carry out attempts at making the process of their production fully automated. This work presents a selection of algorithms designed for automatic music production. The algorithms are implemented in a fully functional production system that uses a hybrid approach, combining robustness and controllability of classic solutions with ability to generate and evaluate complex structures inherent for AI-based methods. So far, the approach has been applied to produce relaxation, dance, and electronic music – genres that are popular in streaming services, and well suited for automatic production.

1. Introduction

The era of streaming services and advances in audio engineering brought better, easier to use, broadly available tools for musicians and music producers, matched by convenient means to quickly publish new music worldwide. It enables more people than ever to be involved on both sides: firstly – in, or around production of music, and secondly – in reception, or as it is nowadays more often referred to, in consumption of music. It is a very favourable time for musical experiments – vast domination of streaming services, and popularity of streamed music consumption enables each music genre to find its niche. Some of more interesting experiments, that have been attempted times and again before, but only recently received appropriate means to succeed, are the experiments in automatic creation of music: not only algorithmic composition, but full process of music production.

This work presents a selection of algorithms implemented in a system designed to produce music automatically. While an automatic composition is a relatively old idea [1–3], the system proposed is a fully-automated tool not only for composing, but also for producing music – from a concept to a recording (Fig. 1). It has been designed to work in a production environment, to provide required amount of musical material upon request. As a consequence, reliability is one of its essential qualities – it needs to produce acceptable results consistently. Therefore, while designing the system, a hybrid solution has been applied to combine robustness and controllability of classic approaches with ability to generate and evaluate complex structures inherent for AI [4, 5].

Algorithms presented in this work are general, and can be applied to produce various music genres. However, not all genres are equally well defined in terms of musical form and structure. Consequently, in current state the system has been applied to produce music belonging to three genres that are popular in streaming services, and well suited for automatic production: relaxation, dance, and electronic music.

User-defined settings
Music production system
Audio recording

Fig. 1. Automatic music production

2. Background

In order for the automated music production to be considered, several techniques need to be available. The process involves sound synthesis, automatic mixing and mastering. It starts, however, with an automatic composition, or broader – with algorithmic composition, which predates all of computer techniques. Its roots reach out to early polyphonic music, when the rules of counterpoint began to take shape. Once matured, these rules allowed to compose proper, natural melodic lines, and join these lines in parallel into complex multi-voice constructions [3]. The rules consisted of precise and context-aware series of prohibitions, orders, and recommendations concerning horizontal intervals and development, vertical structures and chords, as well as mutual relations between parallel voice behaviour.

Formulation of these rules was strict enough to allow it to be applied in one of the first notable examples of computer-based algorithmic compositions – a four-movement string quartet 'Illiac Suite' by Hiller and Isaacson [1]. Hiller and Isaacson supplemented counterpoint rules with another rule-based composition system, more recent, and coded parts of it work according to 20th century dodecaphonic and serial principles [2]. Even though both systems, counterpoint and dodecaphony, were originally developed as composer's tools, they were successfully implemented to automatically compose an entire musical work with the use of digital computer and probabilistic techniques.

A key difference between the algorithmic composition and the automatic composition is in the role of a composer. In the former, the algorithm is one of tools at composer's disposal. The decisions belong to composer, and the algorithm is used to generate parts of musical material on the chosen level of a composition. In the latter, conversely, the algorithm does not require a composer, as it claims this role itself. All the levels and layers of a composition are designed and arranged by algorithms. User interaction may be limited to providing general preferences. Both approaches are actively developed, and share similar techniques and methods, including Markov chains, formal grammars, rule-based systems, neural networks, deep learning, evolutionary algorithms, chaos similarity, and agents based systems [4, 5]. Deep learning seems well suited for generation of new musical material, while evolutionary algorithms can be applied to create its variations [6]. In the end, evaluation, or classification of generated music may be performed through pattern recognition [7] or feature extraction [8].

With regards to its purpose, relaxation music may be considered a functional, or a background music. Accompanying role requires its qualities to comply with various requirements. Dance and electronic genre may be considered functional music as well, albeit to a lesser degree. Naturally imposed limitations, and well defined target, make these genres particularly well suited for algorithmic approaches, fo the sake of ensuring predictable results [9–11].

3. The design

3.1. Overview

Music production is often described as a process consisting of composition, sound design, arrangement, mixing, and mastering. What is not always mentioned, but shall be regarded a key stage, is a critical evaluation of the effect. Negative outcome of the evaluation moves the process back, and forces repeat of some prior stages. In an attempt to replicate this process in automatic approach, a model has been designed, referred to as 'generator-critic' (Fig. 2). The generator is responsible for creative tasks. The critic accepts or rejects the outcome of generator. In case of rejection the generator starts over, with the same input parameters, but due to its internal design, it will produce a different result, which, again, will be evaluated by the critic.

The generator is given a set of initial directions regarding general characteristics of music to produce. Directions may be provided directly by a human user, or set automatically by a supervising script, if the system works in a batch mode. Given the directions, generator composes a symbolic musical score, arranges it, and transforms it to an audio form, thus producing a ready-to-use recording. However, automatically produced output might not always be up to the requirements of a user. Therefore the critic is given a task to evaluate the final recording, and in case of negative evaluation result, to order the generator to start over, with the same user-settings, but with different internal decisions. The process is repeated as long as the evaluation is negative. The output is produced to the user only after receiving positive evaluation from the critic.

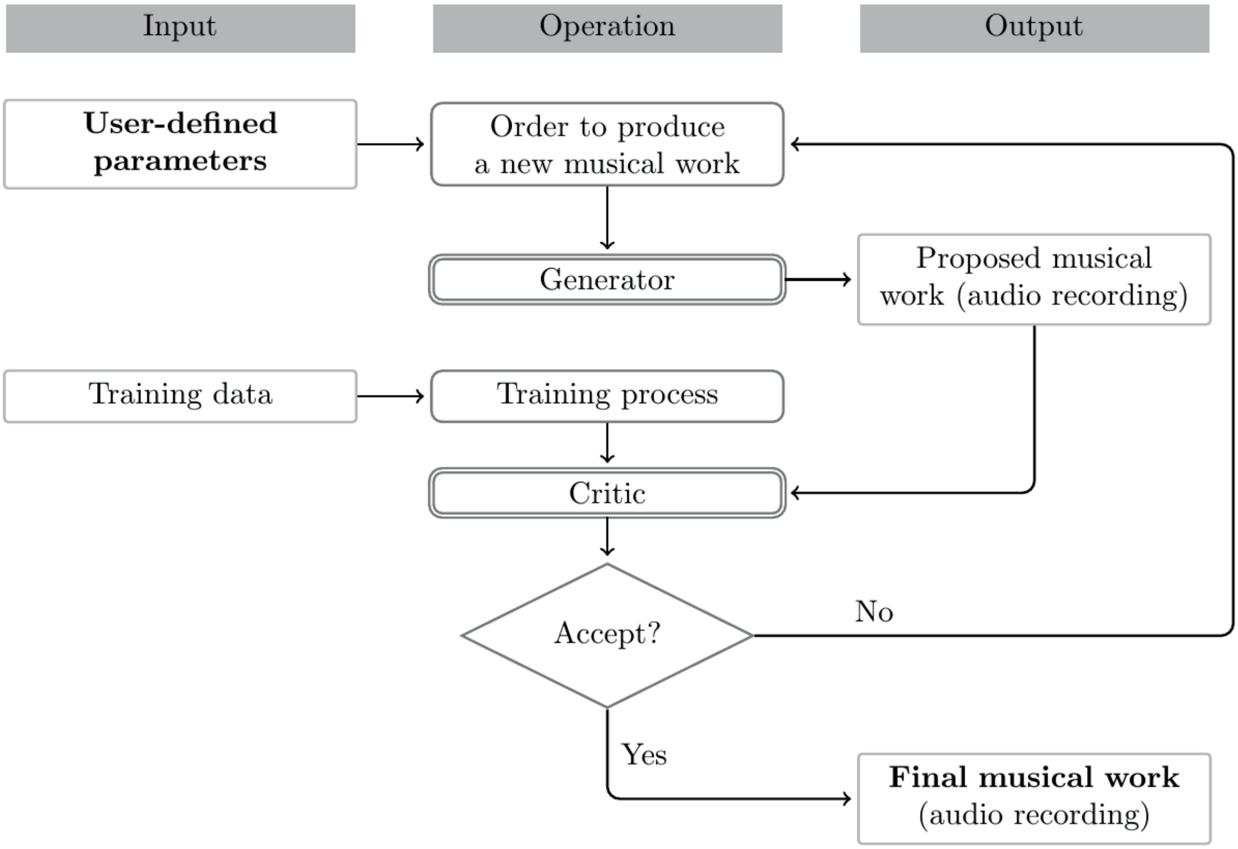
Input
Operation
Output
User-defined parameters
Order to produce a new musical work
Generator
Proposed musical work (audio recording)
Training data
Training process
Critic
Accept?
No
Yes
Final musical work (audio recording)

Fig. 2. The generator-critic model of music production

3.2. Critic and generator

The critic is based on an artificial neural network [12]. Its data are the parameters extracted with music information retrieval (MIR) techniques from final audio signal produced by the generator. The set includes low level audio descriptors (such as dynamic and spectral complexity, or pitch salience), rhythm descriptors (such as beats count and loudness, or onset rate), and tonal descriptors (chords changes rate, or key strength) [8]. Such selection allows the critic to evaluate composition qualities, as well as performance and signal characteristics.

A neural network proved to be a viable solution for the critic, however, no such straightforward solution was available for the generator. In recent years one could observe a fast development and increasing interest in deep learning (DL) techniques applied to art-creation process [13]. While such approach has been considered and tested for the use within the system, the results were often erratic, with unusable output. Moreover, deep networks tend to mimic music used for their training. The problem originates from insufficient amount of music in a score form for particular genres for deep networks to generalize upon. Even with some promising results obtained during development, in its current state DL is not stable and reliable enough. Therefore a different approach has been applied.

The generator performs a chain of operations using classical, and AI-based techniques, the latter including fuzzy logic, genetic algorithms, and rule-based systems. A set of input parameters is interpreted using fuzzy logic. The set is provided either by a human system operator, or by a supervising script in a batch mode operation. It includes genre, duration, tempo, mood, oddity parameter, and fine pitch-tuning information (Tab. 1). They are used to design a form of a musical work to be produced, including harmonic layer and structure of a composition. Rule based systems produce a set of lead motifs, assembled into phrases with genetic algorithm – both techniques apply selected rules of counterpoint to obtain proper structures. Other parts, such as bass, drums, pads, ambient sounds, etc. adapt various combinations of these techniques. Such prepared score is converted to MIDI sequence and passed to a sequencer controlling a sampler that produces audio parts to be mixed, and finally – mastered.

Table 1

User-definable generator input parameters for a single musical work

Input	Interpretation
Genre	Single-choice from the following list: relaxation, dance, electronic
Duration	Target duration [s]
Tempo	Target tempo [BPM]
Mood	Target mood setting in range from *sad* to *happy* – affects scale selection, note density, melody characteristics, etc.
Oddity	Amount of unconventional elements in range from none (setting *normal*) for a conservative composition, to many (setting *odd*) for experimental behaviour
Tuning	Pitch tuning reference point – fundamental frequency of A4 [Hz]

4. Algorithms

A key feature of a process applied in a production environment is its reliability and robustness. While experimental approaches based entirely on DL techniques may produce interesting results, in current state they pose serious problems limiting their usability. A primary limitation is a requirement for very large database of digital scores categorised into styles to learn and generalise upon, which – apart from a few exceptions such as Bach or Mozart – is not, and will not be available. Even provided with appropriate scores, DL-based solutions rarely produce acceptable output without significant per-case human-interaction. Finally, control over the outcome of such processes is limited, so that a user might not affect features of produced music in a meaningful way. On the other hand, traditional approaches, based entirely on predefined rules or statistics, tend to produce predictable results.

As a way to solve above mentioned problems, a process of automatic music production has been divided into stages (Fig. 3) handled by various algorithms integrated into a hybrid system. A few key stages that involve creative tasks are handled by AI algorithms such as genetic algorithms (GA), fuzzy logic (FL), rule-based expert systems (RBS), and artificial neural networks (ANN), as pointed in Table 2.

Table 2
Use of AI-related methods: FL – fuzzy logic, GA – genetic algorithm, RBS – rule-based system, ANN – artificial neural network

TASK	FL	GA	RBS	ANN
Form design	+	–	–	–
Harmonic progression generation	+	–	–	–
Lead motifs generation	+	–	+	–
Lead phrases design	–	+	+	–
Drum patterns design	–	+	+	–
Bass lines generation	+	–	–	–
Accompanying tracks design	+	–	+	–
Applying effects and mixing	–	–	+	–
Classification (critic)	–	–	–	+

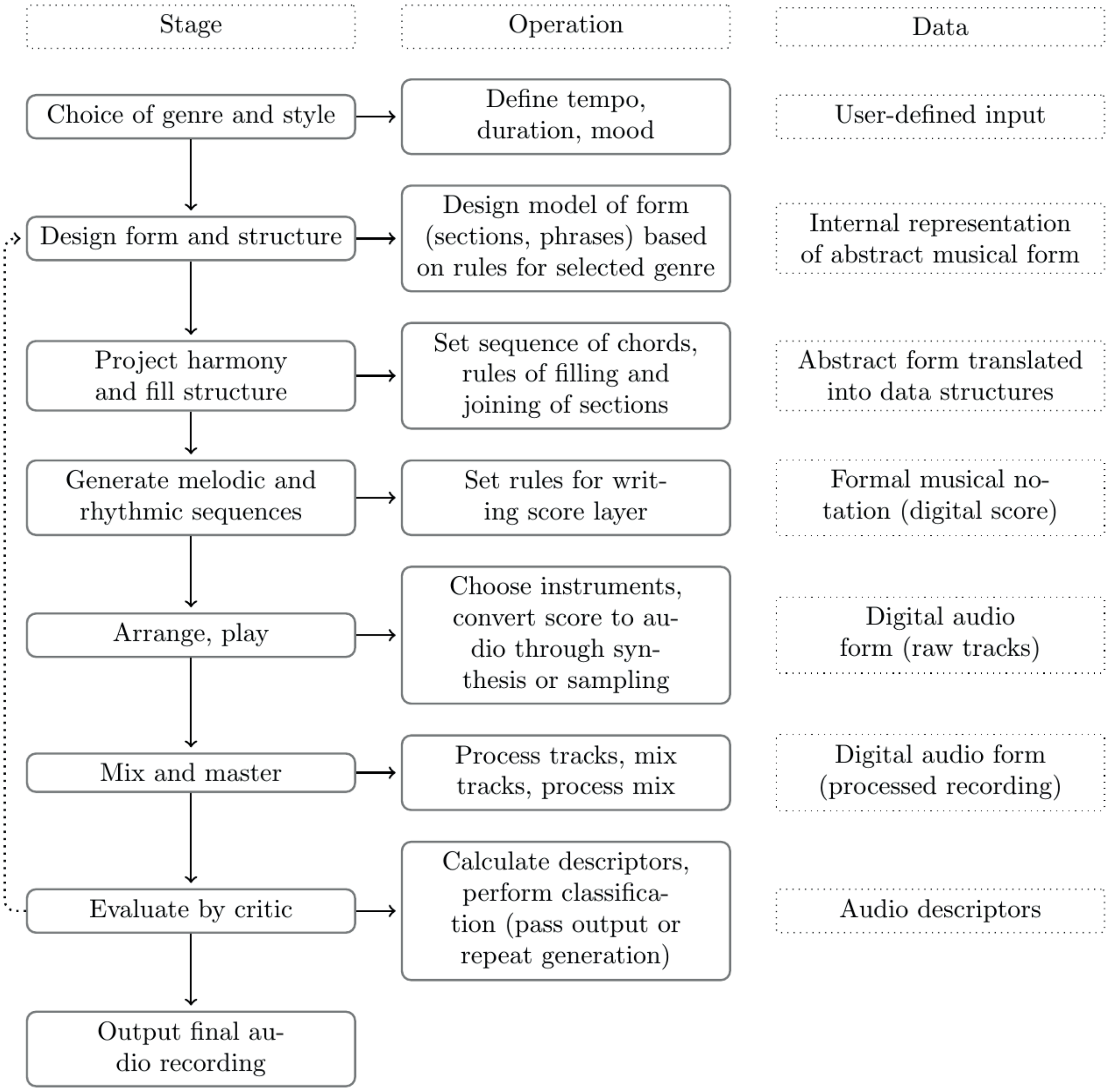
Stage
Operation
Data
Choice of genre and style
Define tempo, duration, mood
User-defined input
Design form and structure
Design model of form (sections, phrases) based on rules for selected genre
Internal representation of abstract musical form
Project harmony and fill structure
Set sequence of chords, rules of filling and joining of sections
Abstract form translated into data structures
Generate melodic and rhythmic sequences
Set rules for writing score layer
Formal musical notation (digital score)
Arrange, play
Choose instruments, convert score to audio through synthesis or sampling
Digital audio form (raw tracks)
Mix and master
Process tracks, mix tracks, process mix
Digital audio form (processed recording)
Evaluate by critic
Calculate descriptors, perform classification (pass output or repeat generation)
Audio descriptors
Output final audio recording

Fig. 3. Stages of automatic music production process

4.1. Form, structure, and harmonic layer

The musical form shapes an overall plan of the composition. It guides division into sections and type of sections sequence. It also controls number, type, and role of instruments. For each genre there is a selection of typical forms assigned, with adjustable characteristics. Selection and adjustments are carried out on the basis of input parameters. FL allows to adjust characteristics gradually, and produce feature-mixed forms.

Once the form has been set, the structure and harmony are designed in parallel, due to their mutual dependence. Harmony is defined as a progression of chords with relations and modes controlled by FL on the basis of input parameters, particularly mood and oddity. It is internally represented by root pitch distances measured in semitones, and coupled with a scale-related chord type.

The structure is designed as a grid, with rows representing instrument tracks, and columns – subsequent sections. Each section is assigned a single chord from the harmonic progression. Section duration may vary, depending on input parameters and length of chord sequence, so that chosen form, such as reprise order (ABA), or sequencing (AB), can be fit into target piece duration. Typically it is a few measures long. For each track a set of different sections is prepared, and arranged into a grid. Within a grid sections can be arranged in any order – they can be sequenced, repeated, looped, or skipped. Their arrangement is guided by selected form. Sections contain the actual score data, generated by dedicated algorithms, depending on role of a track in the form.

4.2. Lead-melody generator

The system is designed to generate various genres of music. Most forms contain either a single lead melody or two lead melodies to be interleaved or played in parallel. Therefore, the melody generator needs to be flexible and configurable. The approach applied is to combine RBS and FL to handle and mix meaningful music features with a GA to combine small elements into varied and evolving, but consistent musical phrases.

Lead melody is generated within two main stages:

- generation of motifs (basic building blocks),
- modification and arrangement of motifs into longer sequences – phrases.

Motifs are the most basic melo-rhythmic sequences, typically few-note long. A set of motifs can be arranged into a sequence as long as required. Generation of motifs is guided by rules of counterpoint that prevent some, while rewarding other pitch intervals, depending on vertical and horizontal note context. Such rules lead to fluent, natural sequences.

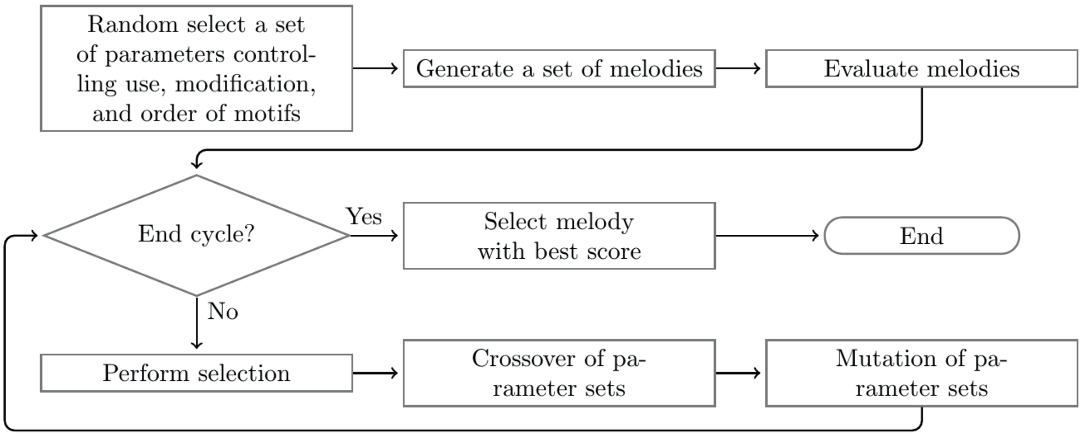
Random select a set of parameters controlling use, modification, and order of motifs
Generate a set of melodies
Evaluate melodies
End cycle?
Yes
Select melody with best score
End
No
Perform selection
Crossover of parameter sets
Mutation of parameter sets

Fig. 4. Lead instrument phrase generator algorithm

A small set of motifs is arranged into longer sequence – a phrase – using a GA-based algorithm presented in Figure 4. However, it is not the melody which is subjected to evolution, but a set of rules guiding composition of a phrase. Yet, the evaluation considers the melody itself. Therefore the GA encoding is a set of parameters controlling transformations and order of motifs, while fitness is estimated using accordance of output melody with rules of counterpoint. As a result, phrases are fluent due to counterpoint restrictions, and mostly homogeneous due to common motif-base – both features are musically desirable in lead melodies. Applied counterpoint rules may be enabled or disabled on the basis of oddity and mood parameters.

4.3. Drum sequence generator

Generator of drum sequences utilises a large set of initial rhythmic patterns. Pattern is a 2-dimensional binary array with 1 representing a note, and 0 – a rest. Rows represent instruments of a percussion set, columns – subsequent time-steps. GA-based algorithm picks two predefined patterns to produce a new one, as shown in Figure 5. In this case the GA encoding is a direct representation of pattern. Number of cycles, as well as other evolution parameters, is controlled by oddity. Initially, the drum pattern is conventional, but with increasing number of cycles it evolves towards unconventional variants.

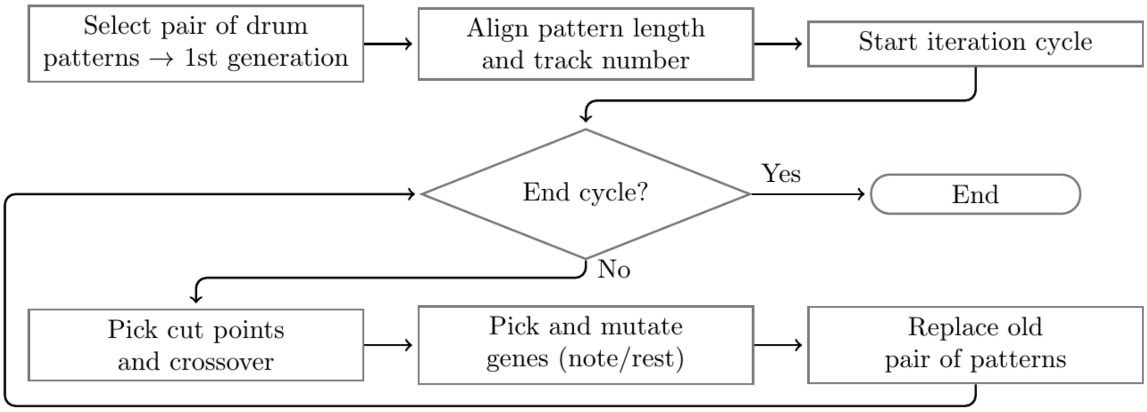
Select pair of drum patterns → 1st generation
Align pattern length and track number
Start iteration cycle
End cycle?
Yes
End
No
Pick cut points and crossover
Pick and mutate genes (note/rest)
Replace old pair of patterns

Fig. 5. Drum pattern generator algorithm

4.4. Accompanying tracks

A set of accompanying tracks vary depending on the form and choice of input parameters, but it can include:

- a bass line,
- a high-pitched accompanying line,
- a slow chord track (pads),
- a rhythmic chord track,
- an arpeggiator track,
- and an ambient track.

Each one has its separate generation algorithm. They are based on large sets of predefined melodic and rhythmic patterns, processed and mixed on the basis of input parameters (genre, tempo, mood and oddity) by FL with an optional aid of RBS. The exception is an ambient track, dedicated for background sounds like wind, rain, or flowing water. These sound are arranged in sequence, with necessary overlapping.

4.5. Instrumentation and effects

In order to convert digital score into a sound recording, the system uses a MIDI sequencer and a sound sampler. Sampler is equipped with a broad range of acoustic and electronic instruments, as well as ambient recordings, organised within a set of sample banks. Depending on genre, each track has a set of assignable instruments, defined by sample bank ID and program ID (Tab. 3).

Once a track is converted from score to a recording by MIDI sequencer and sampler, the system may apply some signal processing effects, such as filters, reverb, modulators, etc. In case of effects that can change over time, rate of change may be adjusted to tempo and rhythm. Afterwards, track levels are adjusted, and tracks are mixed. Finally, depending on genre, some additional effects may be applied to the mix.

Table 3

Example of an instrument assignment for a dance genre; duplicate track names denote multiple choices for a sample bank for a track

Track name	Bank name	Program IDs
Treble	ID_ElectroPlucked.sf2	0 1 2
Lead1	ID_ElectroLead.sf2	0 2 5 6 7 8
Lead2	ID_ElectroLead.sf2	0 2 5 6 7 8
Pads	ID_ElectroPad.sf2	2 3 4 5
Chord	ID_OrganElectric.sf2	0 1 2 3 4
Arp	ID_ElectroPlucked.sf2	0 1 2
Bass	ID_ElectroBassAtt.sf2	0 1 2 3 4 5 6 7 8
Perc1	ID_ElectroDrum.sf2	0 1 2
Perc1	ID_GenericDrum.sf2	0 1 2 3 4
Perc2	ID_ElectroPerc.sf2	0 1
Perc2	ID_GenericPerc.sf2	0 1
Ambient	ID_Ambient.sf2	0 1 2 3 4 5 6 7

4.6. Classifier

Assuming that only a portion of recordings produced by the generator is acceptable, the critic has to classify every new recording into one of two groups: good (acceptable) or bad. The algorithm applied to carry out this task is ANN with supervised learning (Fig. 6).

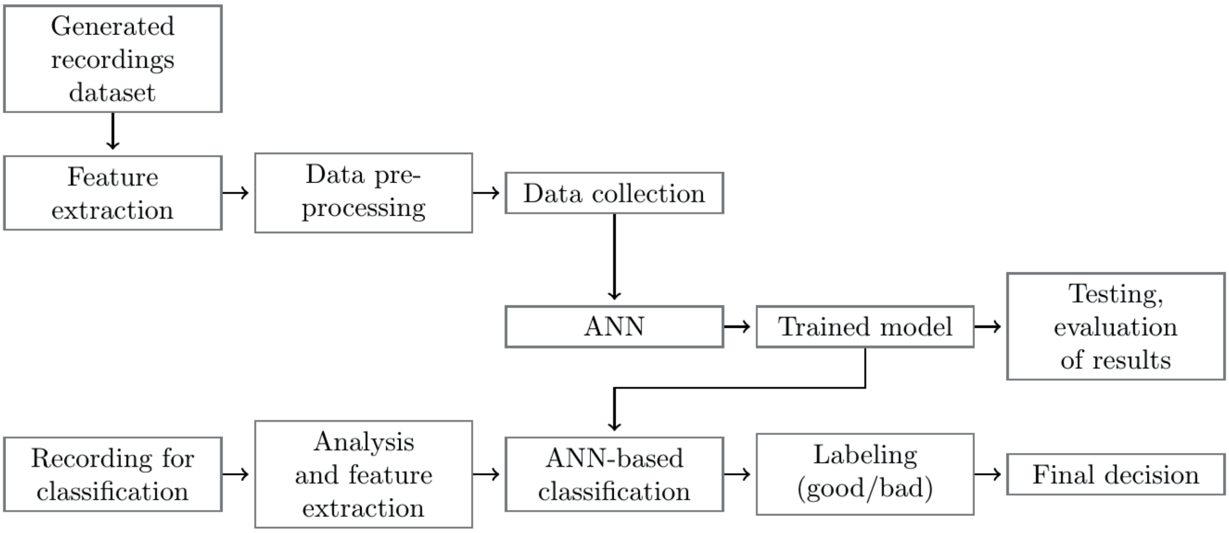

Fig. 6. The critic algorithm

Initially, a large collection of recordings of each genre is produced by the generator and passed to human experts for binary evaluation. For every recording a set of low-level, rhythmic, and tonal descriptors is calculated (Tab. 4). Descriptors with expert evaluation assigned are used as a training set for the multilayer feedforward neural network, with backpropagation algorithm as a training technique. Once trained, ANN is able to generalise and perform classification for new recordings belonging to trained genres.

Table 4

Examples of signal descriptors calculated by the critic to evaluate the recording

Low level descriptors	Rhythm descriptors	Tonal descriptors
• 13 first mel-frequency cepstral coefficients (MFCC) • Dissonance • Dynamic complexity • Pitch salience • Spectral complexity (Shannon entropy of a spectrum) • Spectral energy band (high, low)	• Beats count (number of detected beats) • Beats loudness (spectral energy computed on beats segments) • BPM value • Danceability • Onset rate (number of detected onsets per second)	• Chords changes rate • Key strength using diatonic profile

5. Conclusions

At current stage of development the system produces varied and complete examples of three distinct genres, though the solution is general enough to allow production of other genres as well. All the necessary stages of music production process have been implemented. Due to algorithms applied, the procedure is reliable, configurable, and fast – entire production process is by order of magnitude shorter than duration of generated music. Therefore it is possible to run the system in the background, while playing previously generated piece – thus providing a constant stream of new music.

However, there is always a room for improvement. Further works will involve tuning some of the generator stages, and obtaining broader training set for the critic – the latter involves significant effort of human experts to mark acceptable examples. Quality and diversity of the output may be improved by adding more varied, and higher-quality sound samples to the sampler. Finally, a broader range of means may be applied on the side of signal effects, in addition to the current set of filters, amplifiers, low frequency oscillators and envelope generators.

Acknowledgements

This work is a part of dissemination of the results of R\&D project titled ‘The research and development work on the sound processing and creation of compositions and music recordings based on the artificial intelligence, in the integrated system of the digital content distribution’. Project was co-finansed by the ROP Regional Operational Programme for Mazowieckie Voivodeship 2014-2020, Poland (RPO WM 2014-2020) and realised by Independent Digital Sp. z o. o.

References

[1] L.A. Hiller Jr., L.M. Isaacson, *Musical Composition with a High-Speed Digital Computer*, Journal of the Audio Engineering Society 1958, vol. 6, no. 3, pp. 154–160

[2] R. De Prisco, G. Zaccagnino, R. Zaccagnino, *A Genetic Algorithm for Dodecaphonic Compositions*, in: *Applications of Evolutionary Computation. EvoApplications 2011: EvoCOMNET, EvoFIN, EvoHOT, EvoMUSART, EvoSTIM, and EvoTRANSLOG, Torino, Italy, April 27–29, 2011, Proceedings, Part II*, 2011, pp. 244–253

[3] M. Komosinski, P. Szachewicz, *Automatic species counterpoint composition by means of the dominance relation*, Journal of Mathematics and Music 2015, vol. 9, no. 1, pp. 75–94

[4] J.D. Fernández, F. Vico, *AI Methods in Algorithmic Composition: A Comprehensive Survey*, Journal of Artificial Intelligence Research 2013, vol. 48, pp. 513–582

[5] F. Carnovalini, A. Rodà: *Computational Creativity and Music Generation Systems: An Introduction to the State of the Art*, Frontiers in Artificial Intelligence 2020, vol. 3, 14

[6] P. Donnelly, J. Sheppard, *Evolving Four-Part Harmony Using Genetic Algorithms*, in: C. Di Chio et al. (eds.), *Applications of Evolutionary Computation*, *EvoApplications 2011*, Lecture Notes in Computer Science, vol. 6625, Springer, Berlin–Heidelberg, 2011, pp. 273–282

[7] G. Tzanetakis, P. Cook, *Musical genre classification of audio signals*, IEEE Transactions on Speech and Audio Processing 2002, vol. 10(5), pp. 293–302

[8] T. Lidy, A. Rauber, A. Pertusa, J.M.I. Quereda, *Improving genre classification by combination of audio and symbolic descriptors using a transcription systems*, in: *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR), Vienna, Austria, 2007*, 2007, pp. 61–66

[9] A. Gungormusler, N. Paterson-Paulberg, M. Haahr, *barelyMusician: An Adaptive Music Engine For Video Games*, in: *Audio Engineering Society Conference: 56th International Conference: Audio for Games*, London, UK, 2015

[10] D. Williams, A. Kirke, J. Eaton, E. Miranda, I. Daly, J. Hallowell, E. Roesch, F. Hwang, S.J. Nasuto: *Dynamic game soundtrack generation in response to a continuously varying emotional trajectory*, in: *Audio Engineering Society Conference: 56th International Conference: Audio for Games*, London, UK, 2015

[11] D. Williams, V. Hodge, L. Gega, D. Murphy, P. Cowling, A. Drachen, *AI and automatic music generation for mindfulness*, in: *Audio Engineering Society Conference: AES International Conference on Immersive and Interactive Audio*, York, UK, 2019

[12] A. Engelbrecht, *Computational intelligence: an introduction*, John Wiley & Sons, 2009

[13] J.-P. Briot, G. Hadjeres, F.-D. Pachet, *Deep learning techniques for music generation*, Computational Synthesis and Creative Systems, Springer, Cham, 2019