Automatic Phrasing System for Expressive Performance Based on The Generative Theory of Tonal Music

Madoka Goto[1], Masahiko Sakai[2], and Satoshi Tojo[3]*

[1] Hitachi, Ltd. madoka@trs.css.i.nagoya-u.ac.jp
[2] Nagoya University sakai@i.nagoya-u.ac.jp
[3] Asia University tojo_satoshi@asia-u.ac.jp

Abstract. Music phrase is an ambiguous notion since it often depends on the performer's subjective view. Thus far, we have employed Director Musices (DM) for automatic expressive performance, however, segmentation of phrases has only been given manually. In order to identify phrases from an objective viewpoint, we propose to obtain them from the trees acquired by the Generative Theory of Tonal Music (GTTM). We select the usable subtrees and regard the scope of the subtrees as phrases. We introduce a test tool to generate an expressive performance, given original music data to DM together with GTTM trees, to facilitate the phrasing steps.

Keywords: Automatic Expressive Performance, Generative Theory of Tonal Music, Director Musices

1 Introduction

Automatic expressive performance is an attractive challenge in music information processing, and competition such as RENCON [7] has been held for us to obtain more natural, smooth, and comfortable performance by computers [9]. The key issues in expressive performance concern *dynamique* (loudness of each tone) and *tempo* (speed).

Director Musices™ (DM), one of the distinguished generators of expressive performance, also gives variation in dynamique and tempo upon a phrase, with a specific rule called *phrase arch*. However, a phrase is not given in DM but needs to be given by human hands. Here, since a phrase is a subjective notion dependent on each performer, such a phrase arch also needs to be given by experienced human hands, and thus DM is not user-friendly, especially for those musically untrained users.

We consider giving phrase information automatically, independent of such subjective views, to DM. In this research, we propose to acquire phrases from the Generative Theory of Tonal Music (GTTM) [10]. From the theory, we can acquire syntactic tree

* This research was supported by JSPS Kaken 20H04302 and 21H03572. We thank Gilles Baroin for discussions on effects of composed phrases.

structures on a given music score, regarding each note as a linguistic morpheme, as is explained in the later section. Since such trees may include extraneous information for DM, we need to consider how we can retrieve usable phrases from them.

In this research, we have implemented a user-friendly interface to facilitate the automatic phrasing process, where we show how GTTM analyses are combined with music data, and report examples of expressive performance.

2 Phrase Arches in Director Musices

Director Musices (DM) [2] is a computer system that generates expressive performance based on given performance rules [1]. Input to DM is restricted either to MIDI or to its proper `mus` type file. DM, together with a rule palette of `pal` file in which performance rules are written, renders expressive articulation, and saves its MIDI.

Each performance rule accompanies a parameter called k-value, which specifies a grade of the intended effect of the rule upon music pieces. Among these rules, *phrase arch* that acts on a phrase, plays an important role, as it controls loudness (tone volume) and duration of each note. In the concrete, the beginning part of a phrase receives *accelerando* (gradually faster in tempo), and the ending part does *ritardand* (gradually slower); the loudness in accelerando grows larger while in ritardando smaller. This effect is illustrated in Fig. 1 though here the penult to the final note receives an accelerando.

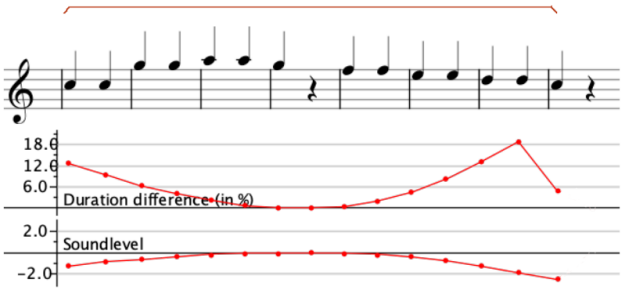

Fig. 1. DM phrasing application

For example, Fig. 2 is a screenshot of DM system. Here appear three layers of phrase arches above the score; since each of which can be given a different k-value, each layer is arranged in a different grade. While MIDI input is automatically converted to `mus` type, phrase arch itself must be edited manually; otherwise, the target music remains insipid and tasteless.

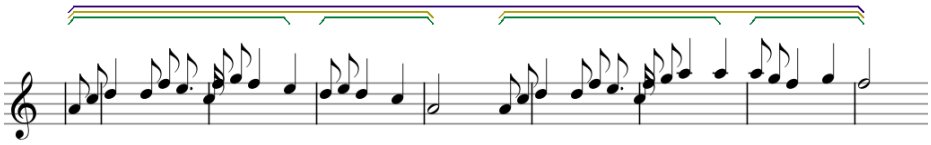

Fig. 2. Example of phrase arch: *Jupiter* of *The Planet* Op.32, Gustav Holst

3 Phrases obtained from GTTM

In order to avoid subjectivity in identifying phrases, we would like to rely on an external method to obtain them. In this section, we introduce a Generative Theory of Tonal Music (GTTM) and show how we can retrieve phrases from its analysis.

3.1 Generative Theory of Tonal Music

At the end of the 19th century, Heinrich Schenker proposed the *reduction principle*; that is, we can reduce the number of notes appearing on the score surface (*Vorgrund*), disregarding decorative notes, and can reach the fundamental structure (*Hintergrund*) or the basic melody line (*Urlinie*), consisting only of intrinsic notes to form cadences.

In order to embody the process from *Vorgrund* to *Hintergrund* in music, GTTM [10] invented a method to build a hierarchical tree in a bottom-up way, at each node of which two adjacent notes are compared and the more structurally important note goes upward, absorbing the less important one. Therefore, each of its nodes becomes either left-branching or right-branching. We call such importance among notes *salience*, according to [10]. Hereafter, we call the salient branch the *prime* branch, and the other the *secondary* branch.

GTTM consists of well-formedness rules that constrain rigid syntax, and other preference rules. In the process of building a tree, multiple preference rules may be applicable, and thus, the process necessarily becomes ambiguous. Hamanaka et al. [6] then assigned weighted parameters to all those applicable rules of GTTM, gave an algorithm to choose the most adequate rule in generating a tree, and realized a semi-deterministic procedure as a computer process.

GTTM consists of four sub-theories of grouping analysis, metrical analysis, time-span analysis, and prolongational analysis. The first three theories contribute to the construction of the time-span tree, and the prolongational analysis, together with the time-span tree, results in the prolongational tree. We summarize these trees as follows.

Time-span tree The grouping analysis finds boundaries in a sequence of musical notes, based on strength, duration, register, accent, and so on. Then, the metrical analysis identifies those notes with strong/weak beats in meters.
Here, we consider the note of group boundary (the beginning or the end of a group) with a stronger beat to be more salient than the neighboring note. Since the grouping structure is hierarchical, that is, smaller groups are merged into a larger group recursively, the comparison of salience also becomes hierarchical. Therefore, notes compose a knockout tournament in regard to structural salience.

We illustrate a time-span tree of *Jupiter* of Holst which we have employed in Fig. 2, in the left figure of Fig. 3.

Prolongational tree The time-span tree does not reflect the harmonic structure of the music piece. In order to represent the dependency of chords, and to organize cadences, we rearrange the branchings of the time-span tree to construct the prolongational tree.

Actually, the fundamental structure that Schenker originally intended was such cadences that are I (tonic) –V (dominant) – I (tonic), I – IV (subdominant) – V – I, I – IV – I, and so on. As a result, the left-hand side of the binary tree represents a progression of chords to cause *tension* while the right-hand side represents *relaxation*.

We show a prolongational tree of *Jupiter* in the right figure of Fig. 3.

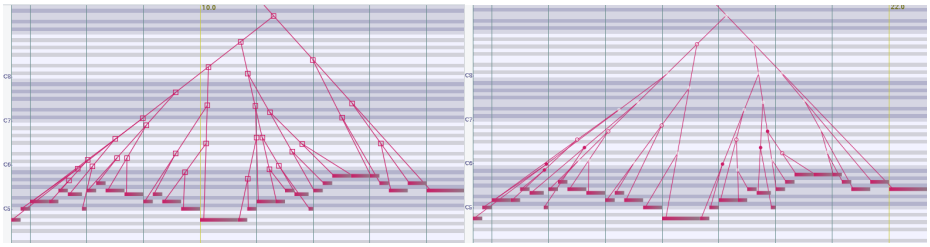

Fig. 3. Time-span (left) and Prolongational (right) trees of *Jupiter* [5]

3.2 Extraction of Phrases

From the two tree structures obtained from GTTM analyses, we can naturally consider that the scope of one subtree becomes a candidate of a phrase. In this research, we assign a phrase to each of hierarchical subtrees, as is shown in Fig. 4.

According to this, phrases become hierarchical; two adjacent phrases compose a larger phrase recursively in a higher hierarchy. Here, we can also naturally abandon smaller phrases, that are near to leaves (notes) in a tree, for the following two reasons.

- Even though we give expressive phrasing in a short phrase, the human auditory sense cannot catch it.
- Useless multiple layers of phrases blur each effect of expressiveness; overlapping of expressive effects may cancel each other, or may unnecessarily be augmented.

In order to avoid the above issues, we exclude those deep nodes counting from the top (root) node. Note that the top node represents the whole piece, and thus, the whole piece itself could be a phrase; however, in this research we do not regard the whole piece as a target to give expressiveness, since we pay attention to local phrasings.

Now, let $root(t)$ be a root node of tree t. Since $root(t)$ can possess two immediate branches, one of the two is more salient than the other; we name the prime (more salient) one $prm(v)$ and the second one $snd(v)$. Also, we provide the following notions.

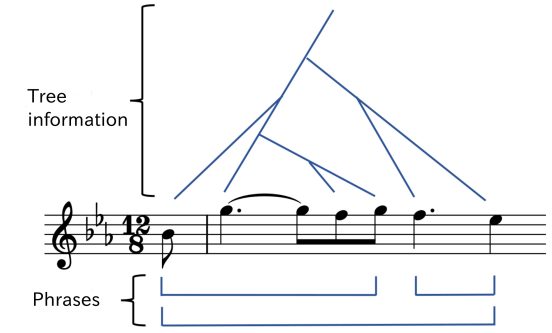

Fig. 4. Subtree as a phrase

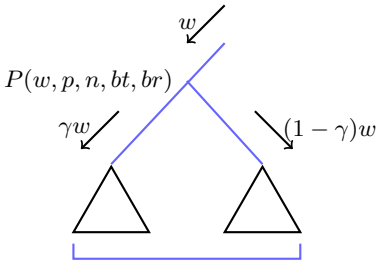

Fig. 5. Weight distribution

– $\#\mathrm{Nt}(v)$: the number of notes below v (except for slurred notes).
– $\#\mathrm{Bt}(v)$: the number of beats in terms of meters below v.
– $\#\mathrm{Br}(v)$: the number of bars below v.

We define a recursive function $phGen(v, w, p)$ for tree node v, weight w, and phrase-level p; when predicate

$$P(w, p, \#\mathrm{Nt}(v), \#\mathrm{Bt}(v), \#\mathrm{Br}(v))$$

holds, a phrase is recognized and we assign the total weight of w which would be distributed to her sub-branches under v, with the ratio of γ: $1-\gamma$ $(0 \leq \gamma \leq 1)$, between $prm(v)$ and $snd(v)$, respectively. The validity of P is adjustable dependent on w so that we can restrict the number of layers of phrases.

The phrase detection algorithm is summarized as follows.

Input : a tree t, an initial weight w_0, a distribution ratio γ, and a predicate P.
Output : layer of phrases, produced by $phGen(root(t), w_0, 1)$.
Procedure $phGen(v, w, p)$:

1. Regard the scope of t as a phrase with level p if $P(w, p, \#\mathrm{Nt}(v), \#\mathrm{Bt}(v), \#\mathrm{Br}(v))$ holds.
2. Call both $phGen(prm(v), \gamma w, p+1)$ and $phGen(snd(v), (1-\gamma)w, p+1)$ recursively and return.

The final step is shown in Fig. 5.

3.3 Adjustable Parameters

In the algorithm in the previous section, phrase arches are constructed depending on a tree t, an initial weight w_0, a distribution rate γ, and a predicate P. Therefore, for a given tree we can adjust these parameters to obtain plausible results.

To fine-tune these parameters through machine learning, we require the appropriate phrase information in DM format though, unfortunately, it is currently not available. As a result, we have opted for a less sophisticated approach as our initial step. We observe

	w_0	γ	$P(w, p, n, bt, br)$
Alg.0	2^3	$1/2$	$w > 1$ and $bt \geq 2$
Alg.1	b_0	$2/3$	$w \geq 4$ and $n \geq 2$
Alg.2	$(b_0)^2/n_0$	$1/2$	$w \geq 3.75$ and $n \geq 2$
Alg.3	$(b_0)^2/n_0$	$3/5$	$w \geq 5.9$ and $n \geq 2$
Alg.4	unused	unused	$bt/n \geq 0.6,\ bt > 4,$ $n \geq 2$, and $p < 4 \vee n \leq 4$
Alg.5	b_0	$1/2$	$w \geq 0.5$, $n \geq 2$, and $br \leq 10/p$

$b_0 : \#\mathrm{Bt}(\mathrm{root}(t)),\ n_0 : \#\mathrm{Nt}(\mathrm{root}(t))$

Table 1. Proposed set of parameters

	Alg.0	Alg.1
Le Cygne	8	5
Salut d'amour	9	4
sum	17	9

Table 2. Preliminary Experiments

the behavior of the algorithm in multiple preliminary experiments with Alg.0 and Alg.1, and propose three different assignments of parameters Alg.2, Alg.3, Alg.4, and Alg.5 in Table 1.

Each preliminary experiment is based on the following consideration. First, Alg.0 simply restricts the number of layers to three; then, Alg.1 revises Alg.0 as follows: (i) a long piece needs more minute segmentation and needs to increase the number of layers, and (ii) the primary branch may need the larger number of layers than the secondary branch.

In order to compare the efficacy of Alg.1 with that of Alg.0, we have experimented on *Le Cygne* (The Swan) of Camille Saint-Saëns, and on *Salut d'amour* (Love's greetings) of Edward Elgar. Table 2 shows that Alg.1 is unpopular; it is said that its tempo shift sounds unnatural. This result seems to be caused by the distribution ratio of $\gamma = 2/3$, which is too unbalanced and may generate too different numbers of layers.

Revising Alg.0 and Alg.1, we propose Alg.2 and Alg.3, the policy of which is commonly the following.

- We revise the distribution ratio to be flatter, as $1/2 < \gamma < 2/3$.
- Those pieces with a smaller number of notes, as opposed to the length of the piece, require more expressiveness. We augment the number of layers if $\#\mathrm{Nt}/\#\mathrm{Bt}$ is smaller.

In the process of weight passing from upper layers to lower ones, when the number of notes is unbalanced, the number of layers may not be even. To avoid unnatural expressive performance, we further propose Alg.4 based on the following two policies.

- We take $\#\mathrm{Nt}$ and $\#\mathrm{Bt}$ into account when we decide if a phrase is producible.
- When $\#\mathrm{Nt}/\#\mathrm{Bt}$ is large, we should avoid minute expressive performance, and avoid also small phrases.

4 System Implementation

We have implemented the algorithm proposed in Section 3.2, and have publicized this system.

4.1 Environmental Notes

We have developed an environment [11] that eases testing phrase-creation strategies, where we can compare performances generated from different phrases/palettes for DM on-the-fly. Prior to that, we needed to extend the file converter `kern2dm` in Humdrum Toolkit [8], to translate a `kern` music file into a `mus` DM-specific music file without phrase information. Thus, we revised `kern2dm` to accept a tree structure in `xml` as well as `kern` file [3]. In addition, we offer the following facilities.

Data downloader accesses GTTM database, and patches their musicXML scores on information such as tempo, title, and composer name if necessary. It generates `pdf` scores by using MuseScore™.

Phrase identifier generates scores with phrase information for DM by applying the extended `kern2dm`.

Performance arranger executes DM to create performances in midi formats, and transforms them into `wav`/`mp3` formats.

Screen interface prepares `html` files to present them on the screen, to compare the performances. Fig. 6 is a part of the index list of detailed pages like Fig. 7 for pieces GTTM music database.[4]

The biggest feature of this system is that it recalculates only the parts affected by the changed files by adopting the *make* system for program development. As a result, the waiting time required for recalculation after changing parameters can be greatly reduced.

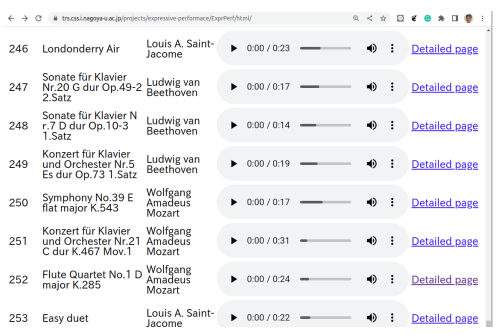

Fig. 6. A Screen Shot of Sample Pieces

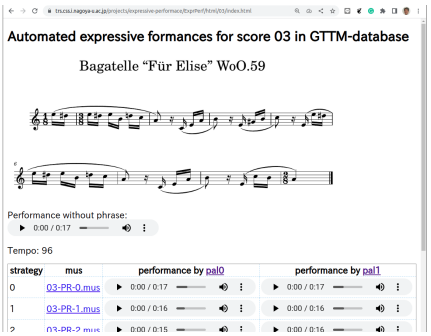

Fig. 7. A detailed page

4.2 Examples

We have conducted experimental analyses. We have applied Alg.2 and Alg.3 to the time-span and the prolongational trees for four pieces in [5], which showed conspicuous

[4] See sample page:
https://www.trs.css.i.nagoya-u.ac.jp/projects/expressive-performace/ExprPerf/html/

effects both in good and bad meanings; that are, Holst: *Jupiter* of *The Planet*, J. S. Bach: *Jesu, Joy of Man's Desiring*, Tchaikovsky: Waltz from *Swan Lake*, and Ravel: *Pavane pour une infante défunte*. We show all the phrases obtained by our method and rule palette employed in artificial expressive performance in Appendix A.

In comparison between the time-span tree and the prolongational tree, we found that there were no big differences. However, as to *Jesu, Joy of Man's Desiring* of J. S. Bach, this result could not be applied because the time-span tree of the piece is extremely deformed to be left-recursive branching. Since we cannot know if this is the adequate result of time-span analysis, we should doubt the reliability of the process of GTTM. In other words, if the original tree is not reliable, the resultant phrase structure also becomes unreliable.

Uncomfortable expressions are sometimes observed when a short phrase is located at the end of a long phrase. This situation is illustrated in Fig. 8, where each blue and green phrase affects the corresponding colored duration differences, and the red line denotes their additive effect. In this occasion, the conflict occurs between the deceleration due to the short phrase and the acceleration due to the long phrase in the first part of the short phrase.

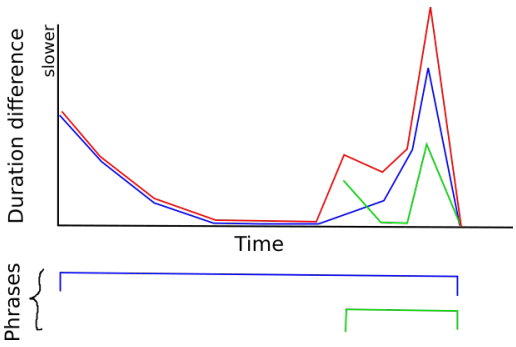

Fig. 8. An effect of composed phrases

In order to confirm the effects of these analyses, we have conducted a questionnaire of 17 examinees, including both of musically trained/ untrained listeners. In comparison between Alg.2 and Alg.3, Alg.2 had a good reputation in both trees (see Table 3); as for the time-span tree 19 vs 14 and for the prolongation tree 26 vs 9, and in sum 45 vs 23 that is 66% vs 34%. Even for each piece, Alg.2 is felt better than Alg.3. Thus, we can say the number of layers should be even.

Table 3. Questionnaire result (pr: prolongational tree and ts: time-span tree)

	Alg.2		pr		ts		pr+ts	
	pr	ts	Alg.2	Alg.3	Alg.2	Alg.3	Alg.2	Alg.3
Jupiter of *The Planet*	5	12	4	1	9	3	13	4
Jesu, Joy of Man's Desiring	15	2	10	5	1	1	11	6
Waltz of *Swan Lake*	6	11	6	0	5	6	11	6
Pavane pour une infante défunte	9	8	6	3	4	4	10	7
sum	35	33	26	9	19	14	45	23

5 Conclusion and future works

In this research, to avoid the arbitrary choice of phrases in expressive performance, we proposed to give phrases by subtrees obtained from GTTM analysis. We have expanded a file converter to include trees as input besides symbolic music data, implemented an environment for performance comparison, and have experimented to give expressiveness for selected pieces in GTTM database.

We have offered multiple parameters concerning the ratio between the number of notes and that of beats, the weight distribution between two branches at a tree node, and so on. As a result, effects upon time-span trees were found to be more natural than those upon prolongational trees, supposedly because of the balanced length of phrases.

In order to aim at better expressive performance, we need to consider the genre and age of target music when we adjust parameters. In general, baroque music is performed stably in tempo and only cadences should be played in *ritardand* as is provided in DM as FINAL RITARD [1], while in the romanticist age the tempo fluctuates rather freely dependent on performers. Thus, the phrase arch effect should be expressed more conspicuous in romanticist music.

As for the overlaid phrase arches, we need further improvement to avoid mutual cancellation/ augmentation of effects given by each phrase. In order to do this, we need to analyze the innate algorithms inside of DM and to revise them so as to include the interaction between phrase effects; this task remains a future work.

Machine learning is promising for accurately adjusting these parameters. For that purpose, creating a corpus consisting of musical scores with phrases extracted from actual performances by humans is necessary. Therefore, extracting phrases from actual performances is an essential issue for the future.

References

1. A. Friberg, R. Bresin, and J. Sundberg. Overviews of the kth rule system for musical performances. *Advances in Cognitive Psychology*, 2(2–3):145–161, 2006.
2. A. Friberg, V. Colombo, L. Fryden, and J. Sundberg. Generating musical performances with director musices. *Computer Music Journal*, 24(3):23–29, 2000.
3. M. Goto and M. Sakai. Extended kern2dm. https://git.trs.css.i.nagoya-u.ac.jp/transcription/humextra/-/tree/kern2dm.
4. M. Goto, M. Sakai, and S. Tojo. https://www.trs.css.i.nagoya-u.ac.jp/projects/expressive-performace/cmmr2023/.
5. M. Hamanaka. GTTM database. https://gttm.jp/gttm/ja/database/.
6. M. Hamanaka, K. Hirata, and S. Tojo. Implimenting a generative theory of tonal music. *Journal of New Music Research*, 35(4):249–277, 2006.
7. M. Hashida, T. M. Nakra, H. Katayose, T. Murao, K. Hirata, K. Suzuki, and T. Kitahara. Rencon: Performance rendering contest for automated music systems. In *Proceedings of the 10th International Conference on Music Perception and Cognition (ICMPC10)*, 2008.
8. D. Huron. Design principles in computer-based music representation. In *Computer Representations and Models in Music*, pages 5–59. Academic Press, 1992.
9. A. Kirke and E. R. Miranda. A survey of computer systems for expressive music performance. *ACM computer surveys*, 42(1):3:1–3:41, 2009.
10. F. Lerdahll and R. Jackendoff. *A Generative Theory of Tonal Music*. The MIT Press, 1983.

11. M. Sakai. Environment for automatic generation of expressive performances. https://git.trs.css.i.nagoya-u.ac.jp/transcription/dm-env.

A Example Analyses

We show phrases employed for phrase arches for four selected pieces, from Fig. 10 to Fig. 13, with a palette of Fig. 9. We also provide supplemental sound data, at [4].

```
(in-package "DM")
(set-dm-var 'all-rules '(
 (DURATION-CONTRAST 1.0
   :amp 1 :dur 1)
 (DOUBLE-DURATION 1.0 )
 (PHRASE-ARCH 1.4 :phlevel 1
   :turn 0.5 :last 0.2 :amp 2)
 (PHRASE-ARCH 1.4 :phlevel 2
   :turn 0.5 :last 0.2 :amp 5)
 (PHRASE-ARCH 1.4 :phlevel 3
   :turn 0.5 :last 0.2 :amp 3)
 (PHRASE-ARCH 1.4 :phlevel 4
   :turn 0.5 :last 0.2 :amp 2)
))
(set-dm-var 'sync-rule-list
 '((NO-SYNC NIL)
   (MELODIC-SYNC T)))
```

Fig. 9. Rule Palette of DM

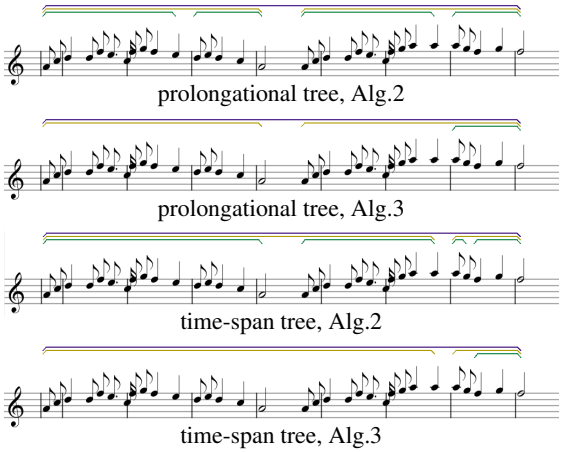

Fig. 10. *Jupiter* (GTTM DB No.49, #Bt = 24)

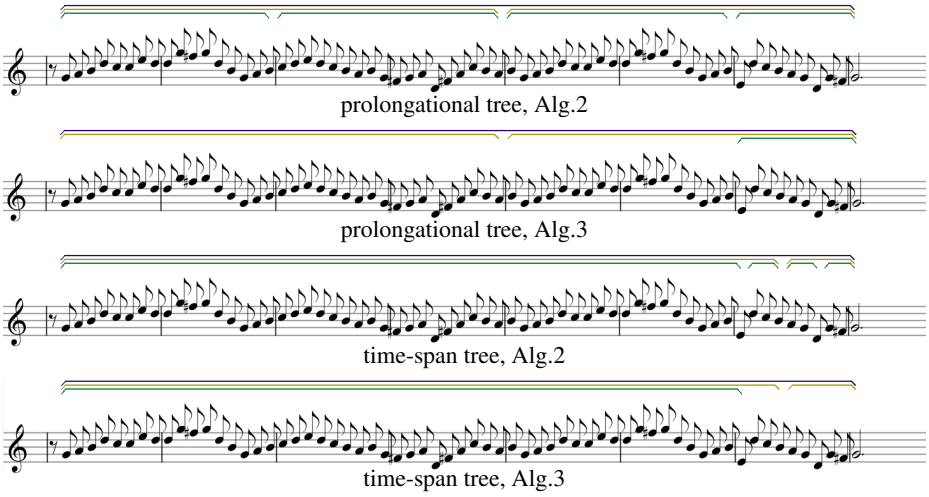

Fig. 11. *Jesu, Joy of Man's Desiring*, (GTTM DB No.70, #Bt = 36)

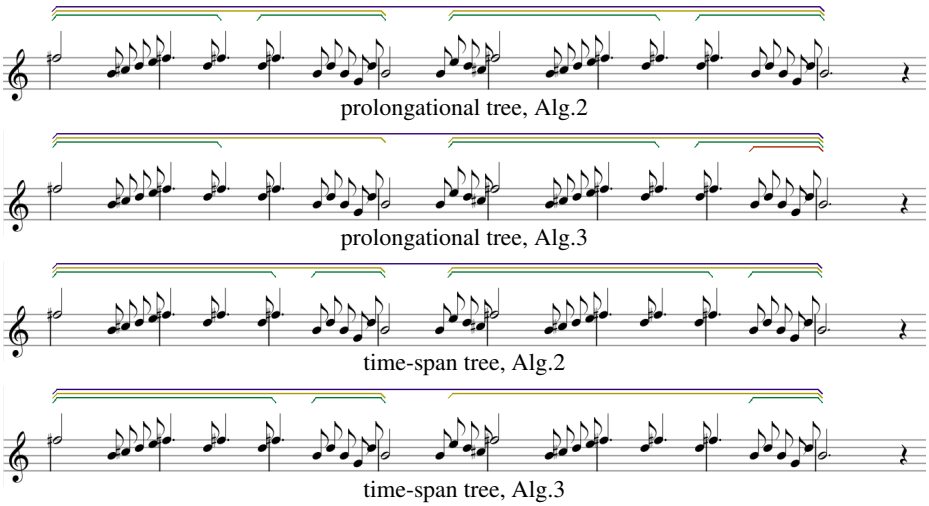

Fig. 12. Waltz of *Swan Lake*, (GTTM DB No.33, $\#\mathrm{Bt} = 32$)

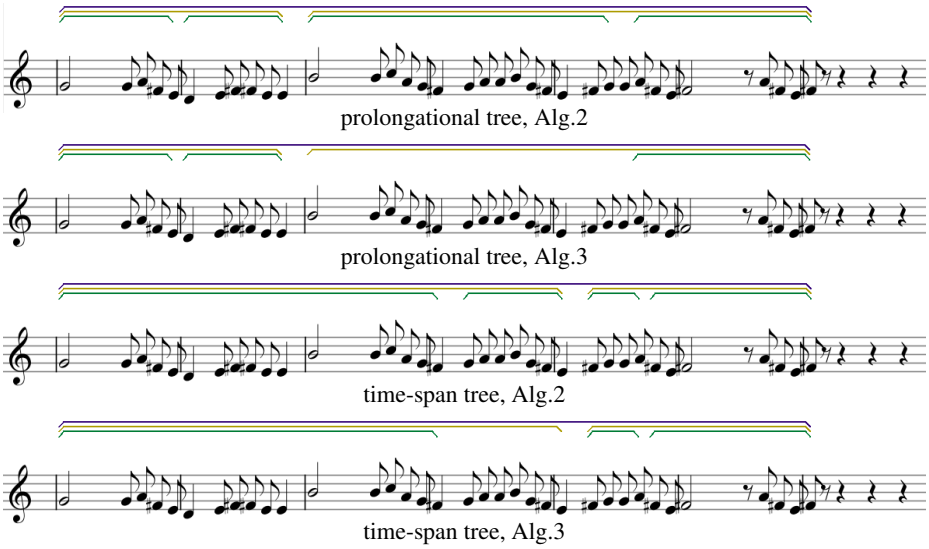

Fig. 13. *Pavane pour une infante défunte*, (GTTM DB No.73, $\#\mathrm{Bt} = 28$)