GTTM DATABASE AND MANUAL TIME-SPAN TREE GENERATION TOOL

Masatoshi Hamanaka
RIKEN
masatoshi.hamanaka@riken.jp

Keiji Hirata
Future University Hakodate
hirata@fun.ac.jp

Satoshi Tojo
JAIST
tojo@jaist.ac.jp

ABSTRACT

This paper describes the publication of a musical structure analysis database and a tool for manually generating time-span trees on the basis of the generative theory of tonal music (GTTM). We previously analyzed 300 pieces of music with the analysis database and the analysis editor on the basis of the GTTM. However, the length of each piece was about eight bars (which is short), and the conventional analysis editor did not work for pieces of music with a large number of bars and notes, which thus took a lot of time to manually edit. Therefore, we developed a tool for manually generating time-span tree analyses for the GTTM that can manipulate a piece of music with a large number of bars and notes. Four musicologists developed an analysis database of 50 musical pieces of 32 bars in length by using the manual generation tool. The experimental results show that the average editing time with the time-span tree generation tool is shorter than that with the previous tool.

1. INTRODUCTION

For over 15 years, we have been implementing music analyzers on the basis of the generative theory of tonal music (GTTM), which was proposed by Lerdahl and Jackendoff [1–5]. The GTTM is composed of four modules, each of which assigns a separate structural description to a listener's understanding of a piece of music. The GTTM outputs a grouping structure, a metrical structure, a time-span tree, and a prolongational tree (Fig. 1).

The grouping structure is intended to formalize the intuitive belief that tonal music is organized into groups that com-prise subgroups. The metrical structure presents the rhythmical hierarchy of the musical piece by identifying the position of strong beats at the levels of quarter notes, half notes, one measure, two measures, four measures, and so on. The time-span tree is a binary tree and has a hierarchical structure presenting the relative structural importance of notes that differentiate the essential parts of the melody from the ornamentation. The prolongational tree is a binary tree that expresses the structure of tension and relaxation in a piece of music.

The time-span tree obtained as a result of GTTM analysis is used for melody morphing [6], melody prediction [7], and music summarization [8]. Therefore, we have developed a manual editor for the GTTM and have analyzed and released 300 pieces of analyzed data [9].

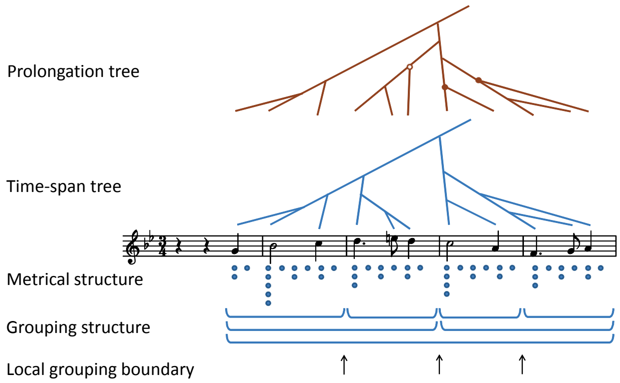

Figure 1. Grouping structure, metrical structure, time-span tree, and prolongational tree.

The previous database and analysis editor for time-span trees was limited, that is, the length of each piece of music in the previous database was restricted to around eight bars because it was difficult for the analysis editor to manipulate a piece with a large number of bars and notes. The shortness of each piece in the GTTM database causes problems. For example, learning-based analyzers, such as the σGTTM [4] or deepGTTM [5], only learn with the data from eight bars of music, and the analysis performance for pieces with more than eight bars suddenly worsens. As another example, we could not verify the melody morphing method with a musical piece of more than eight bars.

To solve the problems with the previous editor, we developed a manual time-span tree generation tool that supports a piece with a large number of bars and notes. The experimental results show that the average editing time with the time-span tree generation tool is shorter than that with the previous tool. By using the generation tool, four musicologists created 50 time-span trees for pieces of music with 32 bars over the course of half a year.

This paper is organized as follows: Section 2 describes related work, Section 3 discusses the problems of the previous time-span tree editor, Sections 4 and 5 describe an generation tool for editing time-span trees and the GTTM database. Section 6 describes the experimental results and Section 7 concludes with a summary and an overview of future work.

2. RELATED WORK

Databases for music research have been created for various purposes [10–16]. The Real World Computing (RWC) Mu-sic Database is a copyright-cleared music database that contains the audio signals and corresponding standard MIDI files for 315 musical pieces [10, 11]. The Repertoire International des Sources Musicales (RISM), an international non-profit organization with the aim of comprehensively documenting extant musical sources from around the world, provides an online catalogue containing over 850,000 records, mostly for music manuscripts [12, 13]. The Variations 3 project provides online access to streaming audio and scanned score images for the music community with a flexible access control framework [14, 15]. The Essen folk song collection is a database for folk-music research that contains score data on 20,000 songs, along with phrase segmentation information, and also provides software for processing the data [16]. However, these research music databases [10–16] did not include deep musical structures like time-span trees.

We previously proposed GroupingXML, MetricalXML, TimespanXML, and ProlongationalXML for expressing deep musical structures, and we developed a manual editor for generating the structures [2, 9, 17] (Fig. 2). However, the manual editor for time-span trees had the following problems:

- It was difficult for the editor to use a piece of music of more than eight bars.
- Inputting the data of the structures needed a lot of time in proportion to the exponent of the number of notes.
- The editor could not handle cadences.

Therefore, we have developed a tool for manually generating time-span trees, which makes it possible to deal with longer pieces of music and input structures in a shorter amount of time, even for pieces of music with a large number of notes. In addition, we have defined and indicated the position of cadences in the analysis data of the TimespanXML.

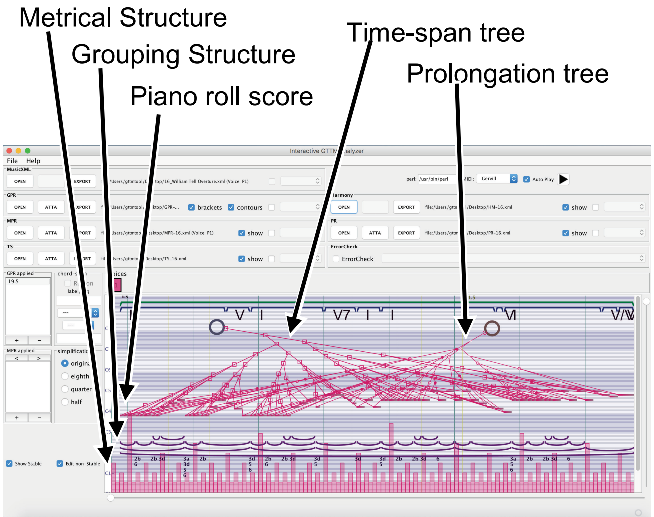

Figure 2. Previous editor with piano roll score.

3. PROBLEMS OF PREVIOUS TIME-SPAN TREE MANUAL EDITOR

Analysis of a time-span tree requires a lot of time, and, although the analysis time was shortened with the previous analysis editor, there were still the following problems:

- The score display was a piano roll. Figure 2 shows the previous analysis editor for manual editing, which was difficult for musicologists to understand intuitively because the score display was a piano roll. To solve this problem, in our new manual generation tool, described in Section 4.1, the score display was changed to a staff score so that chord progression was easy for musicologists to grasp.
- It was difficult to manipulate a piece of music with a large number of bars and notes. When the number of bars in a piece increased, the width of the notes on the piano roll became narrower, which made it difficult to understand each note's value (Fig. 3 a). When the number of notes in a piece increased, the number of branches in the time-span tree increased. Therefore, it was difficult for us to visually distinguish between an essential stem and an orna-mented branch in the time-span trees because stems and branches from the stems were mostly straight lines (Fig. 3 b).
  To solve these problems, as shown in Section 4.2, we implemented the generation tool on a tablet device, which en-abled us to zoom in and out to visually understand the notes' values and the numbers of stems and branches.

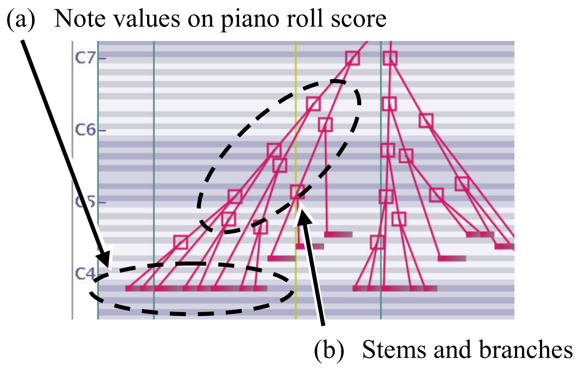

Figure 3. Difficulty of manipulating musical piece with large number of bars and notes.

- Top-down way of editing: In the first stage of time-span tree analysis, musicologists make a local tree from the bottom up by using the results of the grouping and metrical structure analyses. In our previous editor for manual editing, all linked time-span trees were given in advance and edited so that they became the target tree structure. Figure 4 shows the previous editor; the position where the selected branch can be connected is highlighted in the time-span tree. Thus, if we moved the branch to the target position, the branch would become highlighted and then connect to the desired position.

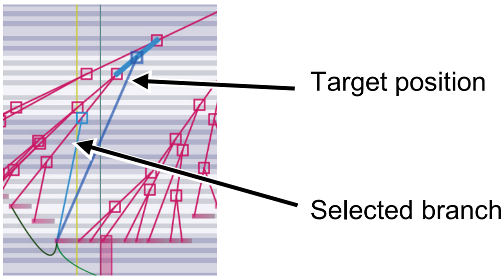

Figure 4. Highlighting the position that can connect the selecting branch in the previous editor for editing time-span tree.

The biggest problem with this method was that it was not possible to do a single action regarding the simplest and most frequently performed operations, such as swapping the parent and child of a branch. For example, when we changed a time-span tree, as shown in Figs. 5 a–c, the structure could not be directly changed because the time-span tree could be made once Fig. 5 b. The method for editing the time-span tree from the bottom up is described in Section 4.3.

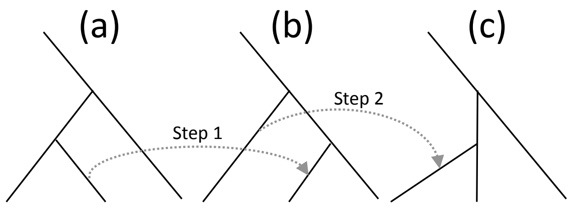

Figure 5. Manipulating time-span tree using previous editor.

4. TIME-SPAN TREE GENERATION TOOL

This section describes the four kinds of tool feature for manually generating time-span trees. The tool is data-compatible with the GTTM editor when using the MusicXML and the TimespanXML.

4.1 Display staff notation on tablet

The tool for manually generating time-span trees loads the MusicXML and displays it in staves (Fig. 6). By zooming the tablet screen in and out, you can check both the entire piece of music and individual sections. This makes it easy to operate, even when the entire piece is long or when there are many notes.

4.2 Handles for tree operation

Since it is difficult to touch and select single notes or branches with one's fingertip, we added circular handles to them. Handles were attached to all notes except for rests and grace notes. After loading the MusicXML , the handles are seen directly above the notes (Fig. 6 a). The heights of the handles are automatically separated because if two handles are very close to each other, it may not be possible to select the desired handle.

4.3 Tree generation in bottom-up manner

By simply placing a handle on another handle, a tree structure can be generated in a way that the original handle becomes a sub branch and the handle that was placed on top becomes the stem (main branch) (Fig. 6 b). After combining a stem and a branch, it is possible to move only the stem's handle. That is, the handle of the branch cannot be independently moved, and it will move following the movement of the stem. By clicking on and holding down the handle of the combined sub branch for a long time, a menu comes up that gives the option of separating them again.

4.4 Cadential retention

In the time-span analysis of the GTTM, the positions of the cadences in the time-span trees are indicated with an egg shape. However, the manual editor did not have such a display function. The cadence was retained from the third beat of the eighth bar in Fig. 6 c to the end of the piece.

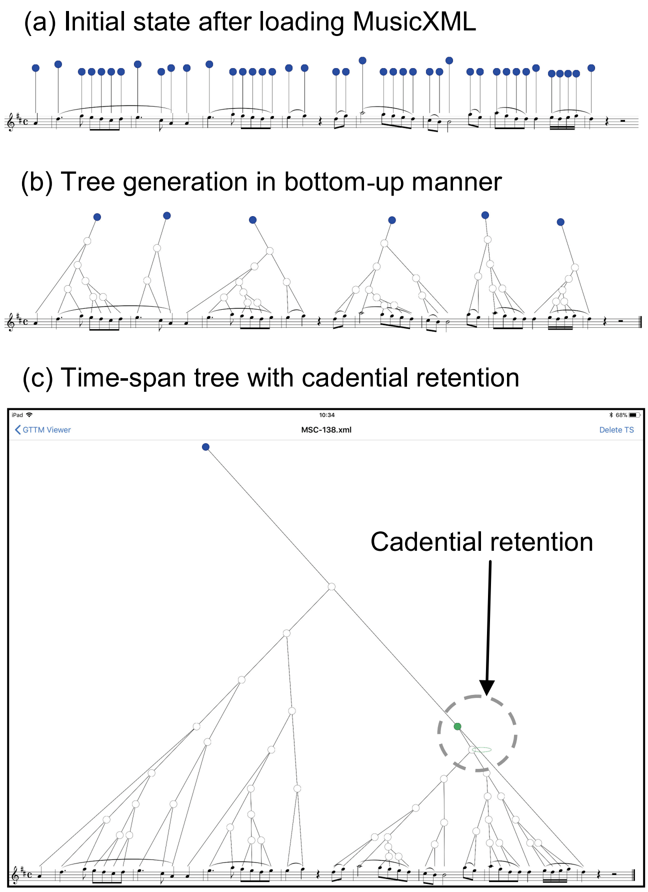

Figure 6. Screenshot of manual time-span tree generation tool.

5. GTTM DATABASE

Although we attempted to analyze polyphonic music, we restricted the target analysis data to monophonic music in the GTTM database because several rules in the theory allow only monophony.

5.1 XML-based data structure

We used an XML format for all analysis data. MusicXML [18] was chosen as the primary input format because it pro-vides a common 'interlingua' for music notation, analysis, retrieval, and other applications. We designed GroupingXML, MetricalXML, and TimespanXML to express each structure. We also designed HarmonicXML to express chord progres-sions. The XML format is suitable for expressing the hierarchical grouping structures, metrical structures, and time-span trees.

5.2 Score data

We collected 50 32-bar-long monophonic pieces of classical music that include notes, rests, slurs, accents, and articula-tions entered manually with music notation software called Finale [19]. We exported the MusicXML by using a plugin called Dolet. The 50 whole pieces and the 32 bars were selected by a musicologist (see appendix).

5.3 Analysis data

We asked four musicologists to manually analyze the score data by faithfully using the GTTM. The time-span tree editor was used for analyzing grouping and metrical structures, and the time-span tree generating tool was used for analyzing time-span trees. The musicologists also analyzed the chord progressions. Three other experts then crosschecked the manually produced results.

6. EXPERIMENTAL RESULTS

We compared the time taken to input information into a time-span tree with the developed generation tool and the previous editor. We asked three subjects who were music beginners that had never used the time-span tree editor or the generation tool before to create time-span trees with the programs over the course of one hour each. The subjects input the desired information while looking at a handwritten time-span tree on a printed staff score.

We prepared five pieces of music which were presented in a random order. There were a total of 10 trials with the previous editor and the new generating tool for creating time-span trees for these pieces, and the time for each trial was recorded. Between the trials, the subjects took a five-minute rest.

6.1 Construction of time-span trees from the starting state

When the time-span tree generation tool first starts up, no branches are connected (Fig. 6 a). On the other hand, in the beginning state of the time-span tree editor, each note is connected to a previous note as a branch (Fig. 7 a).

The trial results showed that the time for making time-span trees was shorter when using the generation tool for all pieces of music (Tab. 1). This is because the time-span tree editor requires a long time to swap parent and child branches.

6.2 Editing the output of the time-span tree analyzer

The start-up state was different between the editor and generation tool (see Section 5.1). In this section, we compare the time for editing time-span trees between the editor and the generation tool.

There is a time-span tree analyzer in the automatic time-span tree analyzer (ATTA) [2, 3] that has 13 adjustable parameters. The performance of the time-span tree analyzers is different depending on the adjustable parameters. It took us an average of about 10 minutes per musical piece to find plausible tuning for the set of parameters. Table 2 shows the F-measures of the time-span tree analyzer for each piece. The F-measure is given by the weighted harmonic mean of Precision P and Recall R. We used the tree structure output by the ATTA as a starting state in this section.

The results show that the average editing time with the generation tool is shorter than that with the editor. However, the editor can edit faster than the generation tool in pieces with high F-measures, such as Moments Musicaux. This is because, in a piece with a high F-measure, there are few times in which switching between the stems and branches is necessary and, as a result, it is possible to quickly edit with the editor. Conversely, the generation tool requires a long time for switching between them because we need to join them again after disconnecting a branch with all the operations of the time-span tree.

	Previous editor [s]	Developed generation tool [s]
1. Moments Musicaux	589	361
2. Wiegenlied	737	390
3. Traumerei	906	513
4. An die Freude	985	588
5. Barcarolle	923	521
Average (five pieces)	757	474

Table 1. Construction of time-span tree from the starting state.

	Previous editor [s]	Developed generation tool [s]	F-measure
1. Moments Musicaux	124	261	0.84
2. Wiegenlied	524	567	0.69
3. Traumerei	758	524	0.63
4. An die Freude	645	456	0.48
5. Barcarolle	345	248	0.60
Average (five pieces)	479	411	

Table 2. Editing the output of the time-span tree analyzer.

7. CONCLUSION

We have described the manual time-span tree generation tool and the GTTM Database. The main contributions of the study in this paper are as follows:

- **Development of a staff-based tool.**
  There have so far been only piano roll-based manual editing tools, musicologists' analyzed music written by hand onto the staves of printed paper because they required the staff for analysis. By developing a tool for manually generating time-span trees, trial and error became possible with the tool, which eliminated the need to print out sheet music.
- **Time-span tree generation in bottom-up manner.**
  Since the time-span tree generation tool generates trees from the bottom up, we have been able to significantly shorten the editing time compared with conventional editing tools. Furthermore, with the zoom function, one can easily grasp the entire piece or individual sections of long pieces of music.
- **Development of the GTTM database with pieces around 32 bars.**
  Since the data analyzed so far was for eight-bar extracts of music, it was a big restriction when making a learning-based analysis system or creating a system using time-span trees. Because the length was increased to 32 bars, it is expected that the performance of the analysis system will improve and that the applied system for time-span trees will expand.

Since we hope to contribute to the research of music analysis, we will publicize our generation tool a dataset of a fifty pairs of a score and musicologists' analysis results on our website:

`http://gttm.jp/`

We plan to analyze pieces other than classical pieces, such as jazz, in the future.

Acknowledgments

This work was supported by JSPS KAKENHI, Grant Numbers 17H01847 and 16H01744.

8. REFERENCES

[1] F. Lerdahl and R. Jackendoff, A Generative Theory of Tonal Music, MIT Press, 1985.

[2] M. Hamanaka, K. Hirata, and S. Tojo, Implementing "A generative theory of tonal music," Journal of New Music Research, pp. 249–277, 2006.

[3] M. Hamanaka, K. Hirata, and S. Tojo, "ATTA: Automatic Time-Span Tree Analyzer Based on Extended GTTM," Proc. of the 6th International Conference on Music Information Retrieval Conference (ISMIR 2005), pp. 358–365, 2005.

[4] M. Hamanaka, K. Hirata, and S. Tojo, "σGTTM III: Learning-Based Time-Span Tree Generator Based on PCFG," Proc. of International Symposium on Computer Music Multidisciplinary Research (CMMR2015), pp. 387–404, 2015.

[5] M. Hamanaka, K. Hirata, and S. Tojo, "deepGTTM-I: Local Boundaries Analyzer Based on Deep Learning Technique," Proc. of International Symposium on Computer Music Multidisciplinary Research (CMMR2016), pp. 8–20, 2016.

[6] M. Hamanaka, K. Hirata, and S. Tojo, "Melody Morphing Method Based on GTTM," Proc. of International Computer Music Conference (ICMC2008), pp. 155–158, 2008.

[7] M. Hamanaka, K. Hirata, and S. Tojo, "Melody Expectation Method Based on GTTM and TPS," Proc. of the 9th International Conference on Music Information Retrieval Conference (ISMIR2008), pp. 107–112, 2008.

[8] K. Hirata and S. Matsuda, "Interactive music summarization based on generative theory of tonal music," Journal of New Music Research, pp. 165–177, 2003.

[9] M. Hamanaka, K. Hirata, and S. Tojo, "Musical structural analysis database based on GTTM," Proc. of the 15th International Conference on Music Information Retrieval Conference (ISMIR2014), pp. 107–112, 2014.

[10] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical, and Jazz Music Databases," Proc. of the 3rd International Conference on Music Information Retrieval (ISMIR 2002), pp. 287–288, 2002.

[11] K. G. Saur Verlag, "RISM: International inventory of musical sources," In Series A/II Music manuscripts after 1600, 1997.

[12] M. Falletta, R. Hüsken, and K. Keil, "RISM: academic and technical challenges of musicological source research in an international framework," Studien und Materialien zur Musikwissenschaft, Hildesheim, 2010.

[13] J. Riley, C. Hunter, C. Colvard, and A. Berry, "Definition of a FRBR-based Metadata Model for the Indiana University Variations3 Project," http://www.dlib.indiana.edu/projects/variations3/docs/v3FRBRreportPhase2.pdf, 2008.

[14] J. Riley, "Moving from a Locally-Developed Data Model to a Standard Conceptual Model," Proc. of the 10th International ISKO Conference, pp. 124–130, 2008.

[15] H. Schaffrath, "The Essen associative code: A code for folksong analysis," In E. Selfridge-Field (Ed.), Beyond MIDI: The Handbook of Musical Codes, Chapter 24, pp. 343–361. MIT Press, Cambridge, 1997.

[16] M. Hamanaka, K. Hirata, and S. Tojo, "Interactive GTTM Analyzer," Proc. of the 10th International Conference on Music Information Retrieval Conference (ISMIR2009), pp. 291–296, 2009.

[17] MakeMusic Inc., "Finale," 2018, http://www.finalemusic.com/.

[18] MakeMusic Inc., "musicXML," 2018, http://www.musicxml.com/.

9. APPENDIX

List of musical piecesname and composer and bars that were used for analysis.

	Name of Piece	Composer	Bar	Measures
1	Symphony No. 40 in G minor KV.550 1st mov.	Mozart	1–42	42
2	Piano Concerto No. 3 in D minor 1st mov.	Rachmaninoff	3–27	25
3	Cavalleria Rusticana "Intermezzo sinfonico"	Mascagni	1–35	35
4	Impromptus Op. 142, D. 935 No. 2 in A-flat major	Schubert	1–46	46
5	String Quartet No. 4 Op. 18–4 in C minor	Beethoven	1–25	25
6	6 Pieces Op. 118 No. 2 "Intermezzo"	Brahms	1–48	48
7	Myrtle of flower Op. 25 No. 1 "Dedication"	Schumann	2–40	39
8	Scenes from Childhood Op. 15 No. 1 "Of Foreign Lands and Peoples"	Schumann	1–22	22
9	Cello Sonata Op. 38 No. 1 1st mov.	Brahms	1–33	33
10	4 Impromptus Op. 90 D. 899 No. 2 in E-flat major	Schubert	1–52	52
11	Violin Sonata 1st mov.	Franck	2–37	36
12	The carnival of the animals No. 13 "The Swan"	Saint Saens	2–27	26
13	Slavonic Dances 2 Op. 72 No. 2	Dvořák	1–32	32
14	Symphony No. 9 Op. 95 4th mov.	Dvořák	10–43	34
15	L’arlesienne 1st Suite No. 4 "Carillon"	Bizet	5–48	44
16	Piano Trio No. 7 Op. 97 "Archduke" 3rd mov.	Beethoven	1–28	28
17	Symphony No. 9 Op. 95 2nd mov.	Dvořák	7–42	36
18	Serenade Lyrique	Elgar	38–69	32
19	String Quartet Op. 18 No. 4 4th mov.	Beethoven	1–34	34
20	String Quartet Op. 64 No. 5 "Lark" 2nd mov.	Haydn	1–34	34
21	Piano Sonata No. 16 in C major K. 545 1st mov.	Mozart	1–28	28
22	Album for the Young Op. 68 No. 10 "Happy Farmer"	Schumann	1–20	20
23	The Blue Danube Op. 314	J. Strauss II	1–32	32
24	The Nutcracker Suite Op. 71a No. 3 "Waltz of the Flowers"	Tchaikovsky	38–69	32
25	9 Preludes Op. 1 No. 3	Szymanowski	1–25	25
26	Myrtle of flower Op. 25 No. 7 "The Lotus-Flower"	Schumann	2–27	26
27	Piano Quintet Op. 44 1st mov.	Schumann	1–26	26
28	Four Impromptus D. 935 No. 3 "Ballade"	Brahms	1–32	32
29	Piano Sonata No. 2 Op. 35	Chopin	9–40	32
30	Swan Lake Op. 20a "Dance of the little swans"	Tchaikovsky	1–32	32
31	The Seasons Op. 37a No. 10 "October-Autumn Song"	Tchaikovsky	1–49	49
32	The Seasons Op. 37a No. 11 "November-Troika"	Tchaikovsky	1–27	27
33	Don Quixote "Grand pas de deux"	Minkus	1–48	48
34	The Corsair "Variation"	Adam	5–52	48
35	O holy night	Adam	2–27	26
36	2 Melodies Op. 4 "Lydia"	Faure	3–37	35
37	Scenes from Childhood Op. 15 No. 2 "A strange story"	Schumann	1–20	20
38	My old Kentucky home, good-night	Foster	1–24	24
39	Old Folks at Home	Foster	1–24	24
40	Piano Sonata No. 16 in C major K. 545 2nd mov.	Mozart	1–32	32
41	Piano Sonata No. 16 in C major K. 545 3rd mov.	Mozart	1–28	28
42	Piano Sonata No. 12 in F major K. 332 1st mov.	Mozart	1–40	40
43	String Quartet Op. 3 No. 5 in F major 2nd mov.	Haydn	1–33	33
44	Divertimento Hob. II–46 in B-flat major 2nd mov.	Haydn	1–29	29
45	String Quartet No. 1 Op. 11 in D major 2nd mov.	Tchaikovsky	1–49	49
46	La Gioconda "Dance of Hours"	Ponchielli	1–16	16
47	6 Ecossaises No. 1 WoO. 83	Beethoven	1–32	32
48	25 Studies Op. 100 "Barcarolle"	Burgmüller	13–47	35
49	Serenade No. 13 in G major K. 525 "Eine kleine Nachtmusik" 2nd mov.	Mozart	1–39	39
50	Serenade for Strings in C major Op. 48 2nd mov. "Waltz"	Tschaikovsky	1–30	42