Journal of Soft Computing Paradigm (ISSN: 2582-2640)
www.irojournals.com/jscp/

Soft Computing for Music Generation using Genetic Algorithm

Akanksha Satpute[1], Mayuri Bajbalkar[2], Makarand Velankar[3], Saishwari Gurav[4], Preeti Abnave[5]

[1, 2, 4,5] Information Technology, CCOEW, SPPU, Pune, India
[3] Associate Professor, Information Technology, CCOEW, SPPU, Pune, India

E-mail: [1]akanksha.satpute@cumminscollege.in, [2]mayuri.bajbalkar@cumminscollege.in, [3]makarand.velankar@cumminscollege.in, [4]saishwari.gurav@cumminscollege.in, [5]preeti.abnave@cumminscollege.in

Abstract

Creating good music is truly a laborious task since it requires lots of effort, extended time, and many instruments. When the tune of music does not sound good after being composed with much toil, then composers must discard it, which is a hectic job. Hence, an easy way to compose the music which will require less time and less effort is required. Genetic algorithm is a possible way of searching the solution to the problem in large dimension search space. Genetic algorithm (GA), a part of soft computing in the field of music composition can be used to solve this issue. This paper proposes the use of GA for composing music and the use of fitness function to select more melodious music. In GA, for music creation, two musical segments will act as parent nodes for creating new music, and by applying genetic operators, there is a change in the music such that breaks are modified between the tunes. Music which sounds pleasing is chosen with the user's help using fitness function, and if the user is satisfied with the generated tune, then the process of generating the music is terminated; otherwise, the selected musical tune by the fitness function will act as the parent node for the next generation of musical tune. Moreover, this work explains which fitness function to be applied on the specific problem.

Keywords: Music composition, Genetic Algorithm (GA), Fitness functions, music notes, melody, soft computing

1. Introduction

Computational music can be classified into broadly 2 areas such as analysis and generation. Music analytics can have various applications such as recommendation [1], pattern

DOI: https://doi.org/10.36548/jscp.2023.1.002
Received: 25.01.2023, received in revised form: 25.02.2023, accepted: 10.03.2023, published: 21.03.2023

recognition [2], emotion recognition [3] etc. Automatic music generation can be carried out using different approaches such as Recurrent Neural Network [4], Long short-term memory [5], Generative Pre-trained Transformer [6], and Genetic Algorithms (GA) [7]. This presented work focuses on the use of GA for music generation.

Genetic algorithm is a sub part of soft computing in which different offspring are generated using parent nodes. This parent node can be anything, but in this scenario parent nodes are segments of musical notes. Music is an expressive language for interpreting some expressions. It contains notes, rhythms, pitch, scale, tunes etc. and arranging these terminologies in appropriate sequence makes proper audible music. But composing music using instruments is a hectic task for composers as they have to compose music and discard it if they don't like the created music. Genetic algorithm is suggested here. Multiple melodies can be composed by using genetic algorithms. Further, by applying proper fitness function to this generated music, good music can be selected. This can reduce the load of music composers to compose different tunes every time. This generated music can be used in any song, reels, or videos as background music. It can also be used to create samples for music databases for training of the system.

Genetic Algorithm:

Genetic algorithms are used for solving optimization problems. It is a sub part of evolutionary computing. The actual concept of genetic algorithm is that it takes one or more parent nodes and applies crossover and mutation functions to generate different populations. Next population is created by taking into consideration the best genomes generated by the previous generation. Important terms of genetic algorithm are as follows:

1.1 Crossover:

It is one of the main operations performed in genetic algorithms where characteristics of the genomes are exchanged between each other. There are two types of crossover operations: One is single point crossover and second is multiple point crossover.

1.1.1 Single point crossover:

In single point crossover, random crossover point is selected and based on that crossover point, genomes are divided into parts; later, the parts of different genomes are combined to create a new genome.

Example: Two genomes A and B are considered. A is represented as 11110000 and B is represented as 11010011. User decides to take the crossover point at the middle i.e., after the 4th binary value. Inputs are A = 11110000 and B = 11010011. After taking the crossover point, A = 1111.0000 and B = 1101.0011. (.) represents crossover point. 1111 and 0000, and 1101 and 0011 are separated for crossover operation. After this actual crossover operation is done, and the output is produced as,

A = 11110011 and B = 11010000

or

A = 11010000 and B = 00111111.

Many such genomes can be created using crossover operation.

1.1.2 Multi point crossover:

This is the case of N point crossover in which multiple random crossover points are taken in both the genomes, and then actual crossover operation (altering the genome values) is done.

1.2 Mutation:

It is one of the important functions of genetic algorithms, where the genome's bits are randomly swapped to create a new genome. This function mostly works on a single genome.

Example: A = 11010011. After the mutation A = 00101100. Not necessarily complete alteration of bits are done; some bits might be kept as it is but at least one bit of the genome is changed.

Fitness Function:

It is an eligibility function which is used to select the best genome (which acts as parent node for the next generation) out of many generated genomes for the next generation which are generated using crossover and mutation functions.

2. Literature Review

In the literature, there are several studies about music composition using evolutionary algorithms, music generation using GA, and fitness function. D.T.V. Dharmajee Rao et al., [8]

proposed music generation using genetic algorithm, which explained Genetic algorithm, crossover and mutation operations and how these operations are performed on musical notes. The paper explained various fitness functions like rating -based fitness function, knowledge -based fitness function, Machine Learning -based fitness function and generative genetic algorithm.

In rating -based fitness function, rating is given by the user to each generated population. This fitness function is the perfect choice to come up with each user's perspective about each generated genome but giving rating to each generated offspring becomes hectic after some time. So, rating based fitness function is not actually a good choice for large scale problems as it consumes user's time as well as requires manual help to come up with good offspring. Knowledge -based fitness function requires the user to have some knowledge of music for finding the fittest offspring. This fitness function is also limited to the specific types of music because a single user cannot be aware of each and every type of music. Machine Learning -based fitness function uses some machine learning algorithm to find the fittest offspring generated and that offspring is used as a parent node for the other generation. This fitness function requires more time for training and testing.

M. Marques et al., [9] proposed music generation using genetic evolutionary algorithms, which explained crossover, mutation operations, how these operations are performed on the musical octave and crossover mutation priorities. As known, music can be different if the sequence of crossover and mutation is changed; so, which operation should be prioritized is explained in the paper.

Iyad Abu Doush et al., [10] proposed automatic music composition using genetic algorithm and artificial neural network. Genetic algorithms were used for generating music and Artificial Neural Network for detecting good or bad musical patterns. Using genetic algorithms and tournament selection (fitness function), the system generated 90% of musical patterns with a high fitness rate. But by using ANN, the system generated musical patterns with 83% success rate and 16.7% error. But using ANN, the time required to generate musical patterns is less.

B.Vijay Kumar et al., [11] proposed music generation using genetic algorithm, which explained how genetic algorithms can be applied for creating music. Each step of the algorithm, from taking input, applying the algorithm and what will be the final output, has been elaborated clearly. Everything is well explained except the type of fitness function used and why that specific fitness function was used.

Sanjay Majumder et al., [12] proposed music recombination using a genetic algorithm, which explained how a genetic algorithm is used to create music automatically when two musical files are given as input. Here, features of both the files were extracted automatically like on the basis of MIDI velocity, and a list MIDI notes on events was compiled for analysis. Then the best music is presented to the users, and if users are not satisfied with the generated music, then the best selected musical file will act as the parent node.

Paper [13] "A Survey on Artificial Intelligence for Music Generation: Agents, Domains and Perspectives" discussed the agents which take part in the music composition process like dataset, model, algorithm, interface, users, and compositions. Furthermore, the human process to compose music depends more on cognitive level such as analysis and AI process to compose music depends more on training the model to have an accurate output. Evaluation techniques can be of two types: subjective and objective. Subjective evaluation depends on user experience with the music. But this technique has a drawback that each model gets evaluated by different people, hence the results are not reproducible. On the other hand, objective evaluation depends more on feature vectors (measuring each parameter of music).

A. E. Eiben et al., [14] proposed Genetic Algorithm with Multi-parent Recombination, where more than two parents are used for generating music, crossover, and mutation function applied on the parent nodes. The paper explained gene scanning techniques like U-scan, occurrence -based techniques like OB-scan and fitness -based scanning like FB-scan. The paper also explained better algorithms in different situations. In some cases, the more parents the better performance, while in some cases, it decreases as the parent number reduces. According to requirements, users have to select appropriate scanning techniques.

3. Methodology

Genetic algorithms work for finding solutions over a large dimension space. Using genetic algorithms in the field of music composition, multiple musical tunes can be created and based on the fitness function melodious musical tunes are captured. Using GA, multiple musical tunes are created by using two musical tunes as the parent node and when genetic operators are applied on the parent node then almost every combination even with slight variations are generated. These variations create a number of musical tunes which are difficult for music composers to compose. Every possible combination of music to create and to find melodious music out of that, is a tedious task which is nearly impossible for music composers to compose. So, when composers want to compose musical tunes by combining two different

musical tunes then a genetic algorithm is used where composers can listen to every small variation in the music. But the difficult task in composing music using GA is to apply proper fitness function which will be suitable for the particular music.

In the proposed approach, rating based fitness function i.e., user interactive fitness function is used because every user loves different music and user's choice varies from user to user. One user may like slow musical tunes while the other may prefer hip-hop music which is quite fast and has a high pitch. So, any other fitness function other than user rating cannot determine such a huge variation in finding fittest music out of many generated musical tunes. In rating -based fitness function, every user gives a different rating to the musical tunes between 0 to 5 (this can be 0 to 10 or 0 to 100, but the proposed system uses 0 to 5) based on their opinion about the particular musical tune.

Figure 1 explains the working of the genetic algorithm on the musical notes along with an example in which only 3 child population are generated from two inputs. However, there can be many such populations and that requires changing the crossover/mutation function.

Algorithm has taken two musical notes as input parameters, say A and B. In the above fig.1, inputs are represented in binary. Then a single point crossover function is applied to input A and B, where crossover point is taken at the middle of the notes. In the next step, three generated populations are shown. After that, rating -based fitness function is applied to the generated population where the user is allowed to rate the generated population.

First two populations are selected for the next process, as they have the highest rating out of the three populations. These two populations will act as input A and B. In the next step, if the user is not satisfied with the previously generated music and wants to create a new musical generation, the process in repeated. Otherwise, if the user is satisfied with the generated music, then the process of generating music will be stopped.

Here, rating -based fitness function is chosen in such a way that, everyone loves to hear different music of many different genres, and any automated function cannot determine how a specific person will react to a particular music. Therefore, to consider every user's perspective, rating -based fitness function is a good choice where a particular person will rate a generated music based on his choice of listening to the music

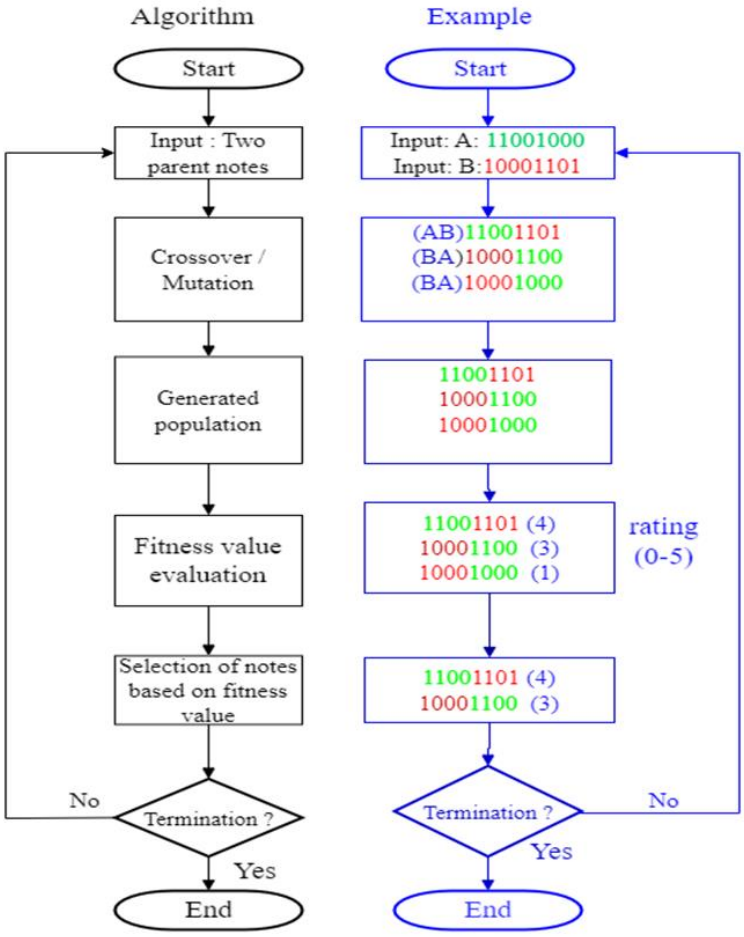

Figure 1. Flowchart for genetic algorithm

3.1. Genetic Algorithms in music

Music has various parameters, which include scale, pitch etc., to determine its musical quality. To create music using a genetic algorithm, two musical segments containing musical notes are taken as input which act as the parent nodes initially. Along with musical segments, scale and pitch are taken as the inputs. The musical notes include [“C”,“D”,“E”,“F”,“G”,“A”,“B”] etc. [13]. This is also called musical octave. There are various scales such as major, minor, etc. and pitches are represented like A4, B4, E5, F6 etc.

When input is given, actual implementation of the genetic algorithm starts. Here, GA is applied on the musical notes (where musical notes are represented in binary internally). At the first point, mutation operation is performed on a single musical segment, where flipping of some bits in a musical note is done. After that, crossover operation is performed by taking a random crossover point in between two musical segments and then altering the genome of both

the musical segments. By performing this crossover operation, a new generation of musical population is created by two parent nodes.

After the new population is created, it is shown to the user. As a rating -based fitness function is used, the user will give the rating to the generated population based on the choice of the particular user. Then two populations with the highest rating are shown to the user and if the user is satisfied with this generated population, then the process of generating the population is stopped; otherwise, a new population is generated by using two musical segments with highest rating which act as parent nodes. When crossover operation is applied on musical segments, there is some change in the created music such as, actual pause in the music is also modified or extra pause is added because of selecting random crossover point. In this implementation of the genetic algorithm for generating music, some musical parameters like scale and pitch are applied to the new population as it is taken from the users.

3.2. Working of each step of genetic algorithm for music creation

Step 1: Input is the two musical segments A and B, where A can be “ABC” and B can be “EFG”. These are internally represented in binary and some more musical parameters like scale, pitch etc. are taken as input.

Step 2: Genetic operators like crossover and mutation are applied on this musical segment. At first, mutation is performed and then crossover is performed for great variation in the music. Output of this step is multiple generated populations like “ABE”, “ABF”, “AEF”, “BCE” etc.

Step 3: All these populations are shown to the user.

Step 4: User is asked to rate each generated musical population between 0 to 5. Output of this step is population with user rating.

Step 5: Highest rated musical tunes are selected out of many generated musical tunes, and they are again shown to the user. Output of this step is the two highest rated musical segments.

Step 6: If the user is satisfied with these two highest rated musical tunes, then the process of generating new population is stopped. Otherwise, these two will act as the parent nodes for generating new musical tunes and the same process from step 1 is repeated till the

user is satisfied with the generated tunes. Output of this step is a melodious musical tune which the user likes.

3.3. Difference between proposed model and existing models

Main difference between the proposed model and existing models is that, the proposed model has used user interactive i.e., rating -based fitness function, unlike other models which have applied different fitness functions like machine learning -based fitness function i.e., automotive fitness function, knowledge -based fitness function etc. The drawback of these other fitness functions is that each user has a different perspective about different types of music, and when an automotive fitness function is applied, some musical tunes which the user may like can also be discarded and because of that, the user may not get the tune which he likes. Secondly, the proposed work is easy to apply and implement, as the user rating is taken as the fitness value while other fitness functions require more knowledge to implement, such as, to implement machine learning -based fitness function, knowledge of different machine learning algorithms is required, and hence these types of fitness functions are difficult to implement.

4. Discussion

The study on the previous papers shows the working of a genetic algorithm and how actually it is used for composing the music. Paper [8] explained various fitness functions used in GA, their advantages, and disadvantages. But the article has not explained which fitness function is applicable in case of music generation and why it is used. In another article [10], the automatic music creation using ANN is explained. In this work, music is composed using GA, but ANN is used for finding the fittest musical tune. This automatic system has a procedure for selecting melodious music out of many generated music; hence, some musical tunes are automatically discarded and there might be music which the user may like to listen to, in that discarded music. Article [14] also discussed music composition using GA. The difference is that the proposed paper uses more than two parents instead of typically using only two parent nodes for creating the next population.

5. Conclusion

Good music can be created using genetic algorithms where two musical segments will act like the parent nodes. Applying genetic algorithms to the musical segment and generating

the next population is a relatively easier task than finding suitable fitness function and applying it to the generated populations. Different fitness functions, their advantages and drawbacks are explained in this paper. By using this proposed method, it is easier to choose proper fitness functions in different cases. For instance, if the user wants some automation to select next input parent nodes, then a machine learning -based fitness function is used, or if users have some musical knowledge, then knowledge -based fitness function is used. Genetic algorithm (GA) is good for getting optimized results. Here, GA is used for getting melodious musical tunes, and fitness function determines how melodious music is created. This paper has explained various fitness functions like knowledge -based fitness function, machine learning -based fitness function etc. but each one of them has some drawbacks and the fitness function which requires user intervention is found out to be more accurate than any other in case of music, but it needs manual contribution of humans for selecting the best melody. Music composition using GA is very helpful for anyone who wants to compose music using two different musical segments. Music composition using GA is very helpful for music composers as it requires less time and less effort to compose the music. In the future, it can be used by professional composers to compose movie songs, various songs of new genre etc.

6. References

[1] Velankar, Makarand, and Parag Kulkarni. "Music Recommendation Systems: Overview and Challenges." Advances in Speech and Music Technology: Computational Aspects and Applications (2022): 51-69.

[2] Velankar, Makarand, Amod Deshpande, and Parag Kulkarni. "Melodic pattern recognition in Indian classical music for raga identification." International Journal of Information Technology 13, no. 1 (2021): 251-258.

[3] Velankar, Makarand, Rachita Kotian, and Parag Kulkarni. "Contextual mood analysis with knowledge graph representation for Hindi song lyrics in Devanagari script." arXiv preprint arXiv:2108.06947 (2021).

[4] Dua, Mohit, Rohit Yadav, Divya Mamgai, and Sonali Brodiya. "An improved RNN-LSTM based novel approach for sheet music generation." Procedia Computer Science 171 (2020): 465-474.

[5] Minu, R. I., G. Nagarajan, Samarjeet Borah, and Debahuti Mishra. "LSTM-RNN-Based Automatic Music Generation Algorithm." In Intelligent and Cloud Computing: Proceedings of ICICC 2021, pp. 327-339. Singapore: Springer Nature Singapore, 2022.

[6] Sarmento, Pedro, Adarsh Kumar, Yu-Hua Chen, C. J. Carr, Zack Zukowski, and Mathieu Barthet. "GTR-CTRL: Instrument and Genre Conditioning for Guitar-Focused Music Generation with Transformers." arXiv preprint arXiv:2302.05393 (2023).

[7] Gupta, Sumita, Rana Majumdar, and Sapna Gambhir. "An Approach to Generate Music using Genetic Algorithm." In 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 1-5. IEEE, 2022.

[8] Dr .D.T.V. Dharmajee Rao,Tangudu Deepthi, Kotni Sai Bharath Kumar,Kella Tarun Kumar, Meesala Pravallika. “MUSIC GENERATION USING GENETIC ALGORITHMS”. International research journal of modernization in engineering technology and science, volume:04, 2022.

[9] M. Marques, V. Oliveira, S. Vieira, AC Rosa.“Music composition using genetic evolutionary algorithms”. IEEE, 2000

[10] Iyad Abu Doush, Ayah Sawalha. “Automatic music composition using genetic algorithm and artificial neural network”. Malaysian Journal of Computer Science, Vol. 33(1), 2020.

[11] B.Vijay Kumar , Sai Hruday P , Farzan Khan , Gouthami G. “Music Generation using Genetic Algorithm”. ISSN, 2022.

[12] Sanjay Majumder , Benjamin D. Smith. “Music Recombination using a Genetic Algorithm”. ISSN, 2019.

[13] Carlos Hernandez-Olivan, Javier Hernandez-Olivan1, and Jose R. Beltran1.“A Survey on Artificial Intelligence for Music Generation:Agents, Domains and Perspectives”. IEEE 2022.

[14] A.E. Eiben, P-E Raué and Zs. Ruttkay. “Genetic algorithms with multi-parent recombination”. Springer-Verlag 2019