



# A combination of multi-objective genetic algorithm and deep learning for music harmony generation

Maryam Majidi<sup>1</sup> · Rahil Mahdian Toroghi<sup>1</sup>

Received: 3 June 2021 / Revised: 8 February 2022 / Accepted: 31 May 2022 /  
Published online: 25 June 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Automatic Music Generation (AMG) has become an interesting research topic for many scientists in artificial intelligence, who are also interested in the music industry. One of the main challenges in Automatic Music Generation is that there is no clear objective evaluation criterion that can measure the music grammar, structural rules, and audience satisfaction. Also, original music contains different elements that should work together, such as melody, harmony, and rhythm; but in the most of previous works, Automatic Music Generation works only for one element (e.g., melody). Therefore, in this paper, we propose a Multi-Objective Genetic Algorithm (MO-GA) to generate polyphonic music pieces, considering grammar and listener satisfaction. In this method, we use three objective functions. The first objective function is the accuracy of the generated music piece, based on music theory; and the other two objective functions are modeled scores provided by music experts and ordinary listeners. The scoring of experts and listeners separately are modeled using Bi-directional Long Short-Term Memory (Bi-LSTM) neural networks. The proposed music generation system tries to maximize mentioned objective functions to generate a new piece of music, including melody and harmony. The results show that the proposed method can generate pleasant pieces with desired styles and lengths, along with harmonic sounds that follow the grammar.

**Keywords** Automatic Music generation · Polyphonic Music pieces · Harmony · Multi-objective genetic algorithm · Bi-LSTM

## 1 Introduction

By growing the multimedia contents, attracting audiences is becoming more difficult day by day. In the meantime, music may be one of the popular options because of its ability to evoke emotions.

---

✉ Maryam Majidi  
Mry20mj@gmail.com

Rahil Mahdian Toroghi  
Mahdian.t.r@gmail.com

<sup>1</sup> Faculty of Media Technology and Engineering, Iran Broadcasting University, Tehran, Iran

Digital advances have also changed the shape of music composing. In recent years several methods have been introduced for Automatic Music Generation (AMG) using computers and algorithms. The machine, like humans, can generate music pieces using the rules and music elements. Also, it can help musicians and composers to make new ideas or to generate effective contents.

AMG models can be classified into these groups: Markov model-based methods [3, 6, 9, 24, 28–30], approaches based on music rules and regulations [4, 8, 18, 20], neural network-based models [1, 2, 5, 11, 15, 21, 22, 25, 26, 32, 35], methods based on evolutionary optimization algorithms, and population-based [12, 16, 17, 19, 31, 34], and algorithms based on local search [7, 14].

However recently, the neural networks, and evolutionary algorithms have been more widely used in the AMG literature, such as conditional rhythms generation of drum sequences with neural networks [21], and using a Hierarchical Recurrent Neural Network (HRNN) for melody generation [35] or combining two types of music generation models, namely symbolic, and raw audio models based on the WaveNet architecture [22]. Furthermore, the idea and innovative scientific contribution of study [34], design a computer program called GenDash that employs evolutionary computation in the composing music and the MetaCompose music generator uses a novel combinatorial evolutionary technique with Feasible-Infeasible Two-Population (FI-2POP) for effective melody generation [31].

Most of the mentioned methods focus on generating melody and rhythm, and do not consider harmony, or generating of appropriate chords. Also, most of the methods do not consider human taste or satisfaction in the generation process. Among the previous works, the study of Farzaneh and Mahdian [12], proposes an evolutionary model interacting with humans as an objective function. But their evaluations are only based on the opinions of regular listeners. Also, it doesn't consider the grammar.

In this paper, we use a multi-objective evolutionary algorithm for harmony generation. This method interacts with both regular and expert listeners. Furthermore, we use music grammars as the third objective function. So, the proposed method, in each execution tries to maximize the listener's satisfaction (regular and experts) as well as the accuracy of the generated music piece according to the music theory and grammar. In order to remove the listeners in the AMG process, and completely automizing the system, we have modeled the human scoring using two Bi-directional Long Short-Term Memory (Bi-LSTM) neural networks (one for regular listeners and another for expert listeners).

So, the contributions of the authors are:

- Introducing a new melody generation system with chords (harmony) based on an evolutionary multi-objective algorithm.
- Combining the human and non-human (music grammar) objective functions.
- Considering both expert and ordinary listeners as separate objective functions.
- Introducing a polyphonic music generation system, including the composition of melody, harmony, and rhythm in desired styles and lengths.
- Modeling audience behavior in understanding the beauty of music by a Bi-LSTM model, and using it as an evaluation function.

The rest of the paper is organized as follows:

Section 2 provides the state-of-the-arts in AMG. Section 3 explains Genetic Algorithm (GA). Section 4 introduces the proposed AMG system. Section 5 reports the experimental results, Section 6 discusses the work, and Section 7 concludes the paper.

## 2 Related works

In this section, we review the literature for AMG. Some of the early works of melody generation used random models. In 2010, Davismoon and Eccles were among the first researchers to introduce melody generation as a hybrid optimization problem with a Markov model integrated into the objective function [9]. Herremans and Chuan took a different approach, inspired by linguistics. They used neural networks to evaluate the ability of semantic vector model patterns (word2vec) to record music text and semantic similarity [15].

In recent years, some researchers have shown the effectiveness of using techniques such as optimization and deep learning over earlier methods such as Markov models [7]. More sophisticated deep learning models such as recursive neural networks have become popular. This trend is partly due to the fact that such models can learn complex relationships between notes with respect to existing components. Some of these models make it possible to create music pieces by repeating patterns and structural concepts. Franklin created a recurrent neural network with Long Short-Term Memory (LSTM) that generates melody or monophonic music pieces [13].

Harmony or polyphonic music is a process of combining individual voices that are analyzed by hearing them simultaneously. The science of harmony often refers to the vertical aspect of music and is distinct from horizontal and melodic motion. The purpose of using harmony is to accompany the melodies by considering the relevant rules that lead to the creation of polyphonic music. Among the works done in the field of harmony, we can mention the synchronization of melodies using the evolutionary optimization algorithm [10]. Chorale generation is one of the most popular works of music generation in terms of harmony and produces very structured music. The most common form is the generation of three sounds, which are used to harmonize a certain melody. The Bach-in-a-Box system provided by McIntyre harmonizes a melody created by the user [23].

Nakamura et al. created a system that automatically generates sound effects and background music for short videos. The melody, harmony, and rhythm of each scene are created by considering its moods, its intensity, and weakness [27]. Tuohy and Potter developed a genetic algorithm that produces playable guitar music by minimizing hand and finger movements [33].

One of the recent AMG approaches is the use of Generative Adversarial Networks (GANs) as well as reinforcement learning methods. In 2017, Yang et al. proposed a system called MidiNet, which in two separate experiments generated melody and melody with harmony by combining the GAN and Convolutional Neural Networks (CNN) [36].

Automatic Music Generation (AMG) systems have so far been either based solely on the music grammar or based on human opinions. The generated music should not only be approved by the rules of music, it should also be attractive for humans. But there is no study that considers the grammatical rules and human taste together. Also, the AMG system should be stable for all kinds of listeners. In fact, the main question of the research is: Is it possible to create an A.I.-based system to generate all the elements of music by using a modeled human scoring function in cooperation with grammar? In other words, we introduce an AMG system that is able to generate melody, harmony, and rhythm while it considers all aspects of evaluations (human tastes as well as grammar).

## 3 Genetic Meta-heuristic algorithm

A Genetic Algorithm (GA) is an evolutionary search method that is able to find optimal or near-optimal solutions. The most attractive feature of GA is its

flexibility in using different types of objective functions. The main reasons for this success are:

- Genetic Algorithm is able to solve difficult problems quickly and confidently.
- It communicates very easily with existing models and simulations.
- It is scalable and combinable.

All of these reasons can be summarized for one reason: GAs are strong. The main advantages of genetic algorithms are the following:

The most important strength of the genetic algorithm is its parallelism so that there are several starting points for solving the problem, and at one time it can search the problem space from several different directions. This increases the efficiency of the genetic algorithm in solving nonlinear problems with a large space.

Since most real problems are nonlinear, in linear problems each element is independent, and the change in one part has a direct effect on the whole system. In nonlinear problems, a change in one part may have an uncoordinated effect on the whole system, or a change in several elements may have a large effect on the system. The parallelism of the genetic algorithm solves this problem e.g., there are 20 solutions to a 10-digit linear problem, and 210 solutions to a 10-digit nonlinear problem.

Another advantage of the genetic algorithm is its Blind Watchmakers feature. That is, the genetic algorithm has no knowledge of the problems it solves. The algorithm for solving the problem shows random changes in the candidate solutions and uses the objective function to measure whether they have made progressive changes or not. This allows the algorithm to start solving the problem in a wider space, and since its decisions are essentially random, all possible solutions are open to the problem. Other advantages include good global search, easy implementation, the ability to optimize with discrete and continuous variables, and solving nonlinear hybrid optimization problems under nonlinear constraints and inequality equations.

Furthermore, in the proposed method, it is necessary to generate a new piece of music for each run. The random generation property of the genetic algorithm provides this demand.

Therefore, in the proposed method, because the population is enough, by carefully encoding the chromosomes, selecting an appropriate objective function and algorithm parameters, the genetic algorithm will be a desirable option for the core of the proposed method.

## 4 The proposed method

At the first stage, a genetic algorithm (GA1) is used to generate a wide range of music pieces (melodies with chords). The objective function of this GA considers the rhythms and similarity between generated pieces and a standard database of polyphonic human-made pieces. So, after the first stage, we have a collection of optimized music pieces. Then, two groups of expert listeners and regular listeners provide scores between 0 and 100 for each piece in the generated collection. The scoring of each group is then modeled by a Bi-LSTM neural network. Then, we run another genetic algorithm (GA2) by adding the trained networks to the objective function. In this way, the multi-objective algorithm considers both the rules and the satisfaction of human listeners in a generation. The final music generation system is shown in Fig. 1.

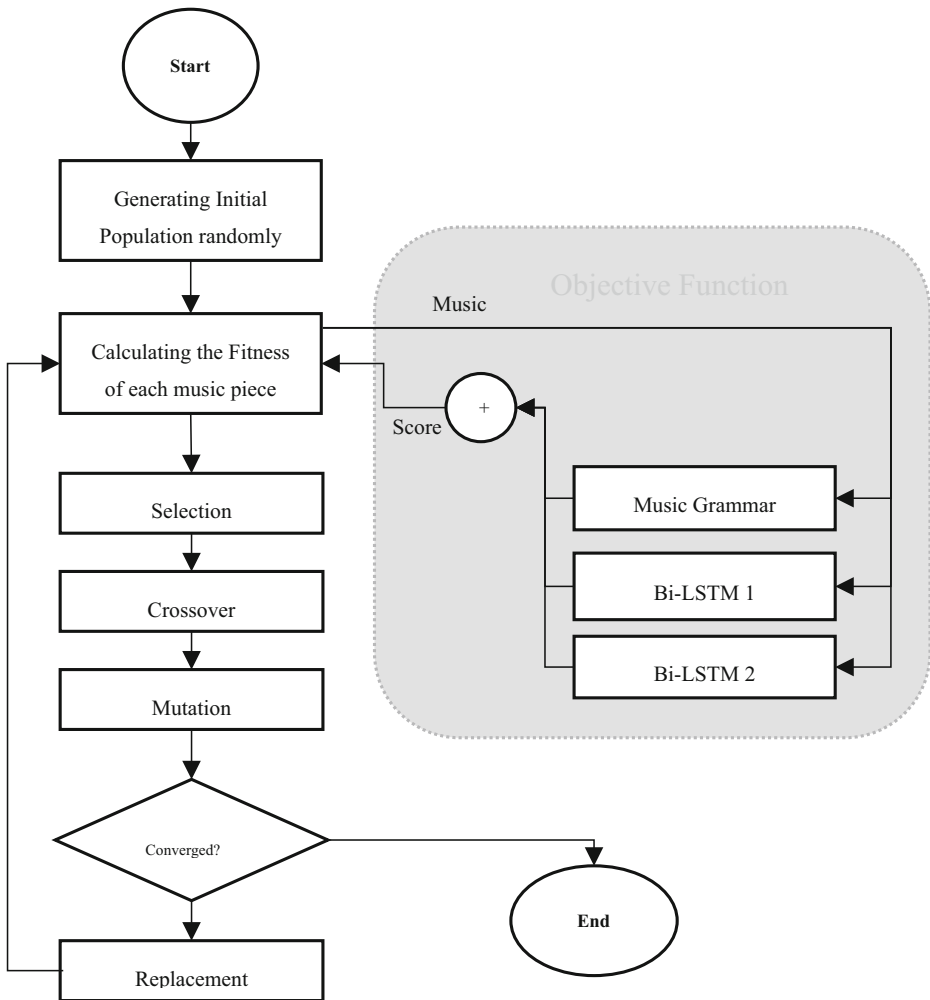


Fig. 1 The proposed music generation system

#### 4.1 Chromosome structure

Each chromosome in GA is actually a music piece. So, GA tries to optimize the music piece based on its objective functions. To generate chromosomes, we convert each music piece (including melody and chords) from ABC notation into a piano matrix.

The piano matrix contains 88 rows for 88 keys on a piano, and optional columns as the number of time intervals. Each time interval is equivalent to 1/64 of the original time unit of the music.

This matrix only contains ones and zeros. One means the presence of sound at a certain key on the piano, and zero means silence. So, according to the letter of each note in ABC notation and its octave, we specify its corresponding row and numbering the matrix elements based on the length of each note.

## 4.2 Genetic algorithm for data collection (GA1)

Using GA1, we generate a collection of music pieces (melody with chords) considering the music rules. In fact, GA1 generates music pieces as its chromosomes. At each iteration of GA1, the best chromosomes in the population are selected based on minimum violation of the rules and the maximum similarity to a human-made polyphonic music database. In other words, GA1 compares each generated music piece with a database of human-made music pieces and calculates a similarity value for them. This similarity value should be maximized. Also, GA1 considers the rules of music and minimizes the number of violations of the rules in the generated music pieces. Therefore, the GA1 starts from a random music piece and after the optimization, provides a new music piece that follows the rules and has the maximum similarity to the human-made database.

So, we use a database as a reference and compute the similarities of Bi-gram, Tri-gram, and 4-g in the generated pieces. In addition, there are costs for violating the rules so that the transmission of the notes to each other is musically sensible and a pleasant frequency interval is created. Therefore, we maximize the probability of each note occurring after another note. For chords that match the melodies, we follow the rules of chord writing and also extract repetitive combinations of simultaneous sounds from the database. In addition, to break the rhythm, a cost is considered in the objective function. Then, the fitness function is calculated from Eq. 1 which should be maximized.

$$\text{Objective Function} = \text{Score} + \frac{e}{e + \text{Cost}} \quad (1)$$

Where *Cost* is the number of rhythm and harmony violations. The constant *e* is a small value to prevent division by zero, and *Score* is the similarity value between the chromosome (generated music piece) and the human-made database:

$$\text{Score} = \frac{N_2 + 10N_3 + 100N_4 + S_2 + 10S_3 + 100S_4}{ML} \quad (2)$$

Where  $N_2$ ,  $N_3$ , and  $N_4$  are the numbers of repeated Bi-grams, Tri-grams, and 4-g in the chromosome and the database respectively (similar rows).  $S_2$ ,  $S_3$ , and  $S_4$  are the simultaneous combinations of two-voice, three-voice, and four-voice repetitive between the chromosome and the database (similar columns). In fact, the number of common combinations of consecutive and simultaneous notes between the generated music and the database is calculated, and by considering the coefficients, the similarity score of the generated piece with the human-made music is obtained.  $M$  is the number of notes in the human-made database, and  $L$  is the number of notes in the chromosome.

To consider the style of the database and transfer it to the outputs, we calculate the probability of occurrence of each note in that style, and based on these probabilities, random generation by genetics occurs. The character probability can be calculated as follows:

$$P_{n_i} = \frac{n_i}{n_{\text{total}}} \quad (3)$$

Where  $n_i$  is the number of occurrences of each character in the entire database, and  $n_{\text{total}}$  is the total number of characters in the database.

At each iteration, crossover and mutation are applied after selecting the most competent chromosomes. The pseudocode for the crossover is as follows,

```

START
    PROGRAM Crossover Operator
    READ best1 and best2
    Half a child is equal to half a best1
    Half a child is equals to half a best2
END

```

Where *best1* and *best2* are the selected chromosomes, and the *child* is a new chromosome after crossover.

The pseudocode for mutation is as follows,

```

START
    PROGRAM Mutation Operator
    READ child
    IF Mutation Rate is equal to 0.1
        FOR 1 through Number of channels
            FOR 1 through Number of child sizes
                IF the random number is less than the Mutation rate and child is
                    not equal to zero
                    SET the child equal to zero
                END IF
                IF the random number is less than the Mutation rate and child is equal to
                    zero
                    SET the child equal to Number of channels
                END IF
            END FOR
        END FOR
    END IF
END

```

Where the *channel* is the number of simultaneous sounds.

### 4.3 Bi-LSTM-based evaluating models

Each generated music piece from the previous stage is heard by two groups of listeners: experts and regular listeners. The music piece fitness is determined from the average score that listeners provide.

The Music is time-series, time-dependent and sequential. Prediction problems in time-series can lead to the solution of a regression problem. In this research, a sequence of generated music in a time frame is considered as the input of a Long Short-term Memory (LSTM) network. The LSTM model is a type of RNNs that uses a memory cell built to show long-term dependencies on time-series data. So it can learn the sequence of music as well. The output of this network is the average scores given to the pieces of music by listeners.

To simulate the way of scoring for each group of listeners, a Bi-LSTM neural network is used. Since all the musical notes are interconnected and their connection is maintained to the end, the Bi-directional LSTM recursive neural network is used, which has a memory and examines the logical connection of the notes from the beginning to the

end, and from the end to the beginning. These networks can provide the output score with the least error by solving a regression problem from input samples that have long-term dependencies. The output range is between 0 and 100. The networks receive a chromosome as a piano matrix which is a sequence and provide scores that show the fitness of each music piece.

In fact, for the training, we need music pieces as inputs and score values as outputs. After the training phase, the network will be able to predict the scores of each generated piece of music and will eventually be used as an objective function of the genetic algorithm instead of the human listeners. So, after training, networks can be used as criteria for evaluating musical sequences and their satisfaction. The architecture of the LSTM network is shown in Fig. 2.

#### 4.4 Proposed automatic Music generation system (GA2)

As can be seen in Fig. 3, GA2 for the AMG system has the same process as the GA1, except that, the fitness function, by adding human, has several objectives: Music rules, Bi-LSTM

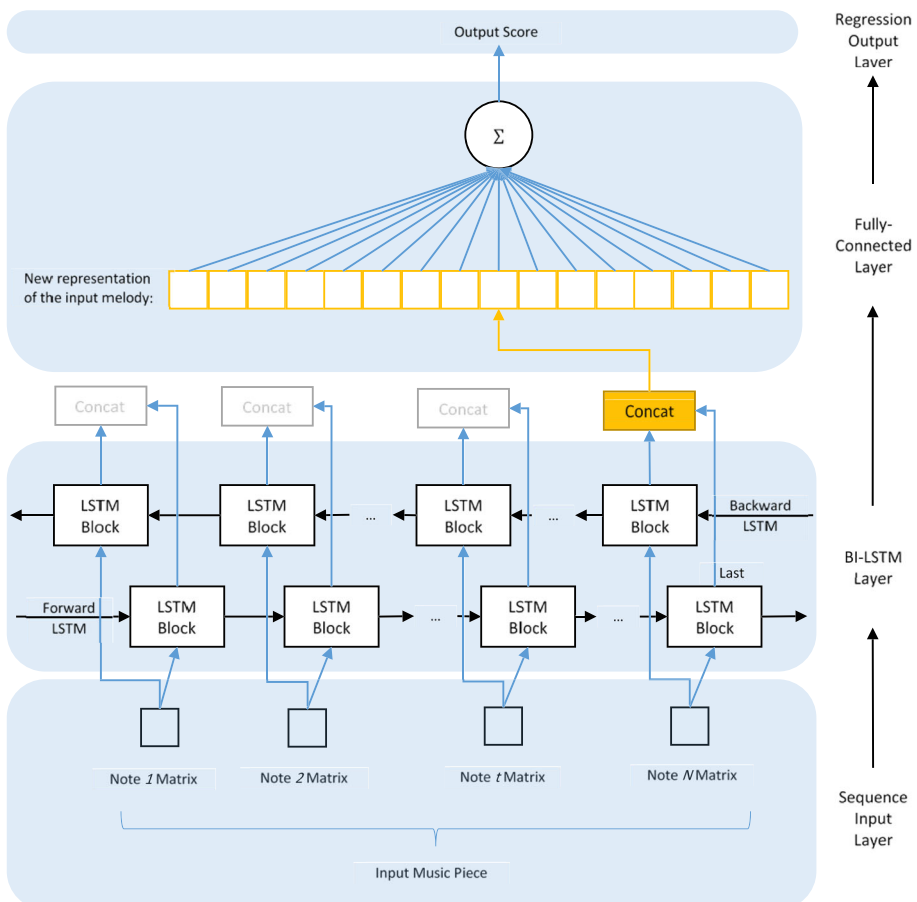


Fig. 2 The architecture of the Bi-LSTM network



trained neural network model with opinions of music experts, and the trained Bi-LSTM neural network model with opinions of regular listeners. We calculate the fitness function of the final automatic music generation system in the following equation,

$$\text{Fitness Function} = w_1X_1 + w_2X_2 + w_3X_3 \quad (4)$$

Where  $X_1$ ,  $X_2$ , and  $X_3$  are the musical grammar score, the average score of the expert listeners, and the average score of the regular listeners, respectively, provided by trained neural network models.

$w$  constants are the weights for each objective function to normalize all the objectives.

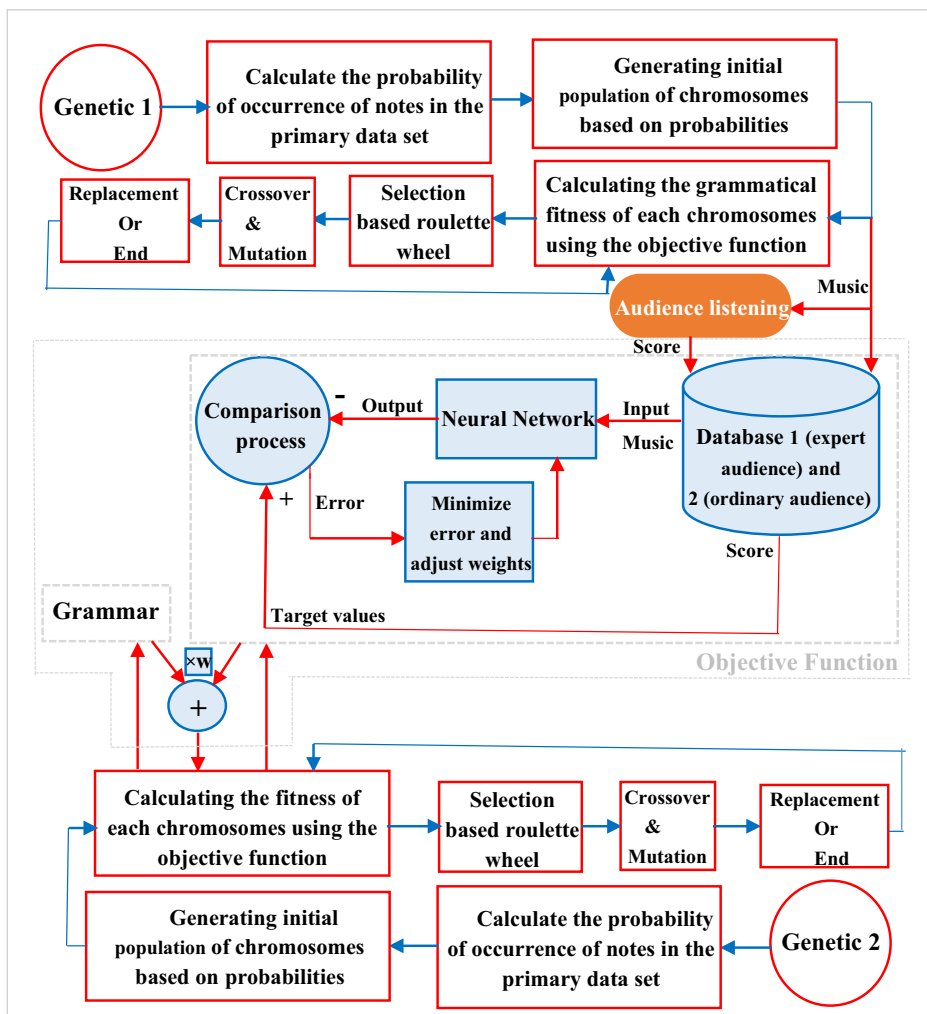
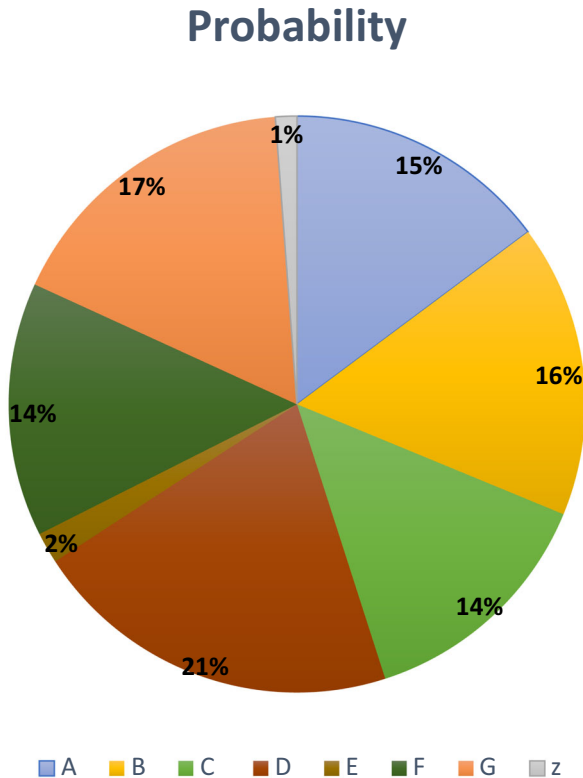


Fig. 3 The flowchart of the Music Harmony Generation System



**Fig. 4** Note's probability of occurrence in Steirar database. Musical notes are represented in the ABC notation standard

## 5 Experimental results

The proposed system is implemented in MATLAB R2018b, and all the executions have been done on a system with CPU Intel Core i7, 8 Gigabytes of RAM, and Windows10 O.S.

To provide the human-made pieces of music, we have used the Steirar database,<sup>1</sup> which contains 235 polyphonic music pieces in ABC notation. Figure 4 shows the probability of occurrence of notes in this database.

The parameters of each algorithm can be seen in Table 1. The parameter values for both GA1 and GA2, are the same for a better comparison of the results. In fact, we want to answer the question: does aggregation of objectives increase runtime and slow down speed?

In Bi-LSTM neural networks, four layers are used, and:

- The first layer in networks is actually the input layer that takes the matrix of notes (piano matrix).

<sup>1</sup> <http://abcnotation.com/tunes>

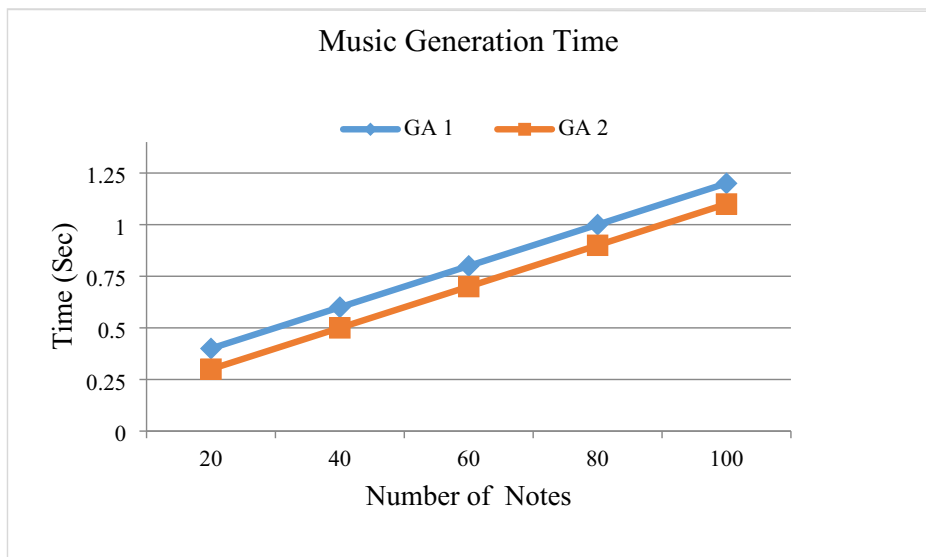
**Table 1** Parameters Settings

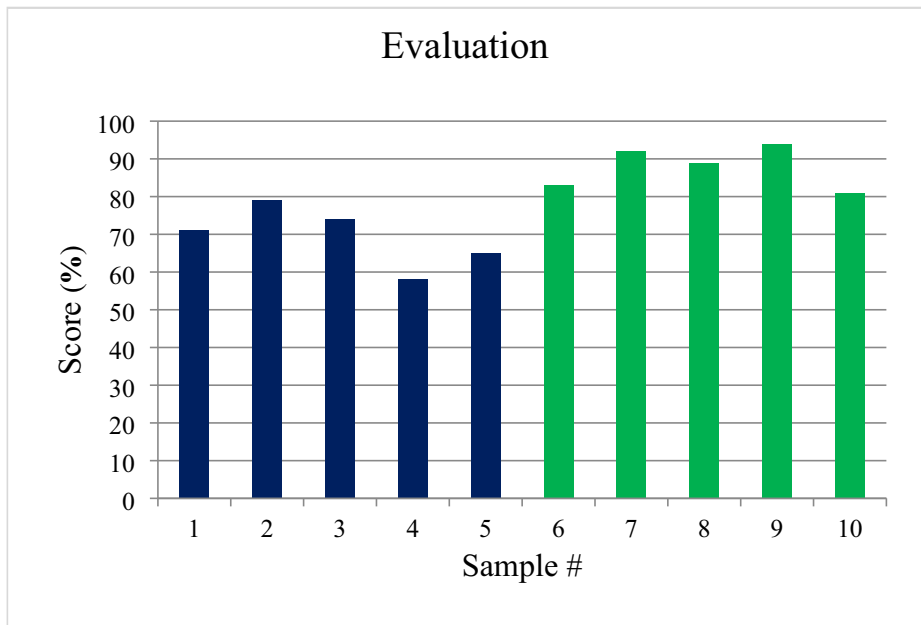
Parameter	GA 1	GA 2
Number of Iterations	3600	3600
Population Size	15	15
Crossover rate	0.5	0.5
Mutation rate	0.1	0.1
Objective Function	Grammar	Grammar & Human

- The second layer is the Bi-LSTM layer, which has 50 neurons in the hidden layer.
- The third layer is a Fully Connected layer with one output.
- The last layer is a regression layer with a cost function of the Root Mean Square Error (RMSE). This layer provides the score of the input music piece.
- The number of learning epochs is set to 5000.

After executing the GA2, we can compare the results of the two steps. Figure 5 shows the time and speed of generation to generate different music pieces with different lengths and 2 simultaneous sounds. It is observed that the GA2 is faster than the GA1. As it can be seen, having three objectives slightly increased the execution time.

In the GA2, if we used only the modeled network, convergence speed would be very high. But in this research, the combination of living objectives (listener opinions) and non-living objectives (musical grammar) has been considered. Therefore, as it was observed, at this stage, the aggregation of objectives did not increase the computation cost. In fact, adding networks to the objective function of the GA1, which was the grammar of music, did not undesirable change system performance. In addition, having three objectives, two of which are neural networks, has accelerated the convergence of the algorithm compared to before. Compatibility of objectives with each other has had a favorable result. In fact, the use of rules individually

**Fig. 5** The execution time at GA1, and GA2 vs. the number of notes



**Fig. 6** Evaluation to ensure compliance with music rules, scored by 15 expert audiences. Average scores vs. Music samples

and in combination with human satisfaction has been effective in the process of improving results. Naturally, by increasing the length of the notes in both phases, we will increase the runtime.

We also gave 5 outputs of GA1, and 5 outputs of GA2 expert and regular listeners, and asked them to give a score between 0 and 100 to the outputs. Figures 6 and 7 show their average scores. The music is subjective and the fitness of each music piece depends on listeners' tastes. The evaluation and assessment in Music Generation is a challenge. Since music is related to human emotion and should be liked by the listener, there is no specific objective or numerical criterion to determine its acceptability. Therefore, to measure the quality of generated pieces, it is better to use the opinions of the listeners.

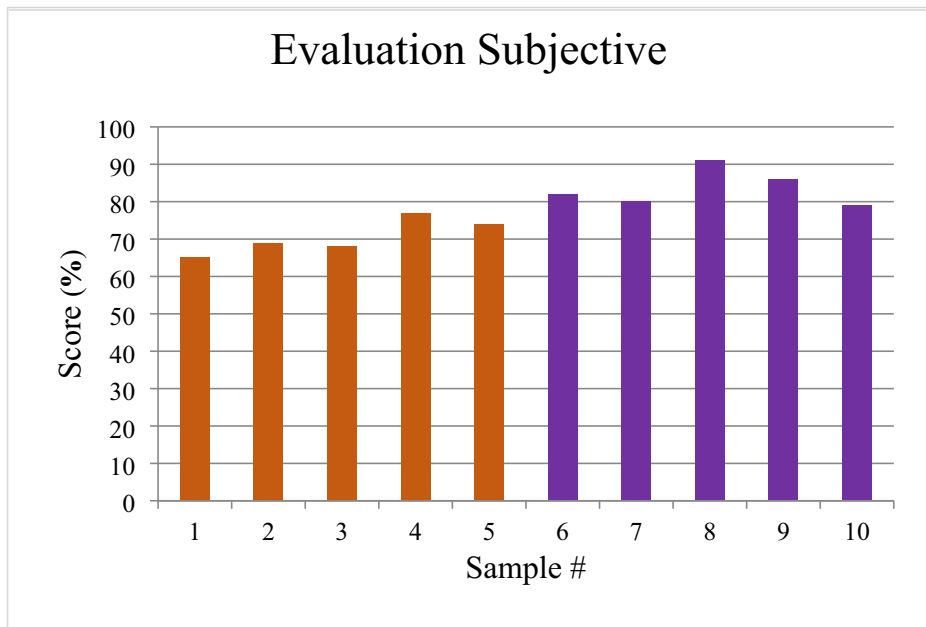
Figure 8 shows a sample output of the proposed AMG system:

## 6 Discussion

In this paper, a new model for AMG is introduced. A combination of the evolutionary algorithm and deep learning which has the strengths and weaknesses that are mentioned.

### 6.1 Advantages of the proposed automatic Music generation system

The proposed system can generate pieces of any width (number of chords) and length. Also, it is capable of generating music in a variety of styles and moods. Also, it can follow the style of a particular composer and combine styles or maintain the desired style.



**Fig. 7** Subjective evaluation of the music samples. Beauty and pleasurability of the music, scored by 15 audiences. Average scores vs. Music samples

The proposed system can compose chords (harmony) and add them to melodies to generate complete pieces of music that have so far received less attention.

The proposed system has a relatively high speed of music generation so that even with a typical personal computer, a very high number of pieces can be generated in a certain period of time.

The proposed system is able to generate pieces that are pleasant to the human listeners.

The proposed system, while satisfying the audience, also follows the rules of music (such as rhythm). In fact, the automatic music generation system is able to generate music based on a multi-objective algorithm and provide a music evaluation model based on the rules of music and the opinions of music experts and ordinary listeners.

## 6.2 Limitations of the proposed automatic Music generation system

The proposed system does not take into account the playing speed of the pieces. While the speed of playing music is very important and necessary for the listeners.

The proposed system does not consider the musical instrument. Also, in the proposed system, the performance of two or more different instruments simultaneously is not considered. While in many pieces of music the rhythm and melody may be played together, however the instrument is different. Also, sometimes if a non-piano instrument is used, two separate melodies must be generated, one for the melody and the other for the rhythm. But, they can be played with different instruments.

This method cannot generate narrative music with emotion or storyline. For example, the proposed system generates music by mentioning a sense, and the output satisfies the desired



**Fig. 8** Sample output of the proposed music generation system

sense. The proposed system can learn the sequence of notes, the probability, and the relationship between the notes from the rules in the music, and generate beautiful and complete pieces of musical pieces with the listener's satisfaction in mind. Therefore, if the proposed system can consider into account the feelings and thoughts in each piece of music, narrative music will also be generated.

### 6.3 Analysis of results

The proposed method of music generation is a combination of evolutionary computing and deep learning, with the difference that we made neural network modeling as an objective function for the evolutionary algorithm. Therefore, it may be better to compare the output results with the methods of music generation in the field of evolutionary algorithms and deep learning. Automatic music generation based on each method will have a number of weaknesses and a number of strengths.

In evolutionary methods, random changes usually occur and by considering specific parameters, duplicate outputs can be avoided. In addition, the evaluation can be done in

real-time and its feedback can be considered in the next iteration. But the outputs may not be very standard because there is no training and learning phase. Creativity in the pieces will be the result of these random generation errors.

Using some of the deep learning methods such as LSTMs for music generation, since the training is based on the previously received data, the outputs will be predictable and repetitive. So, they will most probably follow the general rules of music and generate a sequence of notes according to the learning phase. Training time will be long and we need large databases to achieve good quality.

Therefore, in this research, we have tried to consider the strengths of both methods reduce the problems of each method to some extent by combining methods and based on the opinions of the listeners. In fact, the main criterion for evaluation is the correct generation of polyphonic music based on the rules and opinions of the listeners, and the outputs should be acceptable and pleasant to listeners. Certainly, having more listeners is very effective in improving the quality of results. Even if there are numerical evaluation criteria in measuring music pieces, attracting the listener's satisfaction will be the most important factor for accepting man-made or automatic music generations.

In this paper, the main structure of the proposed method is based on an evolutionary multi-objective algorithm. Given that in the learning process, feedback from the environment is given to the generator, and correction is done based on it, this method can be considered as a Reinforcement Learning (RL) approach.

The most important contribution can be the combination of live and non-living objective functions to generate complete music pieces with all the music components (melody, harmony, and rhythm). In fact, the final AMG system is ideal for an art that does not have a number-focused evaluation because it considers all the effective factors to the pleasantness of the music. Although the goodness of a music piece depends on human taste, the amount of satisfaction is predictable. In addition, the proposed method has unique features. Including the ability to generate music with the desired length and width, the ability to combine and generate music in a variety of styles, the appropriate speed in the generation, and also, paying attention to the opinions of listeners during the generation of notes.

## 7 Conclusion

In this paper, we proposed an evolutionary method for the automatic generation of polyphonic music. In the proposed method, we first generated a collection of polyphonic music pieces using a genetic algorithm and a database of human-made pieces as the objective function. We then stored the range of different outputs generated in this section for scoring expert and normal listeners and trained the two Bi-LSTM artificial neural networks using the opinions of these two groups of listeners. Finally, we used another genetic algorithm and set the music rules as well as trained models as separate objectives for the algorithm. So, living (human) and non-living (musical grammar) evaluators were simultaneously used in the automatic music generation. Joining the rules and humans has increased the reliability of the music generation system. The results showed that the use of neural networks and other objective functions together accelerates the convergence. Furthermore, in a subjective evaluation, it was observed that more desirable samples of music could be achieved, which were able to obtain high scores from the audiences. Overall, the results being obtained in this study and other studies in the field of automatic music generation may eventually lead to integrated models that, while they are acceptable and pleasurable to the audience, can consider all-in-one music rules.

## Declarations

**Competing interests** The authors have no relevant financial or non-financial interests to disclose.

## References

1. Agarwal S, Saxena V, Singal V, Aggarwal S (2018) Lstm based music generation with dataset preprocessing and reconstruction techniques. In 2018 IEEE symposium series on computational intelligence (SSCI) (pp. 455–462). IEEE
2. Agres KR, DeLong JE, Spivey M (2009) The sparsity of simple recurrent networks in musical structure learning. In proceedings of the annual meeting of the cognitive science society (Vol. 31, no. 31)
3. Agres K, Herremans D, Bigo L, Conklin D (2017) Harmonic structure predicts the enjoyment of uplifting trance music. *Front Psychol* 7:1999
4. Anders T, Miranda ER (2009) A computational model that generalises Schoenberg's guidelines for favourable chord progressions. In proceedings of the sound and Music computing conference (pp. 48–52)
5. Boulanger-Lewandowski N, Bengio Y, Vincent P (2013) High-dimensional sequence transduction. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 3178–3182). IEEE
6. Brooks FP, Hopkins AL, Neumann PG, Wright WV (1957) An experiment in musical composition. *IRE Trans Electron Comput* 3:175–182
7. Browne TM, Fox C (2009) Global expectation-violation as fitness function in evolutionary composition. In workshops on applications of evolutionary computation (pp. 538–546). Springer, Berlin, Heidelberg
8. Chemillier M (2001) Improvising jazz chord sequences by means of formal grammars. In *Journées d'informatique musicale* (pp. 121–126)
9. Davismoon S, Eccles J (2010) Combining musical constraints with Markov transition probabilities to improve the generation of creative musical structures. In *European conference on the applications of evolutionary computation* (pp. 361–370). Springer, Berlin, Heidelberg
10. de Freitas ARR (2011) *Música evolutiva: uma abordagem computacional para composição algorítmica* (Doctoral dissertation, Universidade Federal de Ouro Preto)
11. Eck D, Schmidhuber J (2002) A first look at music composition using lstm recurrent neural networks. *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale* 103:48
12. Farzaneh M, Toroghi RM (2019) Music generation using an interactive evolutionary algorithm. In *Mediterranean conference on pattern recognition and artificial intelligence* (pp. 207–217). Springer, Cham
13. Franklin JA (2006) Recurrent neural networks for music computation. *Inform J Comput* 18(3):321–338
14. Herremans D (2016) MorpheuS: automatic music generation with recurrent pattern constraints and tension profiles
15. Herremans D, Chuan CH (2017) Modeling musical context with word2vec. *arXiv preprint arXiv:1706.09088*
16. Herremans D, Sörensen K (2012) Composing first species counterpoint with a variable neighbourhood search algorithm. *Journal of Mathematics and the Arts* 6(4):169–189
17. Herremans D, Sörensen K (2013) Composing fifth species counterpoint music with a variable neighborhood search algorithm. *Expert Sys Appl* 40(16):64276437
18. Hiller L, Isaacson L (1993) Musical composition with a high-speed digital computer. *Mach Models Music*, 9–21
19. Horner A, Goldberg DE (1991) Genetic algorithms and computer-assisted Music composition. In *ICGA* (pp. 437–441)
20. Keller RM, Morrison DR (2007) A grammatical approach to automatic improvisation. In *proceedings of the sound and Music computing conference*(pp.330-337)
21. Makris D, Kaliakatos-Papakostas M, Karydis I, Kermanidis KL (2017) Combining LSTM and feed forward neural networks for conditional rhythm composition. In *international conference on engineering applications of neural networks* (pp. 570–582). Springer, Cham
22. Manzelli R, Thakkar V, Siahkamari A, Kulis B (2018) An end to end model for automatic music generation: combining deep raw and symbolic audio networks. In *proceedings of the musical Metacreation workshop at 9th international conference on computational creativity*. Salamanca, Spain
23. McIntyre RA (1994) Bach in a box: the evolution of four-part baroque harmony using the genetic algorithm. In *proceedings of the first IEEE conference on evolutionary computation*. IEEE world congress on computational intelligence (pp. 852–857). IEEE



24. McVicar M, Fukayama S, Goto M (2014) AutoLeadGuitar: automatic generation of guitar solo phrases in the tablature space. In 2014 12th international conference on signal processing (ICSP) (pp. 599–604). IEEE
25. Mishra A, Tripathi K, Gupta L, Singh KP (2019) Long short-term memory recurrent neural network architectures for melody generation. In soft computing for problem solving (pp. 41–55). Springer, Singapore
26. Music A (1991) Creation by refinement and the problem of algorithmic music composition. *Music Connection*, 212
27. Nakamura JI, Kaku T, Hyun K, Noma T, Yoshida S (1994) Automatic background music generation based on actors' mood and motions. *J Vis Comput Animat* 5(4):247–264
28. Pachet F, Roy P, Barbieri G (2011) Finite-length Markov processes with constraints. In Twenty-Second International Joint Conference on Artificial Intelligence
29. Papadopoulos A, Roy P, Pachet F (2014) Avoiding plagiarism in Markov sequence generation. In Twenty-Eighth AAAI Conference on Artificial Intelligence
30. Pinkerton RC (1956) Information theory and melody. *Sci Am* 194(2):77–87
31. Scirea M, Togelius J, Eklund P, Risi S (2017) Affective evolutionary music composition with MetaCompose. *Genet Program Evolvable Mach* 18(4):433–465
32. Todd PM (1989) A connectionist approach to algorithmic composition. *Comput Music J* 13(4):27–43
33. Tuohy DR, Potter WD (2005) A genetic algorithm for the automatic generation of playable guitar tablature. In ICMC (pp. 499–502)
34. Waschka II, R. O. D. N. E. Y. (2007) Composing with genetic algorithms: GenDash. In evolutionary computer Music (pp. 117–136). Springer, London
35. Wu J, Hu C, Wang Y, Hu X, Zhu J (2019) A hierarchical recurrent neural network for symbolic melody generation. *IEEE transactions on cybernetics* 50(6):2749–2757
36. Yang, L. C., Chou, S. Y., & Yang, Y. H. (2017). MidiNet: a convolutional generative adversarial network for symbolic-domain music generation. *arXiv preprint arXiv:1703.10847*.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.