Creating a New Lullaby Using an Automatic Music Composition System in Collaboration with a Musician

So Hirawata[1], Noriko Otani[1], Daisuke Okabe[1], and Masayuki Numao[2] ⋆

[1] Faculty of Informatics, Tokyo City University
[2] Institute of Scientific and Industrial Research, Osaka University
otani@tcu.ac.jp

Abstract. Many parents often have problems with getting their children to sleep. A publishing company planned to produce a promotional video consisting of pictures from their published book and a new lullaby with a sleep-inducing effect. They requested that the new lullaby would be created through a collaboration of an automatic composition system and a musician. In our previous work, a melody generation method has been proposed to support the creative activities of a musician. However, this method requires too much intervention by a musician to meet the publisher's requirements. In this paper, we propose an automatic composition system that generates a new piece with a chord progression only by specifying some existing pieces. A case study is presented in which a professional musician completed a lullaby based on the piece generated by the proposed system.

Keywords: Lullaby, Music Composition, Symbiotic Evolution

1 Introduction

In a survey of 550 mothers of under-three-year-olds conducted by Interspace Co., Ltd. and Hakuhodo Inc., 66.4 % of the respondents reported having problems with their children's sleep. In addition, 71.3 % answered that they were stressed about getting their children to sleep, and 64.3 % answered that they were troubled about it. The top 3 methods the mothers used to get their children to sleep were pretending to sleep next to them, lying down with and watching over them, and sleeping with them. Although singing to children not only soothes them, but also activates maternal love, increases parental motivation, and improves the quality of parental behavior [3], only 21.5 % and 20.4 % said they read picture books and sang songs, respectively, a behavior that is often criticized in todays' parents. However, according to Takamatsu's [12] survey of 337 parents of 18-month-olds, 87.0 % of the respondents reported having sung songs to put their children to sleep.

⋆ We would like to express our deepest gratitude to all the children and their parents for their cooperation in confirming the sleep-inducing effects, Tokyo City University Futako Kindergarten, Toho Co., Ltd., and Crimzon Technology, Inc.

A sleep-inducing book written in 2015 by Carl-Johan Ehrlin, a psychologist, titled *The Rabbit Who Wants To Fall Asleep*, has become a global hot topic. It has been reported that when parents read this book aloud to their children at bedtime, the children fall asleep more easily. In Japan, a picture book written in 2020 by NOBU titled *Dream Rescue*, which features tapirs who help children with bad dreams, was published. Prior to the book's release, the publishing company planned to produce a promotional video consisting of pictures from the book and a new lullaby with a sleep-inducing effect in an effort to assist children with sleep and reduce parents' burden. According to the publisher's request, the new lullaby would be created through an AI-human collaboration; the music would be generated using an AI-based automatic composition system while humans would be responsible for writing lyrics, playing, and singing.

In our previous work[7], a melody generation method has been proposed to support the creative activities of musicians while satisfying the clients' requirements using a music composition system. The method is based on a constructive adaptive user interface (CAUI) [6, 4], whose goal is to compose music that arouses a particular sensibility in the listener. In order to reflect a musician's creativity and intention in the overall atmosphere of the music, the musician selects some existing pieces and uses them as training data to induce sensibility models. In addition, the musician specifies a chord progression, the pitch extent, and the length of a new piece, and then a melody is generated based on the specified contents and the sensibility models. The steps of generating a short melody are repeated until the musician is satisfied. The musician selects suitable chord progressions and melodies, arranges them, writes the lyrics, and finally obtains a complete piece. The effectiveness of this method has been evaluated by subjective experiments and two case studies involving collaborative work with professional musicians. However, this approach requires too much intervention by the musician to meet the publisher's requirements described in the previous paragraph.

In this paper, we propose an automatic music composition system that generates a new piece with a chord progression only by specifying some existing pieces, that follows basically our previous work [7]. Using the proposed system, we aim to create a new sleep-inducing lullaby that meets the publisher's requirements and assists parents put their children to sleep. We also present a case study in which a professional musician completed a lullaby based on the piece generated by the proposed system.

2 Music Composition Flow

The music composition flow of the proposed system is illustrated in Fig. 1. Some existing pieces are needed as the training dataset. The pieces included in the training dataset and the pieces generated by the system consists of a chord progression and a melody with a 4/4 time signature. The basic duration of a note or rest in a melody is defined as the duration of a sixteenth note. The basic duration of a chord in a chord progression is defined as the duration of a quarter note. A motif, which is the minimum unit of a piece, is set to two bars, and a piece is represented as a sequence of multiple motifs.

First, existing pieces are specified as the training dataset according to the user's sensibilities, aims, and/or the purpose of the intended composition. Beats per minute (BPM) is set to a value randomly selected from a $\pm 0.5\sigma$ range of all the pieces' BPM

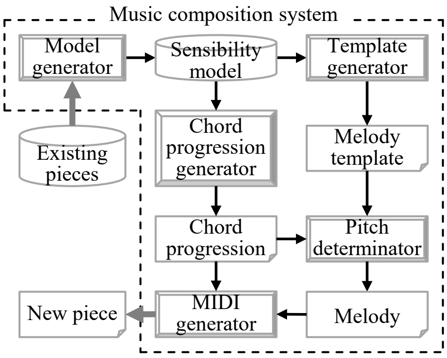

Fig. 1. Music composition flow

distributions in the training dataset. Sensibility models for the chord progression and the melody are obtained based on the training dataset. The next step is to generate a chord progression and a melody template that adapts to the sensibility models and the basic music theory. A melody template indicates the time at which each sound in the melody is played, the length of time each sound is played in succession, and the up-and-down stream of the melody line. In other words, a melody template is a melody without the pitch of each note. Subsequently, the pitch of each note in a melody is determined using the melody template and chord progression. Finally, the chord progression and the melody are combined and output in the form of a MIDI file.

Bainbridge [1] has shown that adult listeners accurately identify unfamiliar lullabies as infant-directed based on their musical features alone, and that infants relax more to unfamiliar foreign lullabies than to non-lullaby foreign songs. They suggested that infants might be predisposed to respond to common features of lullabies. Therefore, to generate a new lullaby using the proposed system, the characteristics of existing lullabies are regarded and used as sensibility models and some traditional lullabies are used as the training data. The generated output piece is then handed over to the musician, who modifies the melody, writes the lyrics, and completes the lullaby.

3 Sensibility Models

A sensibility model comprises a partial music structure that affects the user's particular sensibility or reflects their intentions and is represented by a set of patterns that are common to the pieces in the training dataset. In the proposed system, four sensibility models are induced: one each for the rhythm of the chord progression, chord name of the chord progression, rhythm of the melody, and up-and-down stream of the melody.

3.1 Training Dataset

To induce frequent patterns in existing pieces, the rhythms and chord names of the chord progressions, and the rhythms and up-and-down streams of the melodies in the training dataset, are represented as element sequences or chunk sequences.

The rhythm of the chord progression is represented as a sequence of two types of elements, "beat" and "-," for each motif. Each element represents the state of a note for one beat. The "beat" and "-" elements respectively indicate that a chord is played and the duration of the previous note is extended. The chord name of the chord progression is represented as a chunk sequence. Each chunk indicates one chord and consists of three elements: a degree-notated root note and type pair, tension, and degree-notated on-chord. If there is no tension and no on-chord, the element is "+."

The rhythm of the melody is represented as a sequence of chunks for each motif. Each chunk indicates one beat rhythm, and consists of four elements. Each element represents the state of a note for 1/4 beat, and can be one of "beat," "-," and "null." The elements "beat," "-," and "null' respectively indicate that a sound is played, the duration of the previous note or rest is extended, and no sound is played. The up-and-down stream of the melody is represented as a sequence of elements in which each note other than the first is replaced by an "up," "down," or "flat" profile for each motif. Each element indicates the change in pitch from the previous note: "up" means higher, "down" means lower, and "flat" means the same pitch.

An example of element sequences and chunk sequences is shown in Fig. 2. Chunks are enclosed in parentheses. For the rhythm of the chord progression and the up-and-down stream of the melody, two element sequences are generated from the score, respectively. One chunk sequence for the chord name of the chord progression and two chunk sequences for the rhythm of the melody are generated from the score.

◊Rhythm of the chord progression

1: beat,-,-,beat,beat,-,-,-

2: beat,-,beat,-,beat,-,-,-

◊Chord name of the chord progression

(IV,+,+),(I,+,III),(IIm7,+,+),(IIm7,+,V),(V,+,+),(I,+,+)

◊Rhythm of the melody

1: (beat,-,-,-),(beat,-,beat,-),(beat,-,-,-),(beat,beat,beat,beat),(beat,beat,beat,beat),(beat,-,beat,-), (beat,null,beat,-),(beat,-,beat,beat)

2: (beat,-,-,-),(-,-,-,beat),(beat,-,beat,-),(beat,beat,beat,beat),(-,-,-,-),(-,-,-,-),(-,-,-,-),(-,-,-,-)

◊Up-and-down stream of the melody

1: flat,up,down,flat,up,down,down,down,flat,flat,up,up,down,flat,down,down,up,flat

2: flat,flat,flat,up,down,down,flat,flat

Fig. 2. Examples of element sequences and chunk sequences

3.2 Induction of Sensibility Models

A "don't-care" element "*" that is considered equal to any element is introduced in obtaining a sensibility model for the rhythm of the chord progression. For any element sequence s_i with i elements in an element sequence in the training dataset replaced by "*" ($0 \leq i \leq D_{max}$), the set of element sequences in the training dataset contains more than F_{min} element sequences in which all elements are equal except for "*" F_{min} or more, then s_i is defined as a frequent pattern. Frequent patterns are extracted for all element sequences and used as sensibility models for the rhythm of the chord progression . In a chunk sequence of the chord names of the chord progression, a sequence of any consecutive chunks is called a chunk subsequence. Of all the chunk subsequences in the training dataset, those with a length of L_{min} or more and occurrence of more than F_{min} times in the training dataset are extracted as the sensibility models for the chord names of the chord progression. In addition, to replace the on-code with "+" in all chunk sequences and replace tension and on-code with "+," the chunk subsequences with a length of L_{min} or more and occurrence of more than F_{min} times are also extracted as the sensibility models for the chord name of the chord progression.

In a chunk sequence of the rhythm of the melody, a sequence of any consecutive chunks is called a chunk subsequence. Of all the chunk subsequences in the training dataset, those with a length of L_{min} or more and occurrence of more than F_{min} times in the training dataset are extracted as the sensibility models for the rhythm of the melody.

In an element sequence of the up-and-down stream of the melody, a sequence of any consecutive elements is called an element subsequence. Of all the element subsequences in the training dataset, those with a length of L_{min} or more and occurrence of more than F_{min} times in the training dataset are extracted as the sensibility models for the up-and-down stream of the melody.

To shorten the processing time, the PrefixSpan approach [10] is adopted to induce the sensibility models.

4 Chord Progression and Melody

Symbiotic evolution is applied to generate a chord progression and melody template. This section describes the characteristics of symbiotic evolution and how it is applied to generate a chord progression and melody template.

4.1 Symbiotic Evolution

Symbiotic evolution is an evolutionary computation algorithm that was proposed for forming neural networks [5]. This algorithm results in a fast, efficient search and prevents convergence to suboptimal solutions. It is characterized by maintaining two separate populations: a partial solution population, the individuals of which represent partial solutions, and a whole solution population, the individuals of which are combinations of individuals in the partial solution population and represent whole solutions. In the former population, partial solutions that may be components of the optimal whole solution are generated. In the latter population, combinations of the partial solutions that may be the optimal solution are generated.

A piece of music can be considered a combination of motifs; it is essential to find motifs that may be contained in a suitable piece of music as well as a suitable combination of motifs. As symbiotic evolution is appropriate for generating a piece of music owing to its suitable characteristics, a chord progression and a melody template are generated based on symbiotic evolution in the proposed system. Each motif in a chord progression or melody template is expressed as an individual in the partial solution population, and a whole chord progression or a whole melody template is expressed as an individual in the whole solution population. When generating a piece of $2N$ bars, a chromosome of the whole solution individual is expressed as a pointer sequence to the N partial solution individuals.

Individuals of both whole and partial solution populations in the next generation are generated using the GA operators: two-point crossover and mutation. The partial solution population is evolved with the strategy described in [5]. The minimal generation gap model [11], which is an effective evolution strategy for avoiding early convergence, is applied to the whole solution population.

After generating the partial and whole solution populations of the initial generation, alternation in all the partial and whole solution populations and evaluation of all the whole and partial solution individuals are repeated a specified number of times. Finally, the sequence of the genes of the partial solution individuals pointed out by the best whole solution individual is generated as the output.

4.2 Generation of Chord Progression

Chord progression is generated to adapt to the sensibility models for the rhythm and the chord name of the chord progression using chords contained in the training dataset.

When there are R types of degree-notated root notes and type pairs in the training dataset, each pair is called $rt_1 - rt_R$. The set of tensions of chords whose root note and type pair is rt_i is called T_i, and the set of on-chords is called O_i. A chromosome of a partial solution individual has 24 genes that include 8 root-type genes, 8 tension genes, and 8 on-chord genes. The first root-type gene is a natural number less than or equal to R. The second and subsequent root-type genes are natural numbers less than or equal to R or 0, with 0 representing "-" and a non-zero value i representing rt_i. The tension genes and on-chord genes are 0 or 1 to represent the presence or absence of tension and on-code, respectively. When the root-type gene is a non-zero value i and the tension gene or on-chord gene is 1, the tension and on-chord of the chord are selected from T_i and O_i, respectively, according to their frequency of occurrence in the training dataset.

After the tail of a piece represented by a whole solution individual is converted to the perfect cadence regardless of the gene value, the fitness value $f_{cw}(W_c)$ of a whole solution individual W_c is calculated using (1).

$$f_{cw}(W_c) = \sum_{W_c \to P_c} \{f_{cr}(P_c) + f_{ct}(P_c)\} + f_{cn}(W_c) + f_{ct}(W_c) \ . \quad (1)$$

where $W_c \to P_c$ means that a partial solution individual P_c is pointed out by a whole solution individual W_c. The function $f_{ct}(P_c)$ indicates the degree of adaptability of a partial solution individual P_c to the music theory, and the function $f_{ct}(W_c)$ indicates the degree of adaptability of a whole solution individual W_c to the music theory. The function $f_{cr}(P_c)$ indicates the degree of adaptability of a partial solution individual P_c to the sensibility model for the rhythm of the chord progression, while $f_{cn}(W_c)$ indicates the degree of adaptability of a whole solution individual W_c to the sensibility model for the chord name of the chord progression. The two are calculated using (2) and (3), respectively.

$$f_{cr}(P_c) = \sum_{e \in S_{cr}(P_c)} \{bn(e, P_c) \cdot fq_e(e)\} \ . \tag{2}$$

$$f_{cn}(W_c) = \sum_{c \in S_{cn}(W_c)} \{ln_c(c) \cdot fq_c(c)\} \ . \tag{3}$$

where $S_{cr}(P_c)$ is the set of element sequences that were extracted as the sensibility model for the rhythm of the chord progression and contained in P_c. $S_{cn}(W_c)$ is the set of chunk sequences that were extracted as the sensibility model for the chord name of the chord progression and contained in W_c. $bn(e, P_c)$ is the number of beats other than the "don't-care" ones that an element sequence e covers in P_c, $fq_e(e)$ is the frequency with which e appears in the training dataset, $ln_c(c)$ is the length of a chunk sequence c, and $fq_c(c)$ is the frequency with which a chunk c appears in the training dataset.

A partial solution individual is evaluated using whole solution individuals that point to the partial solution individual. The fitness value $f_{cp}(P_c)$ of P_c is the largest fitness value of these whole solution individuals, as given by (4). The partial solution individual receives a higher evaluation when it is pointed to by a better whole solution individual.

$$f_{cp}(P_c) = \frac{1}{N} \max_{W_c \to P_c} f_{cw}(W_c) + f_{cr}(P_c) + f_{ct}(P_c) \ . \tag{4}$$

4.3 Generation of Melody Template

A melody template is generated to adapt to the sensibility models for the rhythm and up-and-down stream of a melody. A chromosome of a partial solution individual has 32 genes. Each gene is -1, 0, 1, 2, or 3, which mean "rest," "extend," "beat + down," "beat + flat," and "beat + up" respectively.

The fitness value $f_{mw}(W_m)$ of a whole solution individual W_m is defined by (5).

$$f_{mw}(W_m) = \sum_{W_m \to P_m} f_{mm}(P_m) \times \alpha^{k(W_m)} \ . \tag{5}$$

where $W_m \to P_m$ means that a partial solution individual P_m is pointed by a whole solution individual W_m. α is a parameter greater than 1 that promotes a longer phonetic value of the last note in the melody. Let $k'(W_m)$ be the number of genes 0 following the end of the whole solution individual W_m, then $k(W_m)$ is calculated by (6).

$$k(W_m) = \begin{cases} k'(W_m) & (k'(W_m) \leq 3) \\ 4 & \text{(otherwise)} \end{cases} \ . \tag{6}$$

The function $f_{mm}(P_m)$ indicates the degree of adaptability of a partial solution individual P_m to the sensibility models, and is calculated using (7).

$$f_{mm}(P_m) = f_{mr}(P_m) + \frac{1}{4} f_{mu}(P_m) \ . \tag{7}$$

The functions $f_{mr}(P_m)$ and $f_{mu}(P_m)$ indicate the degree of adaptability to the sensibility model for the rhythm and up-and-down stream of the melody, respectively, and are calculated using (8) and (9).

$$f_{mr}(P_m) = \sum_{c \in S_{mr}(P_m)} \Big[\{ln_c(c)\}^2 \cdot fq_c(c) \Big] \ . \tag{8}$$

$$f_{mu}(P_m) = \sum_{e \in S_{mu}(P_m)} \{ln_e(e) \cdot bn(e, P_m) \cdot fq_e(e)\} \ . \tag{9}$$

where $S_{mr}(P_m)$ is the set of chunk sequences that were extracted as the sensibility model for the rhythm of the melody and contained in P_m. $S_{mu}(P_m)$ is the set of element sequences that were extracted as the sensibility model for the up-and-down stream of the melody and contained in P_m. $ln_c(c)$ is the length of a chunk sequence c, $fq_c(c)$ is the frequency with which a chunk c appears in the training dataset, and $bn(e, P_m)$ is the number of beats that an element sequence e covers in P_m.

The fitness value $f_{mp}(P_m)$ of a partial solution individual P_m is the largest fitness value of these whole solution individuals, as given by (10).

$$f_{mp}(P_m) = \frac{1}{N} \max_{W_m \to P_m} f_{mw}(W_m) + f_{mm}(P_m) \ . \tag{10}$$

4.4 Determination of Pitch of Notes in the Melody

In the score of the generated melody, notes are placed at the position where the melody template value is 1 to 3, that is, "beat + down," "beat + flat," and "beat + up." The pitch of each note in the melody is determined based on the generated melody template. First, a pitch candidate set is prepared according to the scale of the tonality and pitch extent. In determining the pitch of a note that is played at the same time as a chord, discords of the chord are deleted from the pitch candidate set.

The pitch of each note is an element of the pitch candidate set. The pitch of the first note in a motif is chosen at random from the pitch candidate set. The pitch of a "beat + flat" note is set to be the same as that of the previous note. The pitch of a "beat + up" note is set to the lowest pitch among the pitches of the pitch candidate set that are higher than that of the previous note. The pitch of a "beat + down" note is set to the highest pitch among the pitches of the pitch candidate set that are lower than that of the previous note. If the target pitch is not contained in the pitch candidate set, the nearest pitch in the pitch candidate set is chosen. In addition, the pitch of the last note in the piece is set to the lowest key pitch among pitches that are higher than that of the previous note for "beat + up," and the highest key pitch among pitches that are lower than that of the previous note for "beat + flat" or "beat + down."

5 Case Study

5.1 Creation of a New Lullaby

A new lullaby was created according to the procedure described in Section 2 with the help of a professional musician who is a member of a Japanese pop duo . The 60 traditional lullabies collected for the training data were classified according to the four criteria listed below. There were 48 categories in total, 22 of which at least one traditional lullaby was classified to. Possible values for each criterion are given in parentheses. “Structure” refers to whether the lullaby is constructed as a one-part or two-part song divided by the rehearsal mark.

– Tonality (Major / minor)
– Structure (one-part / two-part)
– Tempo (slow / medium / fast)
– Rhythm pattern (bounce / four on the floor / eight beat / sixteen beat)

In total 22 pieces were created, with one piece in each category using the lullabies classified to that category. For one-part structure category, a piece with eight bars was generated. For two-part structure category, the first part of the piece was generated using the first part of the lullabies classified to that category. In the same way, the second part was generated using the second part of the lullabies classified to that category. Then the first and second parts were joined to form a piece of sixteen bars. The members of the publisher selected one of pieces: a two-part minor with a slow tempo, and bounce rhythm pattern, and commissioned it to the musician. The musician changed the piece as follows.

1. The first and second bars were moved to the seventh and eighth bars.
2. The third and fourth bars were moved to the first and second bars.
3. The seventh and eighth bars were moved to the third and fourth bars.
4. The pitch of the second and third notes in the first bar were raised by one tone.

The musician wrote the lyrics to this melody and sang the song with the partner of the duo. A promotional video with this song was created and published on the picture book’s website. [3] A warning “Do not play while driving” is attached to the video.

5.2 Effect on Falling Asleep

Parents with children under the age of six were asked to play the new lullaby while putting their children to sleep and to report the children’s sleeping behavior and their observations each time they did this. The 58 participants’ usual average time to fall asleep by age, the need to put the child to sleep, and the method of putting the child to sleep, which were obtained in the preliminary survey, are shown in Fig. 3, 4, and 5, respectively. Multiple answers were allowed for the latter two measures.

According to the results, 67.2 % of children, regardless of age, took more than 16 minutes to fall asleep. In fact, of the 58, only two three-year-olds, one at four-year-old, seven five-year-olds, and three six-year-olds could fall asleep by themselves; assistance from their parents or other adults was not needed. These results indicate the importance of reassuring the children by snuggling with them. They also reveal that 17.2 % of participants use music to put their children to sleep.

[3] https://yume-rescue.com/

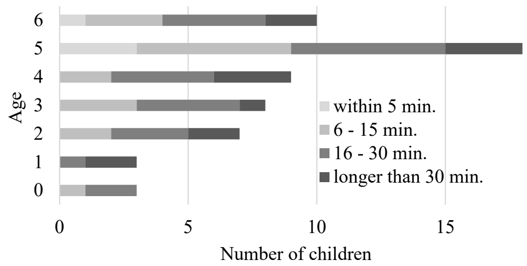

Fig. 3. Usual average time to fall asleep by age

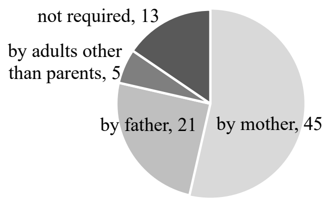

Fig. 4. The need to put the child to sleep

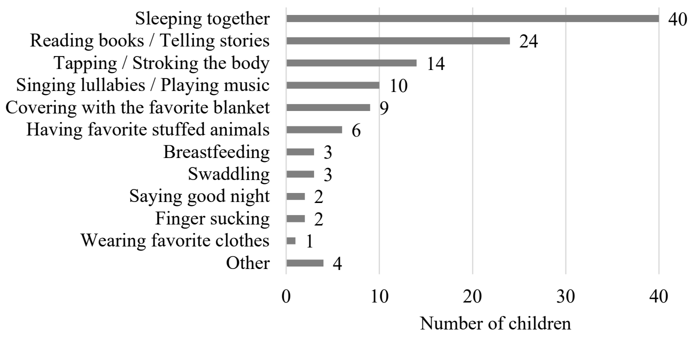

Fig. 5. Method of putting the child to sleep

Each participant used the lullaby 1-9 times to put their children to sleep, producing a total of 126 responses. The children's sleeping behaviors while using the new lullaby are shown in Fig. 6. Although the lullaby was not effective for all of the children, it was effective for the majority of them.

Out of the 126 responses collected, 94 included free descriptions regarding the children's sleeping behavior, such as "Although there was no change in bedtime, a change in being able to go to bed alone without needing to be rocked to sleep continued." These responses were then analyzed using SCAT (Steps for Coding and Theorization) [8, 9]. SCAT is a qualitative data analysis method that weaves themes and constituent concepts that emerge from four steps of coding into a story line and theory. In qualitative research, coding refers to the task of assigning "codes" to text data and codes are concepts that make up the text. The four steps of coding in SCAT are as follows.

1. Identify key phrases in the data
2. Substitute those phrases with phrases outside of the data
3. Provide explanations for those phrases using concepts outside of the text data
4. Identify and conceptualize the themes that have surfaced during the steps 1-3

As a result, it was found that this lullaby has a relaxing effect, but the support of a person the child trusts may be necessary to ensure this effect, and there may be differences in how the length of the song is felt depending on the time it takes to fall asleep. There may also be differences in the sleep-inducing effect depending on the growth stage of the child. If the lullaby could cause the children to fall asleep by conditioned reflex when it is played, then it can be expected that the relaxing effect has improved and that nightmares could possibly be avoided.

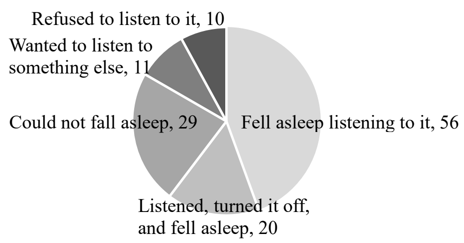

Fig. 6. Sleeping behavior while using the new lullaby

6 Conclusion

In this study, we proposed a new automatic music composition system. This system is not specific to any particular genre, but rather generates music that is based on a personal sensibility. As it can generate melodies and chord progressions simply by specifying some existing pieces of music as the training dataset, it is suited for providing musicians with a basis for their creative activities. In addition, by setting music pieces of a specific genre or existing music pieces that have characteristics that one wants to incorporate in a new piece as the training dataset, it is possible to generate genre-specific or various purpose pieces. A sensibility model that shows characteristics common to training data is not a black box, but it is made explicit, which also helps to identify characteristics.

Here, the proposed system was used to generate a new lullaby. The musicians were able to complete the lullaby without significantly modifying the system-generated piece, and a trial study with under-six-year-old children showed that the lullaby was effective in putting children to sleep to some extent.

In the future works, the target pieces, both inputted into the system and generated by the system, will be expanded to include elements, such as time signatures and the basic durations of notes. In addition, it is important to confirm the effectiveness of the system when applied to various musical genres.

References

1. Bainbridge, C.M., Bertolo, M., Youngers, J., Atwood, S., Yurdum, L., Simson, J., Lopez, K., Xing, F., Martin, A., Mehr, S.A.: Infants Relax in Response to Unfamiliar Foreign Lullabies. Nature Human Behaviour, 5(2), 256–264 (2021)
2. Ehrlin, C.F.: The Rabbit Who Wants to Fall Asleep: A New Way of Getting Children to Sleep. Crown Books for Young Readers, New York (2015)
3. Kobayashi, N., Matsui, I., Tanimura, M.: Parenting and Lullabies (in Japanese). Perinatology, 23(6), 885–889 (1993)
4. Legaspi, R., Hashimoto, Y., Moriyama, K., Kurihara, S., Numao, M.: Music Compositional Intelligence with an Affective Flavor. Proceedings of the 12th International Conference on Intelligent User Interfaces, 216–224 (2007)
5. Moriarty, D.E., Miikkulainen, R.: Efficient Reinforcement Learning through Symbiotic Evolution. Machine Learning, 22(1), 11–32 (1996)
6. Numao, M., Takagi, S., Nakamura, K.: Constructive Adaptive User Interfaces –Composing Music Based on Human Feelings. Proceedings of the Eighteenth National Conference on Artificial Intelligence, 193–198 (2002)
7. Otani, N., Okabe, D., Numao, M.: Generating a Melody Based on Symbiotic Evolution for Musicians' Creative Activities. Proceedings of the Genetic and Evolutionary Computation Conference 2018, 197–204 (2018)
8. Otani, T.: A Qualitative Data Analysis Method SCAT by Four-Step Coding – Theorization Process That is Easy Startable and Applicable to Small Datasets –. Bulletin of the Graduate School of Education and Human Development, Nagoya University, 54, 27–44 (2008)
9. Otani, T.: SCAT: Steps for Coding and Theorization – a Qualitative Data Analysis Method That is Easy Startable with Explicit Procedures and Applicable to Small Datasets –. Kansei Engineering, 10(3), 155–160 (2011)
10. Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: Mining Sequential Patterns by Pattern-Growth: the Prefixspan Approach. IEEE Transactions on Knowledge and Data Engineering, 16(11), 1424–1440 (2004)
11. Satoh, H., Yamamura, M., Kobayashi, S.: Minimal Generation Gap Model for Gas Considering both Exploration and Exploitation. Proceedings of the 4th International Conference on Soft Computing, 494–497 (1996)
12. Takamatsu, A.: Lullabies Today (in Japanese). Memoirs of the Faculty of Education and Regional Studies, Fukui University, Series VI, Fine arts & music, physical education, 35, 1–19 (2002)