

# A Neuro-memetic System for Music Composing

Jacek Mańdziuk<sup>1</sup>, Aleksandra Woźniczko<sup>2</sup>, and Marcin Goss<sup>2</sup>

<sup>1</sup> Warsaw University of Technology, Faculty of Mathematics and Information Science,  
Koszykowa 75, 00-662 Warsaw, Poland  
j.mandziuk@mini.pw.edu.pl  
<sup>2</sup> Entrepreneurs (private sector)

**Abstract.** For thousands of years music has accompanied human existence and development. Over the time it turned into a form of art capable of expressing beauty, ideals and emotions. Our previous work has investigated the possibility of automatic generation of music that would be (to some extent) alike to the music created by human composers. Our focus was on romantic era music, in particular a “Chopin-style” compositions. We have proposed a specifically designed memetic algorithm, which operated based on a handful of parameters and rules, which need to be followed when it comes to classical music. In this paper we review the proposed approach and extend it by introducing “a subjective factor” to the system, in the form of a specifically designed neural network. The role of this component is to provide the subjective preferences of the listener, which are taken into account during the music composition process. Preliminary results of this new system are presented.

## 1 Introduction

The topic of music creation has been of research interest for about 30 years. One of the earliest experiments was the soundtrack to a game called “Ballblazer” generated algorithmically in 1984. The lead melody was assembled from a predefined set of 32 eight-note melody fragments or riffs, which were put together in a random manner. The system had to make several parameter-based choices including the speed and loudness of playing, omitting or eliding notes, or inserting a rhythmic break. The melody was accompanied by bassline, drums and chords, which were also assembled on the fly by a simplified version of the above approach. In effect the music was played forever, without repeating itself, and not straying too far from the original theme [1], [2].

Another interesting example is an interactive music system called *iMUSE* created in 1990 and used in a number of video games. The system is able to “adjust” or “synchronize” the composed music to the game’s action and to make transitions from one musical theme to another [3]. Probably the most similar system to our goal is “Emily Howell” - a computer program with an interactive interface that allows both musical and language communication. The system can be “taught” to compose music more fitting an operator’s taste in the process of “encouraging and discouraging”. The program uses musical pieces, previously invented by another composing program called Experiments in Musical Intelligence (EMI), as a source database for its musical choices [4], [5].

More recent examples of the systems capable of or supporting artificial music composing include ArtSong [6], Symbolic Composer [7], Cybermozart [8] or Lexikon-Sonate [9].

The above-mentioned examples inspired us to develop a system capable of creating pieces of music to some extent similar to those composed by humans and at the same time being able to take into account individual musical preferences of the operator.

The rest of the paper is organized as follows. In the next section we summarize our memetic approach to automatic music creation [10] along with a selective introduction to the theory of music. Section 3 details the role and significance of the neural network component added to the system. Conclusions and directions for possible expansion of this research are placed in the last section.

## 2 A Memetic Approach To Music Composing

The term *music* has too broad scope to be effectively covered in a single approach. Hence, for the sake of tractability, we decided to restrict our research to the *romantic era*, as music created at that time was still following classical rules, but at the same time, the harmony was not evident. Additionally, we decided to compose pieces for the piano as this reflects the expertise of one of the authors.

Our goal was to create a system that would compose a regular piece of classical music based on user's input. In particular, we aimed at verifying the two following aspects: whether the musical rules and regularities could be flexibly applied and whether it is possible to reflect an individual taste in composed pieces.

Our studies referred to three piano music forms popular in the romantic period, i.e. *nocturne*, *polonaise* and *mazurka*. In particular our focus was on the compositions of Fryderyk Chopin, who wrote many such pieces.

### 2.1 The Structure of Composed Pieces / Keys / Repetitions

In order to reflect the rules of music composition and structure of musical pieces we distinguished the following elements (listed in a descending order): a part, a period, a sentence, a phrase, and a motive. Each of these elements (except for motives) is divided

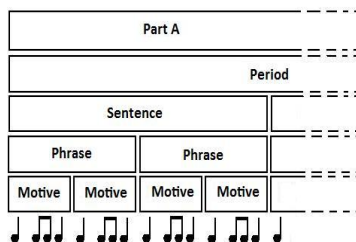


Fig. 1. Exemplary structure of a music piece

into several other smaller elements (see Fig. 1). Each motive consists of two separate lines which are a melody and an accompaniment.

Another important part of the theory of music are keys [11], [12]. Each part of a piece has its base key, but it is possible that the key changes into a temporary key, which may be different from the base key. For each of the keys there are four other related keys: same name key, parallel key, subdominant key and dominant key (see [10] for a detailed explanation). Another crucial aspect of classical music are repetitions. This issue is addressed by creating a repetition matrix defining the probability of repeating each base element within the part as well as between parts. Based on the theory of music different probability schemes were implemented for *nocturne*, *mazurka* and *polonaise*, respectively (see Table 1). In the table  $m$ ,  $h$ ,  $s$  and  $p$  denote motive, phrase, sequence and part, respectively. The row marked with a black square defines the probability of repeating a given element in the same part (e.g.  $s = 0.55$  in column  $A$  means that on average any sentence in part  $A$  is repeated with probability 0.55). The remaining rows define probabilities of choosing a particular part as a source one (for an elements' repetition). For instance,  $m = 0.2$  in row  $B$  and column  $C$  denotes the probability of repeating the a motive form part  $B$  in part  $C$ . The first column denotes the respective repetition probabilities in parts  $A^*$ ,  $B^*$  and  $C^*$ , respectively, i.e. the parts which represent “variations” of the original parts. Based on the theory of music the following schemes were implemented:  $ABA$  for *nocturne*,  $ABA^*$  for *mazurka* and  $ABACA$  for *polonaise*.

**Table 1.** One of the repetition matrices used in the experiment. See description within the text.

	[]*	A	B	C
.	-	m: 0.8 h: 0.5 s: 0.25 p:0.02	m: 0.8 h: 0.5 s: 0.25 p: 0.02	m: 0.8 h: 0.5 s: 0.25 p: 0.02
A	m: 0.7 h: 0.5 s: 0.3 p: 0.02	m: 0.9 h: 0.7 s: 0.55 p: 0.6	m: 0.2 h: 0.05 s: 0.02 p: 0	m: 0.9 h: 0.7 s: 0.3 p: 0.8
B	m: 0.9 h: 0.8 s: 0.55 p: 0.6	-	m: 0.8 h: 0.5 s: 0.3 p: 0.26	m: 0.2 h: 0.05 s: 0.02 p: 0
C	m: 0.8 h: 0.5 s: 0.3 p: 0.25	-	-	m: 0.8 h: 0.6 s: 0.25 p: 0.02

## 2.2 Memetic Algorithm

Our music composing system is based on memetic computing which is a combination of genetic algorithm and in-generation, local optimization. The local optimization

phase serves as an additional mutation operator which does not mutate the specimen randomly, but changes it according to a set of predefined rules that represent domain knowledge. After a pre-defined number of algorithm's generations the system yields a specimen that represents a musical piece with the highest fitness value.

The algorithm receives a set of parameters as its input, part of which is a set of standard parameters of the evolutionary algorithm and the other part results from the plug-in architecture of the fitness function and local optimization. These parameters are set directly by the user and include: the number of specimens in the population, the number of generations, the percentage of the elitist specimens (transferred to the next generation without mutation), the probability of performing local optimization, the weights of particular components of the fitness function, as well as, parameters responsible for mutation (its range, probability of notes modifications, probability of chords modifications, and permutation range).

Furthermore, the algorithm uses parameters which reflect the rules of composing homophonic music: the key graph, probabilities of occurrence of particular chords in a particular key and probabilities of occurrence of particular notes in particular chords in particular keys. The final set of parameters describes the musical form, like possible structures of the piece, its meter, textures and repetition matrix.

### 2.3 Generation of a Population

The process of primary specimens generation is analogous to the base methods of (traditional) music composing, i.e. first harmonic structure of the piece is created and then it is filled with particular notes. The initial population consists of some pre-defined number  $N$  of specimens generated pairwise independently. In this phase of the algorithm musical rules and form parameters are exploited most intensively.

Generation of a single specimen is quite a complicated process and consists of numerous steps, as shown in [10]. On a general note, first, the biggest elements (parts and periods) of a piece are generated, followed by the smaller ones (sentences, phrases and motives).

### 2.4 Memetic (Local) Optimization

The goal of local optimization is to improve the quality of a specimen from musical theory point of view. Having in mind that some amount of "randomness" seems to be beneficial in artificial music composing, local optimization may happen in each generation but with some probability defined as one of the input parameters.

The first local optimization consists in adding a pitch to the existing chord which is the sixth or the seventh step of the chord's key. This operation is based on an extended version of a database of chains of chords with three or four elements used in *Harmonia* educational tool [13]. It includes chains of possible chords together with their possible modifications. If a particular chain is present in the database, it is then modified accordingly.

The second optimization operation is focused on technical ability to play the composed piece (i.e. the scale and chords location can be modified, if necessary, so as to actually allow playing the piece with two hands of a pianist).

The third operation allows improving the chords played on either a melody or an accompaniment so as there are more consonants (concord nice to hear) than dissonances (concord strange to hear). The last local improvement operation provides a way to minimize the number of pauses appearing in the musical piece. As pauses, in general, are in line with musical rules, they are usually “over-represented” in the specimens.

## 2.5 Fitness Function and Selection

The selection is roulette-elite. The fitness function consists of several components which are independently evaluated and have different weights (depending on the form) configured by the user. Specifically, the following components are considered: *Chains* - evaluates the correctness of chains of chords using pre-defined Markov chains, *Concords* - evaluates concords based on the number of existing consonances and dissonances in a strike, *Rhythm* - evaluates the homogeneity of the rhythm in the motives, *Tonality* - evaluates the tonality on the main measures in the meters as high tonality should be maintained, *Pitch* - evaluates the homogeneity of notes' pitches in the motives.

## 2.6 Mutation

The degree of mutation operator depends on the parameter that sets the percentage of the motives that will be mutated. Mutation may either change the heights of particular notes or change the chords. Moreover, mutation may change the order of motives based on the mixing intensity parameter. If mutation is to change single notes, then a decision is made independently for each of them, with a pre-defined probability threshold. If so, first the origin of the new note is decided (whether it is the current chord, a key, base key or random), then a particular note is chosen. If mutation is to change chords, the procedure is similar to the one used for changing notes, but the origin may be main chord of current key, side chord of current key, main chord of another key, side chord of another key or random.

The degree of the other facet of mutation - changing the order of motives - depends on the mixing intensity parameter, which defines the percentage of motives to be re-ordered. First, a set of motives which will remain in the primary order is chosen. All other motives are inserted in random places between the “stable” motives in the piece.

## 2.7 Summary of Results

The initial series of tests aimed at optimization of the steering parameters of the memetic algorithm: *% of elite* - percentage of the specimens with the highest fitness automatically promoted to the next generation, *% of optimization* - probability of the local optimization to take place in the current generation, *mutation range* - the percentage of motives to be mutated, *notes/chords modification factor* - a factor describing the relation between probabilities of mutation of notes and chords, e.g. 3/1 means 75% for notes and 25% for chords, *permutation range* - the percentage of motives that will remain in the primary order. Each combination of parameters was verified in 10 independent experiments with population size equal to 150. Each ensemble of 10 tests was performed three times, once per each considered form (*mazurka*, *polonaise* and *nocturne*).

The final selection of the algorithm’s parameters was based on the average score of the best specimen evaluations across 10 populations. It turned out that for each of the tested forms different sets of parameters were the ones most effective. The best set of parameters of the *mazurka* form: % of elite= 1, % of optimization = 15, mutation range = 33, notes/chords modification factor = 3/1, permutation range = 25, has been chosen for further final tests, which involved three human testers.

Each of the testers received the same set of 300 pieces (of *mazurka* form) to listen and evaluate using the scale of -1 (worst) to 1 (best) with a step equal to 0.5. The confusion matrices for each pair of the testers are presented in Fig. 2. Generally speaking, it is

		Tester 1					
		Mark	-1	-0.5	0	0.5	1
Tester 2	-1	18	10	8	5	1	
	-0.5	11	30	46	23	7	
	0	12	26	31	21	6	
	0.5	4	10	8	16	3	
	1	0	1	1	2	0	

		Tester 1					
		Mark	-1	-0.5	0	0.5	1
Tester 3	-1	18	12	8	8	2	
	-0.5	19	44	38	23	2	
	0	6	10	34	19	9	
	0.5	2	8	11	14	3	
	1	0	3	3	3	1	

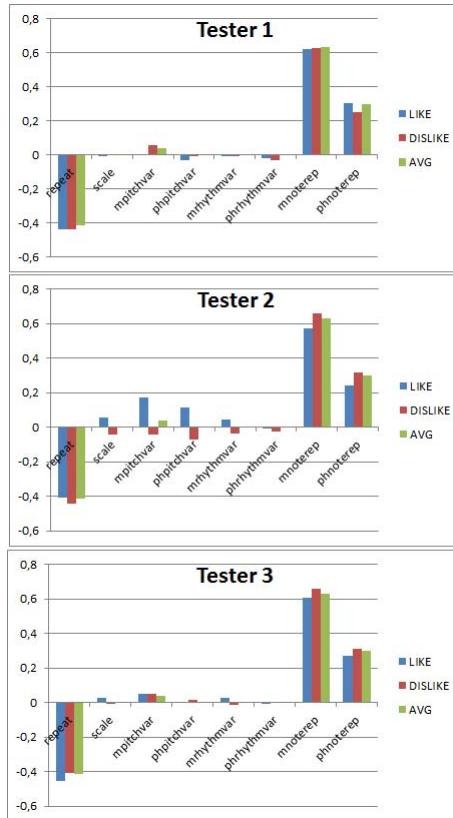
		Tester 2					
		Mark	-1	-0.5	0	0.5	1
Tester 3	-1	18	11	13	6	0	
	-0.5	14	49	47	16	0	
	0	8	38	21	9	2	
	0.5	2	17	11	6	2	
	1	0	2	4	4	0	

**Fig. 2.** Comparison of testers’ evaluations in the form of confusion matrices. For instance, value 11 in the first table means that there were 11 pieces evaluated as -1 by Testers 1 and as -0.5 by Tester 2.

easy to notice that the testers are often consistent when it comes to the pieces marked lower. This can be seen in the top left corners of the matrices. Furthermore, the testers do not always agree when it comes to the pieces rated with the highest marks which is most probably caused by differences in testers’ tastes as well as a small number of the highest rated pieces in the set. A detailed analysis of results is presented in [10].

### 3 A Neural Network Component

The above described system can be regarded as an “objective” mechanism for composing music according to the rules (parameters) appropriate for a given musical form. In the current version of the system we added a neural network component, which reflects the user’s taste and whose score was added to the fitness function. Neural network training is preceded by extracting style parameters from the pieces listened by the testers



**Fig. 3.** Average values of style parameters of pieces highly evaluated by the testers (evaluations of 0.5 and 1) and pieces low evaluated (evaluations of -1 and -0.5) in comparison to averages of style parameters in the whole population

(see Fig. 3). The analysis of these graphs allows to observe that the testers have different style parameters in the pieces they prefer. For instance, it is easy to observe that *Tester 2 negatively judges pieces with low repetition rate of motives. The same pieces are evaluated higher by Tester 3 or Tester 1 positively evaluates pieces with low variance of notes height in motives, unlike Tester 2.*

In the training process the input to the network consists of a set of eight style parameters, depicted in Fig. 3, extracted from the 300 evaluated pieces:

- the ratio of the number of motives versus the number of unique motives
- the difference between the highest and the lowest note in the piece
- average value of the variance of the height of notes in motives
- average value of the variance of the height of notes in phrases
- average value of the variance of the rhythmic values in motives
- average value of the variance of the rhythmic values in phrases

- average ratio of unique heights of the notes versus number of notes in motives
- average ratio of unique heights of the notes versus number of notes in phrases

The choice of parameters reflected our willingness to find such a characteristics of a piece that may take different values for various pieces and may be correlated (positively or negatively) with the individual musical taste of the listener (tester).

For each specimen (musical piece) obtained in the previous experiment, all eight parameters are extracted, normalized to  $\langle -1; 1 \rangle$  and used as an input to a neural network. The hidden layer consists of four neurons and the activation function is bipolar sigmoid. The network’s single output belongs to  $\langle -1; 1 \rangle$  and represents the degree to which the piece reflects the listener’s taste.

Neural networks (one for each tester) taught in the previous test were included in the algorithm and each of them was used to generate 50 pieces. Additionally, 50 pieces were generated with the help of previously described memetic system, with no use of a neural network.

For each of the testers, after mixing up 50 pieces generated with the use of the respective neural network and 50 ones generated without the network, the 100 pieces set was ready to use in the subsequent tests. Each set was mixed twice (the pieces were shuffled) and given to the respective tester without the labels. As a result each tester had two lists of the same pieces in different order and listened to them with no knowledge which pieces were generated using the neural network and which were not. In order to increase the credibility of the test listening to the first set was followed by a 24 hour break before the tester started to listen to the second set.

The average scores of the testers are presented in Fig. 4. For the sake of clarity it should be noted that among the three testers there were two co-authors of this paper. The third person, Tester 3, was completely unrelated with the paper and implementation of the project and may be found as the most independent source of tests data. On the other hand, due to “blind” construction of tests it was practically impossible to intentionally manipulate the results of Testers 1 and 2, thus we decided to publish them on equal rights as those of Tester 3.

The results of the performed tests may be assessed as moderately positive. Even though the evaluations are not always fully consistent, in all three cases the average evaluation of all listenings is higher for the pieces generated using the neural network. This allows us to say that using the neural network is beneficial for matching the created music to ones taste. On the other hand the observed improvement in the quality of

	AVG Neuro	AVG Other	AVG Neuro	AVG Other	AVG Neuro	AVG Other
	Run 1	Run 1	Run 2	Run 2		
Tester 1	2,78	2,52	3,02	2,72	2,9	2,62
Tester 2	3	2,74	2,96	2,9	2,98	2,82
Tester 3	2,8	2,5	2,58	2,58	2,69	2,54

Fig. 4. Summary of the results of “blind” test



specimens is not substantial. Relatively large differences between two runs of the experiment performed by the same user should, to a large extent, be attributed to changing and uncertain nature of testers' (human) perception.

In summary, the results allow us to say that the approach presented in the paper is a promising idea in the area of computer music generation. The proposed memetic system is able to construct pieces imitating popular musical forms and the additional use of neural network allowed generation of the pieces statistically more attractive to the recipient. This allows to ascertain that generated musical pieces reflect some aspects of human/artificial creativity and extend beyond the formal rules and theory of music.

During the experiments a few hundreds of music pieces of various forms (*polonaises*, *mazurkas* and *nocturnes*) and quality were generated by the system. Some of them clearly sound “artificial” and by no means may pretend to be composed by a (non-novice) human composer. There are, however, also quite many examples of “more human-like” compositions. An example of such a piece (of the form of *mazurka*) is presented in figure 5.

Test Piece

Random Composer

The figure displays a musical score for a piece in the form of a mazurka. It is presented in two columns: 'Piano' on the left and 'Random Composer' on the right. Each column contains five systems of music, each with a treble and bass clef staff. The key signature is one sharp (F#) and the time signature is 3/4. The piano part features a melodic line with characteristic mazurka rhythms, including dotted eighth and sixteenth notes, and some chromaticism. The random composer part provides a harmonic accompaniment with chords and bass lines. The score is numbered 1 through 21 across the systems.

Fig. 5. Example piece of the form of *mazurka*

## 4 Conclusions and Directions for Future Work

In this work we studied two main aspects of the composed music. The first one refers to the initial stage of generating a musical piece. The hybrid algorithm allows creation of pieces, which are consistent with the user's expectations in the field of musical rules, harmonic and rhythmic restrictions, types of used textures and construction of the whole

form. The fitness function evaluates the actual state of the piece and local optimization brings in the “formal order”. Repeated execution of these operations allows achieving the goal of a piece containing expected musical substance. The end result is substantially dependent on the user’s intentions and ability to translate theoretical knowledge into configuration of individual parameters and the input of the algorithm.

The second relevant aspect that we analyzed in the paper regards reflecting individual taste of particular listeners. Test results clearly show the influence of the use of neural network that learned from a person’s preferences on the final quality of the generated piece. Thanks to this mechanism it is possible to not only generate music which fulfills the expectations related to the musical substance, rhythmic, harmonics and texture, but also reflects, to some extent, the sense of aesthetics and beauty of an individual user.

Our current focus is on deeper analysis of the factors defining user’s individual preferences, which may potentially enhance the neural network’s results. We also plan to perform more test involving other testers possibly with the help of social networks.

**Acknowledgements.** The authors would like to thank Dr Eddy Chong from NIE (Singapore) for providing them a database of chords chains that was used as part of domain knowledge representation in the local optimization phase.

## References

1. Langston, P.S.: Six Techniques for Algorithmic Music Composition. In: 15th International Computer Music Conference, Columbus, Ohio, p. 6 (November 2, 1989)
2. Langston, P.S.: (201) 644-2332 or Eddie & Eddie on the Wire, An Experiment in Music Generation. In: 1986 Usenix Association meeting, pp. 4–5 (Summer 1986)
3. Smith, R.: Rogue Leaders: The Story of Lucas Arts. Chronicle Books, San Francisco (2008) ISBN 978-0-8118-6184-7
4. Cope, D.: Experiments in Musical Intelligence. A-R Editions, Madison (1996)
5. Cope, D.: The Algorithmic Composer. A-R Editions, Madison (2000)
6. Algorithmic Composition Software - ArtSong, <http://www.artsong.org>
7. Symbolic Composer, [http://www.symboliccomposer.com/page\\_main.shtml](http://www.symboliccomposer.com/page_main.shtml)
8. CyberMozart, <http://www.yav.com/CyberMoz.html>
9. Karlheinz Essl, Lexikon-Sonate, <http://www.essl.at/works/Lexikon-Sonate.html>
10. Mańdziuk, J., Goss, M., Woźniczko, A.: Chopin or not? A memetic approach to music composition. In: IEEE Congress on Evolutionary Computation (CEC 2013). IEEE Press, Cancun (2013)
11. Benade, A.H.: Horns, Strings, and Harmony. Science Study Series S 11. Doubleday & Company, Inc., Garden City (1960)
12. Kostka, S., Payne, D.: Tonal Harmony, 5th edn. McGraw-Hill, New York (2004)
13. Chong Kwong Mei, E.: National Institute of Education in Singapore (2013)