



THE USE OF A GENETIC ALGORITHM FOR MUSIC HARMONIZATION OF TRANSITIONAL FRAGMENTS IN COMPUTER GAMES

Oleksandr Komarov, Oleg Galchonkov,
Alexander Nevrev, Oksana Babilunga

Odessa National Polytechnic University, 1 Shevchenko Ave., Odessa, Ukraine, 65044
o.w.komarow@gmail.com, o.n.galchenkov@gmail.com, a.i.nevrev@gmail.com, babilunga.onpu@gmail.com

Paper history:

Received 03 May 2019
Received in revised form 23 May 2020
Accepted 08 July 2020
Available online 27 September 2020

Keywords:

genetic algorithm;
chromosome;
fitness function;
tonal model;
population;
harmonization;
computational costs.

Abstract: The problem of musical accompaniment harmonization for adjacent transitional fragments in computer games is considered. A harmonization method based on a tonal model of the musical composition in conjunction with a genetic algorithm seeking the global extremum of the harmonic function is proposed. For the method under consideration, all operators of the genetic algorithm and fitness functions are defined. A scheme was developed for the formation of transitional musical compositions and a flowchart was proposed for using the genetic algorithm in its implementation. Using examples of well-known musical compositions, a study was conducted of the genetic algorithm features and its most appropriate parameters were found in the context of the problem under consideration. This allowed us to obtain expression for the computational costs of the proposed method, which are necessary for balancing the computational load in the computer between the game algorithm itself and the musical harmonization algorithm. Moreover, the study showed that using genetic algorithm allows even for a small number of iterations to achieve melody harmonization level no worse than professional composer achieved, and even exceed this level when performing extra iterations.

Copyright © Research Institute for Intelligent Computer Systems, 2020.
All rights reserved.

1. INTRODUCTION

Video games are an actively and continuously developing modern media trend [1]. Studies show that approximately 60% of Americans play video games, and the average age of players is 30 years [2]. One of the important components of a modern video game is music. In a video game, musical accompaniment reinforces the narrative [3], represents characters and scenes, enhances dramatic tension, and can also be a meaningful part of the gameplay [4].

Along with the growing interest in procedurally generated content, there is also a request for the use of procedurally generated music (PGM) [5]. This is due to the desire to reduce the cost of game development [1, 6], the need for personalization of the game experience [7], a long or often repeated gameplay, the full cycle of which can take several tens of hours [4].

Although the practices and methods of writing music for games come from film scoring, the

interactive nature of the video game requires significant changes in the approach to creating musical accompaniment. This distinction is a result of an active, rather than passive character of the player role in the game. There are adaptive and interactive PGM systems [5]. In the interactive PGM system music is considered as part of the gameplay, and the player has the ability to directly influence the music parameters. In adaptive PGM systems, interaction with the player occurs more indirectly - through associations of game states and events with various musical themes [8]. Such system should be synchronized with the main game subsystems, since it needs to quickly react to changes in the gameplay and notify the player about this by changing the music [9].

From the aesthetic point of view the harmonious transition from the current fragment of musical accompaniment in the gameplay to the next one is extremely important. In PGM implementation harmony is the compliance of musical

accompaniment to all the rules of classical musical harmony [10]. We mean harmonization as the process of assignment musical chords to a given musical theme, presented as a melody. In adaptive PGM harmonization usually is used only for the current musical theme, and when changing the theme, a pause is used in musical accompaniment.

Using harmonization for musical theme transitions could significantly improve perception of the gameplay. Therefore, the development of methods for smooth harmonic change of musical themes is relevant.

This task is complicated by both the uncertainty of the moment of changing the musical themes and the uncertainty of the next theme choice, which are caused by the current game state and also the player's actions. Providing a continuous harmonic change of musical themes in these conditions is possible only if there is a continuous preliminary calculation of all possible transitions between musical themes. However, the volume of these calculations should not exceed the available computing power in order not to slow down the calculation of actions in the computer gameplay itself.

2. RELATED WORKS AND PROBLEM FORMULATION

Current use of the well-known PGM methods in video games is limited due to the heavy CPU load [5, 9]. The standard response time for player's actions in gameplay should be within 150-200 ms [11, 12]. When playing online, delay response over this value results in partial loss of gameplay control [13]. The time delay requirement for adaptive musical accompaniment is not so strict due to its indirect connection with the player's actions, but music should be synchronized with the gameplay, and its changes must correspond to the gameplay pace. The musical accompaniment should react to game state changes and notify the player about it [3].

In the video game the musical accompaniment should be associated with key game states by certain musical themes [4, 8]. The total set of all the musical themes used in the current game session we will call the repertoire.

The scheme of game states for each specific episode of the game can be represented as a graph, the vertices of which are game states, and the edges are possible transitions (Fig. 1). The transition from the current state to one of the other possible will be called a scenario. If the musical themes of the game states are represented by melodies, they must be harmonized before playing.

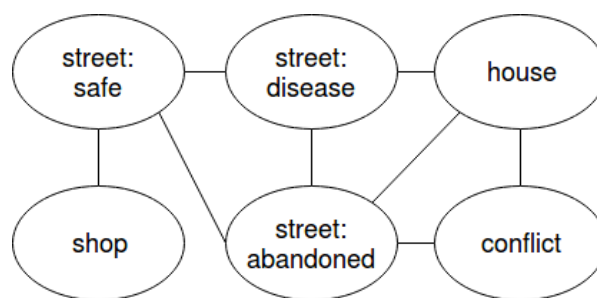


Figure 1 – Simplified game state graph in Pathologic game (2005) [14]

To ensure a harmonious musical transition from one game state to another, it is necessary to provide mutual harmonization of the boundary sections of both melodies. These boundary sections of adjacent musical themes we will call the transition area. Fig. 2 presents an example of a possible dynamically transition areas (dotted line).

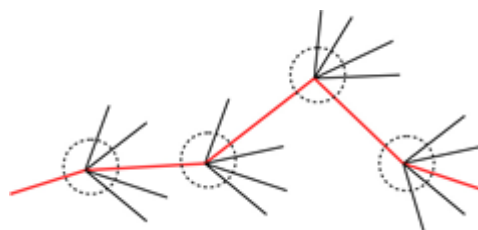


Figure 2 – Possible musical themes transition scenarios

For each game state (if it is not the only one) there is at least one scenario connecting it with other states, or one degenerate scenario case which continues the current theme.

The choice of the scenario and the transition moment is usually determined by the player action so they are random for PGM. The adaptive music accompaniment system, therefore, should provide a harmonious transition from the current musical theme to any other possible in real time without delay. In this regard, each playback time period can be considered as a transition area.

Music has its own, inherent time division [15]. In temporal sense music duration is a series of bars - equal time parts, containing short, completed musical motifs or their constituent parts. Rhythm is one of the most important components of music perception [16]. When two melodies are combined it is necessary to ensure smooth transition between closing bar of the current melody, and the opening bar of the next melody. Therefore, transition area can occur only after bar closing. The duration of transition area and melody bar in the music accompaniment should be correlated.

Within one melody, the bar duration has constant time value, but for different melodies it can be different. The bar duration is determined by the tempo and rhythm of the melody [17]. It was shown that both the tempo [18] and the rhythm [19] are important features for the emotional perception of music, so we should expect that every game scenario will include melodies with different tempo and rhythm and therefore with different bar duration (Fig. 3).



Figure 3 – Example of the transition area between bars of two melodies with a different rhythm

As the maximum duration of the transition area, we should select the shortest bar duration in the musical accompaniment. This will allow for calculation of all transition scenarios by the end of any bar, and make the transition to the first bar of a new melody immediately.

To realize a harmonious musical transition, joint harmonization of both melodies can be applied. By combining the end of the previous melody with the beginning of a new one, the resulting music passage can be considered as a new short melody. Having harmonized it, we will get a short composition, which can be used as complete transition between two melodies. Further we will call it a musical composition of the transition.

The melody harmonization means supplement of the single melody with chords that satisfy all the rules of musical harmony. Currently, there are various formulations for the rules of musical harmony and, accordingly, different methods of the same melody harmonizing [20]. The implementation of computational melody harmonization requires an effective formal presentation of the musical composition and the formal function of the harmonization quality of the musical composition. A large number of such models are known in the literature [21-23], but they must also be considered in terms of ease of use in conjunction with modern methods of searching for global extrema of complex functions. In this sense the most promising is the tonal model [24], which makes it easy to adapt one of the most effective modern methods of searching for a global extremum - the genetic algorithm for the task of melody harmonizing [24-26].

The classical structure of the genetic algorithm is as follows (Fig. 4).

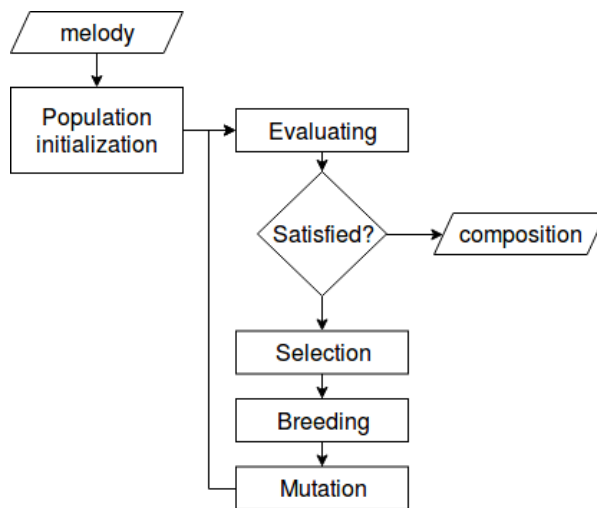


Figure 4 – Algorithmic diagram of melody harmonization using a genetic algorithm

Melody data is algorithm input. Melody is combined with each of the N randomly created chord progressions and forms N compositions. Each composition is called chromosome and a full set of them is called population. The value of harmoniousness function is calculated for each composition of the population. The best harmonious compositions are selected and considered as parents for the crossbreeding. During crossbreeding data of the parents are mixed, and new chromosomes are created. These chromosomes become new population, and this process must be repeated again. To avoid sticking to the local minimum of the harmoniousness function, each iteration includes random changes of chromosomes which are called mutation.

The tonal model [24] is a three-dimensional matrix, the elements of which encode the pitch of the note and the vector of notes durations attached to it (Fig. 5). The rows of the matrix set the individual voices. The columns correspond to the chords. At the intersection of the row and the column, there is a pitch vector of three elements: the tonality degree D , the octave O and the alteration sign A .



Figure 5 – Model of a musical composition used as a chromosome in a genetic algorithm

The model is complemented with vector of durations, showing the duration of each column in

relative value named beat, and the special function d^{-1} , which relates tonality degree with specific pitch class of sound.

The pitch P is calculated as follows:

$$P = d^{-1}(D) + 12 \cdot O + A \quad (1)$$

The standard way to evaluate harmony is a list of harmony rules that describes the relationships of adjacent chords [24, 27, 28], and the total is calculated as the weighted sum of the values of the rules [29]. The set of rules has the general form $F = \{f \mid f : E \times E \rightarrow \{1,0\}\}$ in which rule f gives the value 1 in case if pair of chords E violates it, and 0 otherwise.

Harmony H is a value that can be calculated

using the following formula for composition c using the rules F :

$$H(c) = 1 - \frac{1}{Q(L-1)} \sum_{i=1}^{|F|} \sum_{j=1}^{L-1} q(f_i) f_i(a_j, a_{j+1}), \quad (2)$$

where c is the composition;

L is the number of all chords;

Q is the sum of the weights of all rules;

f_i is the i -th harmonic rule;

$|F|$ - number of rules;

q is the function of the weights of the rules;

a_j - j -th chord in composition.

To calculate the harmony value, we will use a limited list of the most significant rules of musical harmony (Table 1) [24].

Table 1. Harmony rules

Weight q	Rule f	Violation condition
1	Slip in voice	at least one voice contains a horizontal interval longer than 6 semitones
1	Melodic chord connection	two in-line chords do not have common sounds
2	Doubling of the third	at least one chord of the pair contains a doubled third
2	Parallel voice movement	the length of all horizontal interval has one sign
2	Slip in the melody is not compensated by a bass slip	the melody contains a slip, but the lowest voice does not contain it
4	Parallel fifths	horizontal interval with a length of a fifth or more in voices (fifth = 4 steps)
4	Parallel octaves	horizontal interval with a length of an octave or more in voices
4	Non harmonic chord progression	violation of the rules of chord progression (see table. 2)
8	Non tertian chord structure	at least one chord has a non-third structure

Chord progression is formed using the rules that show for each chord the functions of the chords, which can be used after the given one. The function of a chord g means its harmonic value within one

tonality [30,31]. These rules are listed in Table 2. The table shows whether it is possible to use the chord of $g2$ function after the chord of $g1$ function [30].

Table 2. Rules for the chord progression formation

g		$g2$						
		I	II	III	IV	V	VI	VII
g1	I	+	+	+	+	+	+	+
	II	+	+	+	-	+	-	-
	III	+	-	-	+	+	+	+
	IV	+	+	-	+	+	-	-
	V	+	-	+	-	+	+	+
	VI	+	+	+	+	-	+	-
	VII	+	-	-	-	-	-	+

3. PURPOSE AND OBJECTIVES OF THE STUDY

The aim of the work is to develop a method for matching musical accompaniment of adjacent fragments of computer games, based on the use of the tonal model in musical compositions and the genetic algorithm for finding the global extremum of the multi-extremal harmony function.

To achieve the goal, the following tasks were set:

- develop a general computational scheme for the method harmonizing music for transitional fragments in computer games;
- explore typical parameters of the harmonization process using the example of well-known melodies;
- determine the computational operation volume for the developed method and find the conditions under which this volume does not slow down the computational processes in the computer game itself.

4. DESCRIPTION OF THE PROPOSED METHOD

Flowchart in Fig. 6 shows the forming process of the musical composition for a current transition. For its work, the system stores the following internal data:

- repertoire of musical themes M_i ($i=0,1, \dots$),
- M_0 , the current music theme;
- tr , the duration of the transition;
- t , play time M_0 before start playing musical composition of the current transition.

Forming process of the musical composition for a current transition has four stages.

At the first stage, M_0 is input to the SCEN block. The block accesses the scenario map to determine to which music themes $M_1...M_n$ can potentially be transitions from the current theme M_0 , and then extract them from the repertoire repository.

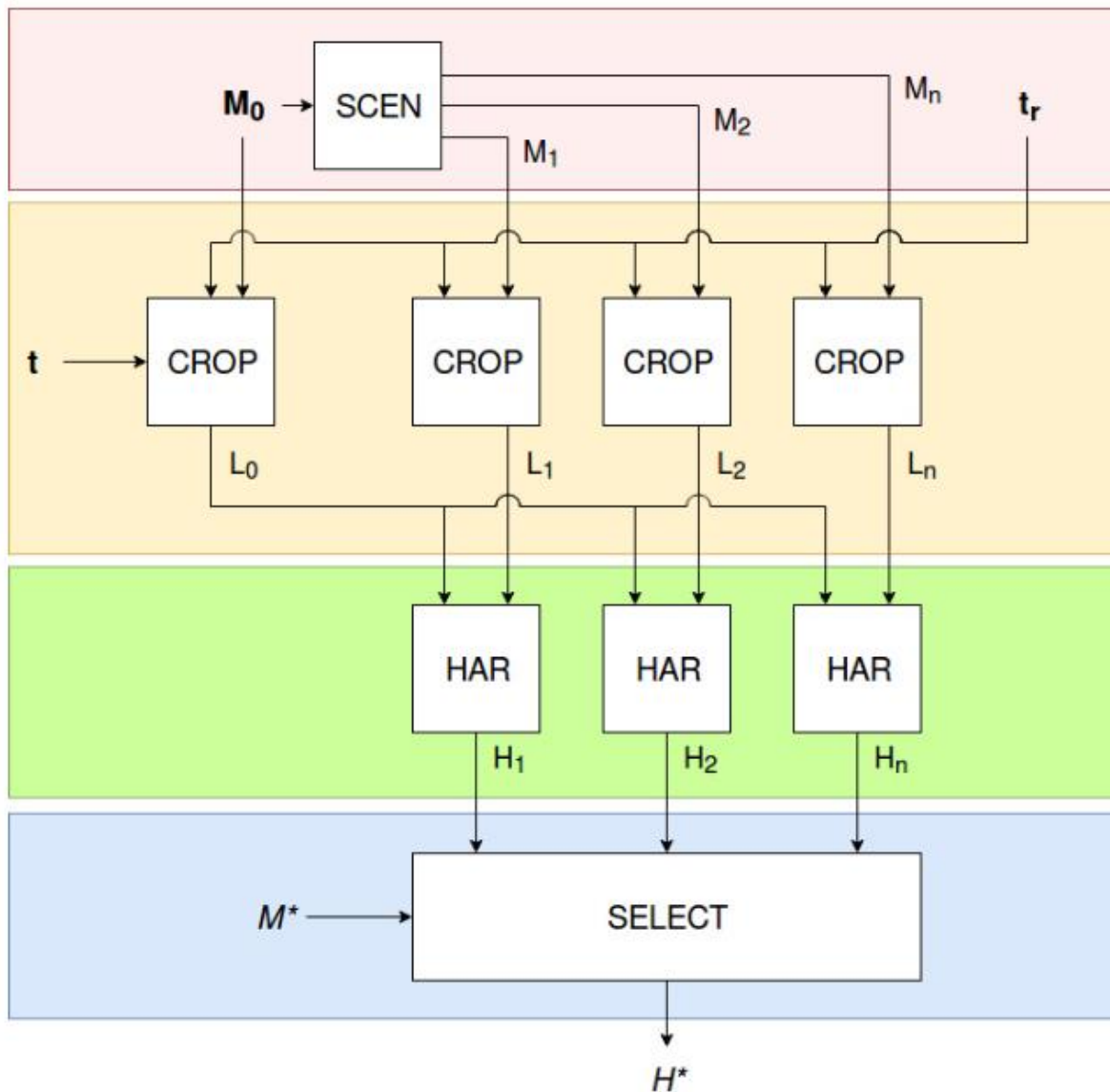


Figure 6 – Flowchart for the forming process of the musical composition for a current transition

At the second stage, CROP blocks, using transition duration t_r , extract the $L0...Ln$ sections from the selected musical themes which will form musical compositions for potential transitions. For themes $M1..Mn$, the segment is always extracted from the beginning of the melody, but for theme $M0$ it is necessary to cut off the part that has already been played. The parameter t allows us to specify the exact place in the $M0$ from which transition area begins.

At the third stage the sections are combined and harmonized. The left and right melodic parts of the transition area are input to each HAR block. The resulting outputs of the HAR block are harmonized musical compositions of the potential transitions $H1...Hn$.

At the fourth stage, the parameter M^* , which determines the musical theme to which the transition is necessary input to the SELECT block. From the set $H1...Hn$ the required musical composition of the transition H^* is selected, which provides a harmonic transition from the theme $M0$ to the theme M^* . It is resulting output musical composition for the transition.

If $M^* = M0$, the time counter t is increased by the value of t_r . Otherwise, the duration of the used initial section of the M^* theme is set in the counter, and the value M^* is assigned to the $M0$ parameter.

5. CHARACTERISTICS OF THE TRANSITION AREA

In general, any number of chords, up to two, can be in the transition region. For each played bar it is necessary to calculate the musical compositions for all potential transitions that may occur after its completion. The worst is the situation when the current theme $M0$ has a minimum bar and we transit to a musical theme also with minimal bar. Then the transition duration of the musical composition will be resulting duration of the two shortest bars.

Bar duration is characterized by two values: the number of beats per minute (BPM) and the time signature. Beat is a conditional time value in which the duration of notes is expressed. BPM allows one to calculate the duration of one beat in seconds using the formula:

$$t = \frac{60}{BPM}. \quad (3)$$

The most commonly used are tempos in the range of 80-160 BPM [32]. The time signature R indicates the bar duration in beat and is usually represented as a natural fraction with a denominator multiple of 2. Used common time signatures usually are $2/2$, $2/4$,

$3/2$, $3/4$, $3/8$, $4/4$, $6/4$, $6/8$ [17]. The number of beats is calculated by the formula:

$$\theta = 4R, \quad (4)$$

where θ is the bar duration in beats.

Thus, the duration of any bar is calculated by the formula:

$$t = \frac{240R}{BPM}. \quad (5)$$

The shortest duration bar will occur in compositions with a minimum time signature and maximum tempo. Among those listed the minimum time signature is $3/8$, and the highest rate is 160 BPM . The minimum duration of such bar according to the formula (5) is $t_{min} = 0,56$ s, therefore the duration of the transition area will be $2t_{min} = 1,12$ s.

The most significant parameter of the computational complexity is the number of chords involved in harmonization, which determine the length of the chromosome. The chromosome has a maximum length if the entire transition area is filled with chords of the shortest duration. It is calculated by the formula:

$$L_{max} = 2 \cdot \frac{\theta}{\theta} = 8 \cdot \frac{R}{\theta}, \quad (6)$$

where θ is the smallest permissible sound duration.

Taking $\theta = 0.5$, which corresponds to the duration of the so-called eighth note, we get $L_{max} = 6$.

6. COMPUTATIONAL COMPLEXITY OF THE METHOD FOR MATCHING MUSICAL ACCOMPANIMENT OF ADJACENT FRAGMENTS

The computational complexity of genetic algorithms is determined by the complexity of the fitness function. Due to the high repeatability of the fitness function calculations, the total calculation volume of the algorithm full cycle is almost equal to the calculation volume of the fitness function [33]. By simplifying the model, we can assume that the time required for chromosome evaluation corresponds to the duration of the entire iteration. Thus, the computational complexity for one iteration is comparable to the computational complexity of assessing the population for harmony.

In the proposed method, computational operations for evaluation the global extremum of the

harmony function (fitness function) are operations with integers.

Let λ be the number of operations with integers necessary to calculate the fitness function for a chromosome with minimum length (2 chords). Then the number of operations required for calculating the fitness function for a chromosome with length L will be $(L-1)\lambda$. The fitness function is calculated for each of the N individuals of the population for each of I iterations in algorithm. The number of operations of the entire cycle for genetic algorithm will be

$$IN(L-1)\lambda. \quad (7)$$

The required computational power M for harmonizing α scenarios in t seconds can be calculated using the formula:

$$M = \frac{\alpha IN(L-1)\lambda}{t}. \quad (8)$$

The high independence of these calculations makes it possible to widely use parallel computing, calculating the fitness function of individuals or even individual pairs of chords in separate computing units.

For $\alpha N(L-1)$ computing units, with the slowest one has a performance of M_{min} , the full cycle operation time will not exceed the value obtained by the formula:

$$t = I \frac{\lambda}{M_{min}}. \quad (9)$$

For the fitness function described by formula (2) and Tables 1 and 2, λ is 3025 operations with integers.

There are no clear theoretical considerations for determining the optimal size N of the population in the case of nonbinary chromosomes. Optimal configuration of the genetic process is traditionally considered complicated problem, requiring special attention to the nature of the data represented by the chromosome [34].

Increasing the size of a population always leads to increase in the overall harmony of the population, since it increases the chance of good results already in the initial population. Increasing the chromosome length increases the search area and reduces this chance, so the choice of population size should be made according to the length of the chromosome.

Simultaneously increasing the size of the population and the length of the chromosome, however, significantly increases the amount of computation, so we should determine the optimal

value of N .

It can be argued that a population increase gives a significant increase in harmony only to a certain value, after which the improvement practically is insignificant. It could be found experimentally.

The harmonization way of musical compositions and the type of harmony function is determined primarily by the musical style in which the composition is written and by the harmony relations specified within the style. These harmonic relations are determined by a set of certain rules, which, as a rule, are independent on composers. The skills and qualification of the composer definitely affects the quality of the musical work, however, for simplicity we believe that if the composer works within the bounds of the desired genre, he compiles the harmonious requirements in the composition at a level sufficient for the operation of the procedural music system. Since at the moment there are no harmony statistics for the music works of various composers, it will be logical to choose well-known works that are considered to be quite harmonious in the music world for determining numerical parameters. In the work we choose the classic harmonic style (for simplicity). For comparison, compositions in classical four-part homophony were selected.

Results in Fig. 7 shows averaged harmony achieved for populations of various sizes with different number of iterations. There melodies from the musical compositions "There is a Balm in Gilead" [35], "He Keeps me Singins" [36], "Moment by Moment" [37], "Jesus, Thy Boundless Love to Me" were harmonized [38]. From the results in Fig. 7 it can be seen that already for the value $N=600$, regardless of the number of completed iterations, the subsequent increase in population size N has no notable effect on harmony.

The result based on the harmonization calculation for typical melodies make it possible to find a simple relationship between the desired level of harmony of the musical composition and the required number of iterations in the genetic algorithm.

Results in Fig. 8 show the dependence of the harmony level on the number of iterations for the genetic algorithm using the population size $N = 600$. For harmonization, melodies from the same musical compositions as in Fig. 7 were used. The dotted line on each graph indicates the level of harmony achieved by the composer. It can be noted that in the range of 10-16 iterations, the algorithm result can already be equal to the composer's level of harmony, which will be quite enough for adaptive music system.

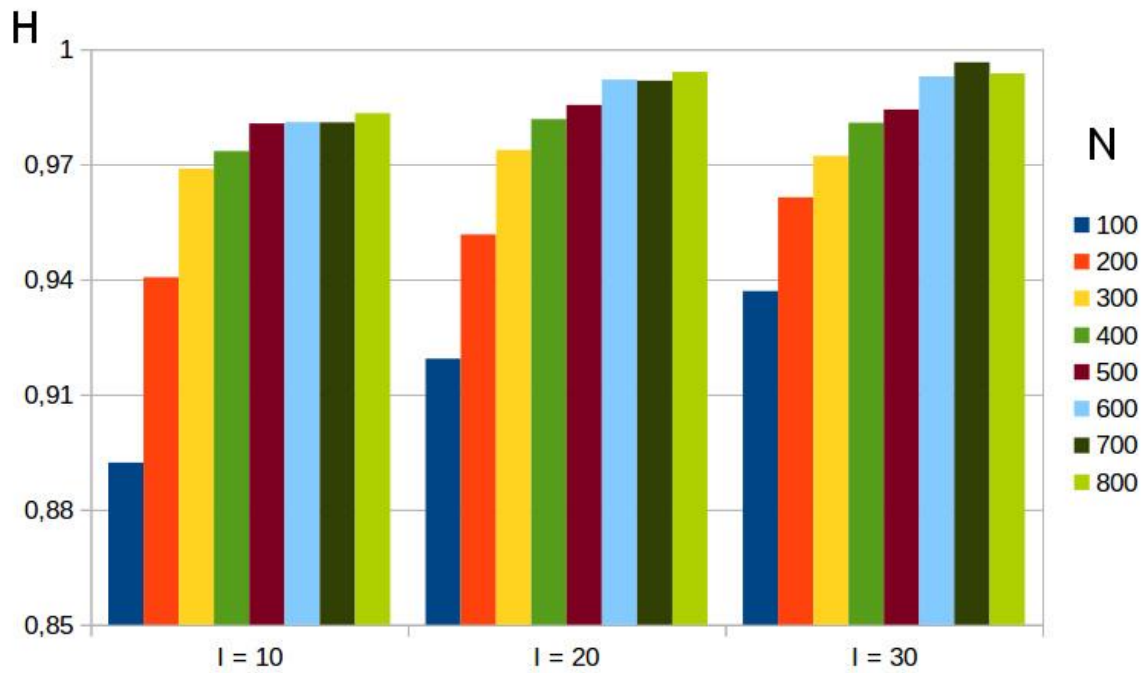


Figure 7 – Harmony diagrams of compositions for different number of iterations I and population size N

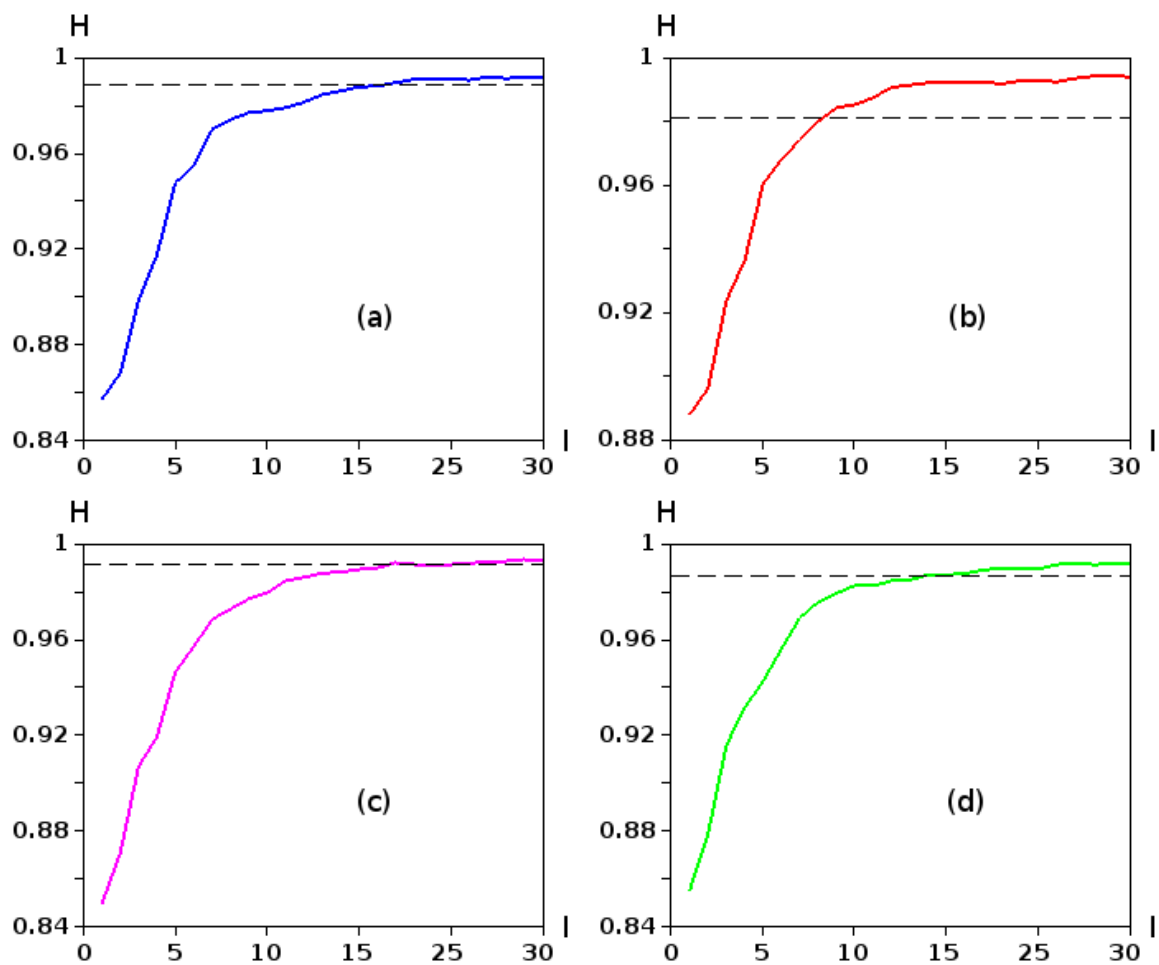


Figure 8 - The results of harmonization using the genetic algorithm for melodies from the compositions: (a) “There is a Balm in Gilead”, (b) “He Keeps me Singins”, (c) “Moment by Moment”, (d) “Jesus, Thy Boundless Love to Me”

Averaging the values of H over all values of I , we can calculate the average harmony value as a function of the iteration number I . The average harmony value for compositions of 6 chords, calculated from the author's music, is $H_{\text{author}} = 0.988$.

The relationship between I and H is summarized in the table (Table 3). This allows us to estimate the required number of iterations for determining the computational complexity of calculating the transition music for the desired level of harmony.

Having estimated all the components in the formula (7), we can determine the necessary computational performance M required to calculate α scenario with a given level of harmony H :

$$M = 19444700\alpha I(H). \quad (10)$$

Table 3. The relationship between the number of iterations I and the harmony H .

I	H	I	H
5	0,950	15	0,989
6	0,959	16	0,990
7	0,971	17	0,991
8	0,976	18	0,991
9	0,980	19	0,992
10	0,982	20	0,992
11	0,984	21	0,992
12	0,986	22	0,993
13	0,988	23	0,993
14	0,989	24	0,993

For example, for parameters: the number of scenarios $\alpha = 5$, for the required level of harmony $H = 0.988$, the required number of iterations will be $I = 13$ (according to the results of Table 3) and then required computational performance according to formula (9) will be about $M = 96$ million integer operations per second.

7. CONCLUSION

New method for music harmonization for adjacent transitional fragments in computer games is proposed. The method is based on using the tonal model of musical compositions and the genetic algorithm for finding the global extremum of the multi-extremal harmony function. In this sense, the goal of the work is shifted from the methods of searching for the extremum to determination of the correct parameters of the algorithm and preparation of the input data for it. At the moment, procedural

methods for generating music in video games are used rarely because of their high demands on computing resources. The proposed method allows for the practical use of theoretical results by harmonization methods. The method of computation of harmony transitions during the game is shown, a formula for calculating performance and required resources is obtained. The results of computational experiments prove the effectiveness of the developed approach.

The method allows for a reasonable computational cost ensuring smooth harmonious transitions between musical themes in a computer game. An additional advantage of the method is the use of mainly only integer operations, in contrast to the software of computer games, in which mainly floating-point operations are used. Together with the high computational parallelism of the proposed method, this significantly reduces the competition for processor resources between the game algorithm and the music generation algorithm.

Based on the analysis of the use of the tonal model and the genetic algorithm for the harmonization a number of well-known melodies, typical parameters values of the method were obtained. This made it possible to obtain the relationships for calculating method parameters depending on the number of game states, the temporal characteristics of the associated musical themes and the desired level of harmony.

The use of these relations allows balancing the computational load in the computer between the game algorithm itself and the musical accompaniment calculation algorithm.

An additional result of the work is the fact that the use of computer algorithms based on the tonal model and the genetic optimization algorithm makes it possible to obtain an even higher level of harmonization of musical composition than it can be obtained by professional composers.

8. REFERENCES

- [1] T. Adam, M. Haungs, F. Khosmood, "Procedurally generated, adaptive music for rapid game development," *Proceedings of Workshops Co-located with the 9th International Conference on the Foundations of Digital Games*, Florida, April 2014, pp. 1-7. [Online]. Available at: http://fdg2014.org/workshops/ggj2014_paper_02.pdf.
- [2] *Entertainment Software Association: Essential Facts About the Computer and Video Game Industry*. ESA, Washington, DC, 2018. [Online]. Available at: <http://www.theesa.com/>

- about-esa/essential-facts-computer-video-game-industry/
- [3] A. Berndt, K. Hartmann, "The functions of music in interactive media," in Spierling U., Szilas N. (eds.), *Interactive Storytelling. ICIDS 2008. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2008, vol. 5334, pp. 126-131, DOI: 10.1007/978-3-540-89454-4_19
- [4] M. Sweet, *Writing Interactive Music for Video Games: A Composer's Guide*, Addison-Wesley Professional, Piscataway, 2014, 512 p.
- [5] K. Collins, "An introduction to procedural music in video games," *Contemporary Music Review*, vol. 28, pp. 5-15, 2009, DOI: 10.1080/07494460802663983.
- [6] M. Hendriks, S. Meijer, J. van der Velden, A. Iosup, "Procedural content generation for games: A survey," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 9, pp. 1-22, 2013, DOI: 10.1145/2422956.2422957.
- [7] D. Plans, D. Morelli, "Experience-driven procedural music generation for games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, issue 3, pp. 192-198, 2012, DOI: 10.1109/TCIAIG.2012.2212899
- [8] W. Phillips, *A Composer's Guide to Game Music*, MIT Press, 2014, 288 p.
- [9] P. Peerdeman, *Sound and Music in Games*, VU Amsterdam, 2010.
- [10] I. Karydis, D. Makris, S. Sioutas, "Automatic melodic harmonization: an overview, challenges and future directions," in Kostagiolas P., Martzoukou K., Lavranos C. (eds.), *Trends in Music Information Seeking, Behavior, and Retrieval for Creativity*, IGI Global, 2016, pp. 146-165.
- [11] C.E. Palazzi, S. Ferretti, S. Cacciaguerra, Roccetti M. "On maintaining interactivity in event delivery synchronization for mirrored game architectures," *Proceedings of the IEEE Global Telecommunications Conference Workshops*, 2004, pp. 157-165. DOI: 10.1109/GLOCOMW.2004.1417568.
- [12] S. Ferretti, M. Roccetti, "Fast delivery of game events with an optimistic synchronization mechanism in massive multiplayer online games," *Proceedings of the International Conference on Advances in Computer Entertainment Technology ACE'2005*, 2005, pp. 405-412. DOI: 10.1145/1178477.1178570.
- [13] G. Armitage, "An experimental estimation of latency sensitivity in multiplayer Quake 3," *Proceedings of the 11th IEEE International Conference on Networks*, 2003, pp. 137-141. DOI: 10.1109/ICON.2003.1266180.
- [14] Pathologic Classic HD. [Online]. Available at: https://store.steampowered.com/app/384110/Pathologic_Classic_HD
- [15] C. Schmidt-Jones, *Understanding Basic Music Theory*, 12th Media Services, 2018, 278 p.
- [16] E. Geisera, E. Ziegler, L. Jancke, M. Meyer, "Early electrophysiological correlates of meter and rhythm processing in music perception," *Cortex*, vol. 45, pp. 93-102, 2009, DOI: 10.1016/j.cortex.2007.09.010.
- [17] I. Sposobin, *Elementary Theory of Music*, Moscow, 1996, 208 p. (in Russian).
- [18] Y.Y. Shi, X. Zhu, H.G. Kim, K.W. Eom, "A tempo feature via modulation spectrum analysis and its application to music emotion classification," *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo*, 2006, pp. 1085-1088. DOI: 10.1109/ICME.2006.262723.
- [19] P. Gomez, B. Danuser, "Relationships between musical structure and psychophysiological measures of emotion," *Emotion*, vol. 7, pp. 377-387, 2007. DOI: 10.1037/1528-3542.7.2.377.
- [20] H.V. Koops, J.P. Magalhaes, W.B. de Haas, "A functional approach to automatic melody harmonization," *Proceedings of the first ACM SIGPLAN Workshop on Functional Art, Music, Modeling & Design*, 2013, pp. 47-58. DOI: 10.1145/2505341.2505343.
- [21] A. Zacharakis, M. Kaliakatsos-Papakostas, C. Tsougras, E. Cambouropoulos, "Musical blending and creativity: An empirical evaluation of the CHAMELEON melodic harmonisation assistant," *Musicae Scientiae*, vol. 22, pp. 119-144, 2018, DOI: 10.1177/1029864917712580.
- [22] F. Simonetta, F. Carnovalini, N. Orio, A. Roda, "Symbolic music similarity through a graph-based representation," *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion, (AM'18)*, September 12-14, 2018, Wrexham, United Kingdom, pp. 26:1-26:7, DOI: 10.1145/3243274.3243301.
- [23] D. Deutsch, J. Feroe, "The internal representation of pitch sequences in tonal music," *Psychological Review*, vol. 88, pp. 503-522, 1981, DOI: 10.1037/0033-295X.88.6.503.
- [24] S. Phon-Amnuaisuk, A. Tuson, G. Wiggins, "Evolving musical harmonization," in:

Pearson D.W., Dobnikar A., Albrecht R. (eds.), *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, Portorož, Slovenia, 1999, pp. 229-234. Springer-Verlag Wien, Vienna, 1999, DOI: 10.1007/978-3-7091-6384-9_39.

- [25] G. Papadopoulos, G. Wiggins, "AI methods for algorithmic composition: A survey, a critical view and future prospects," *Proceedings of the AISB'99 Symposium on Musical Creativity*, 1999, pp. 110-117.
- [26] A.R.R. Freitas, F.G. Guimaraes, "Melody harmonization in evolutionary music using multiobjective genetic algorithms," *Proceedings of the 8th Sound and Music Computing Conference. Sound and Music Computing Network*, 2011, pp. 346-353. [Online]. Available at: <http://www.alandefreitas.com/en/papers/melody-harmonization-in-evolutionary-music-using-multiobjective-genetic-algorithms-2011-padua>
- [27] S. Phon-Amnuaisuk, G. Wiggins, "The four-part harmonisation problem: A comparison between genetic algorithms and a rule-based system," *Proceedings of the Symposium on Musical Creativity AISB'99*, 1999, pp. 22-34.
- [28] D.A. Matic, "Genetic algorithm for composing music," *Yugoslav Journal of Operations Research*, vol. 20, pp. 157-177, 2010, DOI: 10.2298/YJOR1001157M.
- [29] J.D. Fernandez, F. Vico, "AI methods in algorithmic composition: A comprehensive survey," *Journal of Artificial Intelligence Research*, vol. 48, pp. 513-582, 2013, DOI: 10.1613/jair.3908.
- [30] Yu.N. Kholopov, *Harmony. A Practical Course*, Moscow, Publishing house "Composer", 2005, 472 p. (in Russian).
- [31] I. Dubovsky, S. Evseev, I. Sposobin, V. Sokolov, *A Textbook of Harmony*, Moscow: Muzyka, 1987, 436 p. (in Russian).
- [32] D. Moelants, "Preferred tempo reconsidered," In: Stevens C. J., Burnham D. K., McPherson G., Schubert E., Renwick J. (eds.) *Proceedings of the 7th International Conference on Music Perception and Cognition*, Causal Productions, Sydney, Adelaide, 2002, pp. 580-583.
- [33] T. Jones, *Evolutionary Algorithms, Fitness Landscapes and Search*, 1995, 249 p. [Online]. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.4779&rep=rep1&type=pdf>
- [34] G.R. Harik, F.G. Lobo, "A parameter-less genetic algorithm," in: Banzhaf W.,

Daida J.M., Eiben A.E., Garzon M.H., Honavar V. (eds.) *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation GECCO'99*, 1999, pp. 258-265.

- [35] There is a Balm in Gilead. [Online]. Available at: <https://hymnary.org/hymn/BH2008/119>.
- [36] He Keeps me Singing. [Online]. Available at: <https://hymnary.org/hymn/BH2008/148>
- [37] Moment by Moment. [Online]. Available at: <https://hymnary.org/hymn/BH2008/159>
- [38] Jesus, Thy Boundless Love to Me. [Online]. Available at: <https://hymnary.org/hymn/BH2008/166>



Oleksandr Komarov graduated from Odessa National Polytechnic University with speciality "Information Control Systems and Technologies" in 2017. At the moment he is working on the Ph.D. degree at the Department of Information Systems of the Institute of Computer Systems of Odessa National Polytechnic University. His areas of scientific interests include artificial intelligence, evolution systems, machine learning.



Oleg Galchonkov received the Dipl. Ing. degree from the Odessa Polytechnic Institute in 1977 and received Candidate of Science degree (Ph.D) in Electrical Engineering from Leningrad Electrotechnical Institute (St. Petersburg, USSR), in 1987. He spent 33 years in industry, with Design Organization "Diskret", Golden Telecom, MMDS Ukraine before joining Institute of Computer Systems ONPU in 2010, as an assistant professor. His current research focus is on adaptive antenna processing, digital signal processing, adaptive noise compensation, artificial intelligence systems.



Alexander Nevrev graduated from Odessa Polytechnic Institute in 1975. He defended a Ph.D. thesis in the Radio engineering Institute of USSR Academy of Science (Moscow) in 1986. He spent 36 years in industry, with Design Organization "Diskret" and Golden Telecom. Alexander currently works as an Associate Professor at the Department of Information Systems at the Odessa National Polytechnic University,

Ukraine. His research interests include digital information processing, adaptive noise cancellation, artificial intelligence systems, evolution systems, pattern recognition, machine learning.



Oksana Babilunga received engineer qualification in the specialty organization of automated processing of economic information in Odessa Polytechnic Institute, Ukraine. She was working in branch research laboratory for non-destructive testing as a research fellow. At 2008 she received Ph.D. degree

by specialty systems and instruments of an artificial intelligence in Odessa National Polytechnic University. Her scientific thesis was devoted to the increasing speed of intelligence processing and pattern recognition systems by development of iterative methods of the hierarchical contour segmentation images at the hyperbolic wavelet transformation space. Oksana currently works as an Associate Professor at the Department of Information Systems at the Odessa National Polytechnic University, Ukraine. Her research interests include digital information processing, pattern recognition, artificial intelligence systems.