

History of Algorithmic Music Generation: From Mozart to Modern AI

Jingyang Li

*Engineering School, Rensselaer Polytechnic Institute, Troy, USA
Lij53@rpi.edu*

Abstract. Algorithmic Music Generation is an old topic. From very basic systems of controlled randomness to advanced data-driven systems. This paper summarizes the general history of algorithmic music generation, as well as three key eras: the random combination era, the rule-based algorithm generation era and the deep learning era. Mozart's Musikalisches Würfelspiel (meaning musical dice game) is the first stage. This is a simple example of using predefined music materials for music generation, and random selection. The second stage was ushered in with the advent of electronic computers and algorithms such as Markov-chain and stochastic modeling. In this period the musical elements of pitch, rhythm, harmony and orchestration became formalized as parameters for computation. Even so, they were still limited in their creative abilities by rules set forth explicitly by humans. Third stage is based on deep learning. In this time period, algorithmic music generation shifts to data-driven modeling. For systems that learn directly from large amounts of data, musical patterns are learned using transformer architectures, diffusion models, and cross-modal alignment techniques, and the generation of symbolic music or high-quality audio is based on musical or natural-language inputs. In this review the mathematical and computational foundations are explored for each era, along with the shift in the role of humans to computational systems. Additional difficulties, such as long-range structural coherence, controllability, and the assessment of originality are also discussed.

Keywords: music generation, algorithm with rules, deep learning

1. Introduction

Music has been regarded as a magical and individual form of expression of human emotions. But in this case, far from what people might think, music's form is pure mathematics. The pitches, the ratio of the note lengths, or the mathematical patterns of the harmonic progressions, are all clearly revealing the deep relation between art and science. This has resulted in the concept of formalizing, standardizing and even algorithmizing the musical creation process. These ideas and experiments have demonstrated all the history of western music. In 1787, Wolfgang Amadeus Mozart created "Musikalisches Würfelspiel"(Meaning "musical dice game"), one of the most significant memorials to this long journey. In this work Mozart divided the Minuet into various musical parts and then decides on the sequence of those parts by the random roll of the dice. This creative technique enabled composers to compose an infinite number of variations without having to have a lot of

knowledge in composition. This creative activity was a step more towards a game than a true creation in the Enlightenment period, but it nonetheless harbored the two fundamental principles of algorithmic music generation: a set of well-specified rules and controlled randomness.

After Mozart, the concept of algorithmic music was gradually developed in some experiments and only became a sound concept with the arrival of the electronic computer in the middle of the 20th century, which created a completely new impetus. In 1957, Max Mathews wrote the program Music IV which allowed the first computer to synthesize sounds, and automatically create scores. This means the shift from purely theoretically deduced music to the digitally computed music. At the same time, composer Iannis Xenakis applied systematically the theory of probability and stochastic processes in the structure of his large-scale orchestral compositions, abandoning the traditional motivic development. The paradigm of algorithmic music generation during this period is roughly summarized in the term "rule-driven": the composers first lay down mathematical models, pitch constraints and rhythmic logic and the computer is then a computational executor of the conditions the composers set. The results produced had formal novelty, but never lacked stylistic unity and expressive richness because of the simplicity of the rules formulated by the composers.

The advent of deep learning has revolutionized the approach to algorithmic music generation in the 21st century. With the tremendous amount of digital music information and the greatly increased computing power of graphics processing units (GPUs), the music generation paradigm has made the transition from rule to data. Melody continuation and multi-part arrangement have been addressed one after another using recurrent neural networks (RNNs), variational autoencoders and generative adversarial networks (GANs). Recent research efforts have seen the emergence of large trained models, like OpenAI's Jukebox and Google's MusicLM, capable of producing detailed and intricate pieces of music by leveraging text-to-music instructions, or style tags to produce music that has a strong stylistic similarity to the input. No longer explicit rules set by composers. Rather, the models learn high dimensional probability distributions of pitch sequences, rhythmic patterns and harmonic progressions from vast quantities of music data. This type of result is normally considered to be at the mid to high level of composition with respect to aural perception.

Its development of the algorithmic music, from the dice roll to the deep neural networks, is huge, and involves many technologies. The goal of this paper is to systematically review and analyze the historical development of algorithmic music generation. Three main stages will be studied: the random combination stage, the rule construction stage and the deep learning stage. Following the mathematical principles and computational logic of the representative methods in each stage, the deep changes of creative concepts behind the technological changes will be discussed. This paper will take a step back and look at the history of music, thereby creating a clear historical context for musical creativity.

2. Literature review

2.1. Dice game era

The first technical model in algorithmic music generation dates back to 1787 and is called "Musikalisches Würfelspiel" (Musical Dice Game) by W. A. Mozart. The key to this piece of work is the triple core mechanism: a library of score measures pre-established; rules of random selection; a defined musical structure. The game offers a number of written measures to be played based on dice roll (typically between 2 and 12) that represent the creation of a minute. Each time the user rolls two dice, they pick the measure correspondingly from the library, and then put the results together to form a complete binary form music piece [1].

Technically, this game can be characterized by a number of attributes, from an algorithmic point of view:

It is first discretized encoded. A measure is a pitch-rhythm vector of a fixed length, each measure containing harmonic functions and all measures follow the same tonality and cadence rules.

It may then be regarded as a random mapping table: A table of the correlation between result of the dice roll and numbers of measure, is explicitly written. It is an index and the output is the result of a table look-up and not a result of logical operations.

The great amounts of possibilities of combinations is also an important point. There are 11 combinations for each of the 16 chords, although rhythmic links and chord sequences place a tacit limit.

This method cannot be applied if the structural constraints are not applicable. This isn't a totally random game. Instead, the musical form is broken up into a number of slots and each slot is restricted in the range of tones (tonic, dominant etc.) in order to guarantee that the phrases produced are harmonically well connected.

The technical part of this work is the strategy of a combination of prepared materials and random indexes, which breaks down the creative decision into two autonomous processes: composition of materials and selection of materials. While there are no calculations or adaptive rules, it did manage to break down the musical structure and also to provide the simplest operational model of algorithmic music generation, namely that of generating infinitely many variations from a finite number of materials.

2.2. Algorithm generation era

During the middle of the 20th century, the development of electronic computers gave an algorithmic jump in making music. The generation shifted from hand-looked-up tables to programmable computation. In 1957, Max Matthews at Bell Labs invented the Music I program, which for the first time was able to generate music in scores using the computer.

The program had two steps of workflow. The first one is the set-up. By digital encoding, note parameters (pitch, duration, volume, timbre) are entered into a note table, each line of which represents an occurrence of a note. The second step is the rendering. In this step, the program is able to calculate the digital audio signal point by point, based on the samplings per second of the audio signal and the wave function of the audio signal, then output the calculated result as audio file. The unit generator architecture was also introduced in Music IV and the later versions. Modularizes signal processing units like oscillators, envelope generators or filters into freely connectable components. The separation of data and process in music generation was realized with introducing the note sequences, and the synthesis algorithms were separately controlled by users who wrote their own score files and orchestra files [2].

In 1957, however, the Illiac Suite String Quartet, written by Lejaren Hiller and Leonard Isaacson, was completed, thus marking another avenue for symbolic algorithmic composition. The computer used in this work was the Illiac I and the basic model was in the form of Markov chains. It's technique can be summarized in a few steps. First, a certain string quartet score is used to statistically derive a matrix of transition probabilities for pitch, duration and interval. Next, based on the state value of the current note, a random number generator selects the transition direction of the next note by using the random number generated, and then repeat the process to obtain a sequence. Lastly, filtering and correction of the created sequence is performed according to the predefined counterpoint rules and harmonic restrictions. The key concept of this approach is that the music composition is a state transition case, where the initial state, state transition probability matrix and

random seed are all combined to determine the output of the case, and the computation process involves only two basic operations, probability sampling and rules verification [3, 4].

On another side, Iannis Xenakis systematically brought in advanced probability theory and stochastic processes in the production of large scale orchestra works. "Metastasis" (1953-54) is Xenakis' masterpiece, which was structurally generated from stochastic calculus and Poisson distribution model. He manually, with computer, calculated the sound blocks with probability distribution functions for the pitch density, entry time and duration of each instrumental part. The method of Xenakis is distinct from that of Mathews, his pitch level synthesis, and Hiller, his symbol sequence generation. It doesn't determine the pitches to be used for each note, it only indicates the distribution range and trend of the pitch set with statistical parameters. The music generated is a continuous flow at the macro level, and retains some randomness at the micro level. This technical approach is parametric random generation, that is, the controlled variable is changed from concrete notes to probability [5].

In general, algorithmic music generation technologies of this period, three paradigms of generation had already taken place: direct synthesis (Mathews) based on digital signal processing, state transition (Hiller & Isaacson) driven by stochastic processes, and statistical modeling (Xenakis) using probability distributions as control parameters. These are the three methods that will correspond to the modelling methods of the signal layer, symbol layer and the structural layer respectively. The amount of computing technology was minimal by today's standards, but their approach to using formal methods to represent musical elements – abstracting the pitch, rhythm, harmony and orchestration into controllable parameters – laid the methodological groundwork for later data-driven paradigms.

2.3. Deep learning era

In the third decade of the 21st century, the ascendancy of deep learning technology has placed algorithmic music generation in a new level. The key paradigm shift of this era is that generation rules are learnt via neural networks using huge volumes of data rather than written by a human. According to the distinction of the granularity of the models, current mainstream methods can be classified into three technical levels.

The first level is the symbol level. For tasks that generate symbolic music data like MusicXML and MIDI, the transformer architecture or its variants have become the norm. At its heart, lies self-attention and positional encoding. In particular, the music sequence is considered as a time series of symbols, and the dependency between any two symbols from the sequence is captured by the self-attention layer, which could be a good solution to the gradient decay bottleneck of Recurrent Neural Networks (RNNs) handling long-range dependencies [6]. For instance, in the case of Google Magenta's Music Transformer, the conditional probability distribution and learn this distribution in an autoregressive fashion during the training phase, given its encoder-decoder architecture is effective. Specifically, it predicts a probability distribution of the next note, using the preceding notes it created in the generation phase, and then, it generates the whole sequence by successive sampling in the inference phase. In order to further enhance the generation quality, the work of [7] added a relative position attention mechanism, which means that the absolute position of each note in the generated sequence was no longer used and replaced by the interval between the notes, so that the rhythmic periodicity and phrase structure of the music can be better preserved. OpenAI's MuseNet takes this further by providing joint modeling of multiple instruments, which jointly encodes instrument numbers and note tuples to symphonic arrangement generation, for over 16 instruments [8].

While symbol level generation is using the symbolized information as material for the generation, audio level generation is directly using the original waveform or spectrogram as material for the generation without losing the information during the symbolization process. At present, the most typical technical approach of this type is the diffusion probability model-based generation framework. The diffusion model is based on two processes, forward diffusion and reverse denoising. It gradually corrupts the original audio signal with Gaussian noise in the forward process and learns the conditional probability of slowly denoising the signal in the reverse process (using a deep neural network), and begins inference with only Gaussian noise and slowly denoises the signal to produce new audio. Google's AudioLM and MusicLM extend this by proposing the use of a layered modeling approach. This approach is implemented by following several steps: initially, Continuous Waveforms are discretized into several layers of semantically and acoustically meaningful tokens using audio tokenization technology. Next, using transformer to model the semantic structure and diffusion model to model the acoustic details, and then the ability to generate high fidelity music based on text description is realized. This method is very near to the studio quality in terms of sound quality, reverberation and instruments timbres [9, 10].

Another milestone technology of the time is model alignment. Models with the CLIP architecture (e.g., CLAP) connect text descriptions and snippets of music using contrastive learning. After this training process, the generative model (e.g., diffusion-based AudioLDM) is able to take texts as conditional input, which then steers the generation direction in the inverse denoising, and finally generates music fragments that semantically correspond to the text description. This framework adds the "input format" – natural language, which reduces the threshold of music generation and changes the way of controlling the generated music from specific parameters adjustment to generalized natural language based interaction.

3. Conclusion

A trajectory of algorithmic music generation over more than 200 years from dice rolling to deep networks can be observed. It is the simultaneous increase in the formalization and automation in music creation. This change primarily manifests itself through three basic paradigm shifts.

The first change was in the late 18th century, when Mozart's musical dice game was created, where the concept was of preset materials + random indexes. The contribution of this stage is technical: music can be structured into independent units that can be manipulated one at a time; music can be generated by looking up pre-defined rules, thus for the first time it can be repeated and measurable (although the rule itself might be very limited and set by the composer).

The second shift dates to the mid-20th century, and involves Mathews' direct synthesis, Hiller & Isaacson's Markov chain sequence generation, and Xenakis' statistical parameter modeling. The major innovation of this phase is rules can be programmed. During this time musical elements became abstracted into numerical parameters, compositional logic became abstracted into algorithmic flows and the computer was transformed from a static look-up instrument to a dynamic logic executor. Each of these three technological paradigms (signal synthesis, state transition and statistical modeling) was able to realize a formalized closed circuit of music generation starting from the three different perspectives (audio, symbolic, and structural). But, they share the disadvantage that all rules are programmed by hand and diversity of the results achieved is very limited, based on the music skill of the coder.

The third transformation is the one that happening currently in the data-driven paradigm with deep learning. The toolbox of this period was the autoregressive prediction of transformers, iterative denoising of diffusion models, and model alignment by contrastive learning. The distinction with the

previous two transformations lies in the fact that the generation rules have been moved from being explicitly coded as mathematical equations or logical statements, to high-dimensional probability distributions learned from data. This transformation has transformed the manner in which creators and equipment connect. In this change, algorithms have gone from being passive executors to active generators.

Algorithmic music generation is grappling with a number of fundamental technical issues. An obvious problem is that of long-range structural coherence. While current models work well on the scale of local musical phrases, the growth of motifs, and thematically recapitulation, is still not easy to model effectively. Interactive and controllable generation cannot be ignored, either. The method which enables user to clearly define the direction of the melody, harmony color or emotional tone without retraining the model is not yet discovered. Last but not least, the evaluation of the originality of generated results is important, too. There are many more works in human music history for which there is training data. Determining novelty of the results generated as well as avoiding any unintended plagiarism is difficult. This calls for a framework of evaluation which goes beyond similarity indicators.

Putting it succinctly, algorithmic music generation is a history of the externalization of creative wisdom in the form of computation. Music creation is an expanding frontier with every technological breakthrough, and one that invites the reconsideration of the nature of creativity. So in an age of music generating machines, creators will need to come up with a more precise definition of creation.

References

- [1] Hedges, S. A. (1978). Dice music in the eighteenth century. *Music & Letters*, 59(2), 180–187.
- [2] Mathews, M. V., & Miller, J. E. (1981). *The technology of computer music*. MIT Press.
- [3] Hiller, L., & Isaacson, L. (1993). *Musical composition with a high-speed digital computer*.
- [4] Hiller, L. A., & Isaacson, L. M. (2012). *Experimental music: Composition with an electronic computer*. Greenwood Publishing Group Inc.
- [5] Xenakis, I. (1992). *Formalized music: Thought and mathematics in composition*. Pendragon Press.
- [6] Eck, D., & Schmidhuber, J. (2002). A first look at music composition using LSTM recurrent neural networks. In *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, 103(4), 48–56.
- [7] Huang, C.-Z. A., et al. (2018). Music transformer. *arXiv Preprint*, arXiv:1809.04281.
- [8] Sexton, T. (2023). MuseNet. *Music Reference Services Quarterly*, 26(3–4), 151–153.
- [9] Borsos, Z., et al. (2023). AudioLM: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 31, 2523–2533.
- [10] Agostinelli, A., et al. (2023). MusicLM: Generating music from text. *arXiv Preprint*, arXiv:2301.11325.