ENUMERATING LEFT HAND FORMS FOR GUITAR TABLATURES USING NON-DECREASING FINGER NUMBERS AND SEPARATORS

Gen Hori
Department of Data Science, Faculty of Business Administration
Asia University, Tokyo 180-8629, Japan
hori@asia-u.ac.jp

ABSTRACT

We introduce a guitar fingering decision method based on HMM and a tree diagram which we call "note-tablature-form tree" that can handle fingering decision of polyphonic pieces. To construct note-tablature-form tree for a given polyphonic chord, we need to (i) enumerate tablatures for a given chord, and (ii) enumerate left hand forms for each tablature. For the former, we introduce an enumeration method of tablatures for a given chord using a concept of permutations. For the latter, we introduce a new idea for exhaustive enumeration of left hand forms for a given tablature based on non-decreasing finger numbers and two kinds of separators that assign fingers to string-fret pairs.

1. INTRODUCTION

The process of determining optimal guitar fingerings for a given musical passage is a complex and subjective task that has long challenged guitarists of all levels. In recent years, researchers have explored the use of computational models to aid in this process. (See Sayegh[1], Radicioni et al.[2], Radisavljevic and Driessen[3], Tuohy and Potter[4], for example.) One such approach is the use of hidden Markov model (HMM), a statistical modeling technique that can capture the underlying patterns and structures in time series. As for applications of HMM to fingering decision, Hori et al.[5] applied input-output HMM to guitar fingering decision and arrangement, Nagata et al.[6] applied HMM to violin fingering decision, and Nakamura et al.[7] applied merged-output HMM to piano fingering decision. Hori and Sagayama[8] and Hori[9] proposed extensions of the Viterbi algorithm for fingering decision.

The purpose of the present study is to extend guitar fingering decision method based on HMM[5] and a tree diagram[10] for monophonic cases to one that can handle polyphonic cases. We cast guitar fingering decision as a decoding problem of HMM whose output symbols are musical notes and hidden states are left hand forms. In this case, we need to enumerate left hand forms for each chord in a given piece to perform fingering decision for polyphonic pieces. To do that, (i) we enumerate tablatures for a chord, and then (ii) enumerate left hand forms for each tablature.

For the former, we introduce an enumeration method of tablatures for a given chord using a concept of permutations in Section 3.1. For the latter, we introduce a new idea for enumeration of left hand forms for a given tablature using non-decreasing finger numbers and separators in Section 3.2, which provides a new insight for exhaustive search for all the possible left hand forms for a given tablature.

The rest of the paper is organized as follows. Section 2.1 reviews guitar fingering decision method based on HMM[5] and Section 2.2 note-tablature-form tree for enumeration of left hand forms using monophonic cases[10]. Section 3 extends note-tablature-form tree for polyphonic cases where Section 3.1 introduces an enumeration method of tablatures for a given chord and Section 3.2 our new idea for exhaustive enumeration of left hand forms for a given tablature. Section 4 concludes the paper.

2. NOTE-TABLATURE-FORM TREE FOR MONOPHONIC CASES

This section reviews the guitar fingering decision method based on HMM whose output symbols are musical notes and hidden states are left hand forms[5] and note-tablature-form tree for enumeration of player's left hand forms[10]. Although we limit our attention to a monophonic case to simplify the explanation in this section, the results extend to polyphonic cases. See [5] for HMM for polyphonic cases. Note-tablature-form tree is extended for polyphonic cases in the following section.

2.1 Fingering decision based on HMM

To play a single note on a guitar, a guitarist holds down a string-fret pair,

$$p_i = (s_i, f_i),$$

with a finger h_i of the left hand and picks the string s_i with the right hand, where $s_i = 1, \ldots, 6$ is a string number (from the highest to the lowest), $f_i = 0, 1, \ldots$ is a fret number where $f_i = 0$ means an open string, and $h_i = 1, 2, 3, 4$ is a finger number where 1,2,3 and 4 mean the index, middle, ring and pinky fingers, respectively. Therefore, a left hand form q_i for playing a single note can be expressed in a triplet q_i,

$$q_i = (s_i, f_i, h_i).$$

The MIDI note number of the note played by the form q_i is calculated as follows where o_{s_i} denotes the MIDI note number of the open string s_i,

$$n(q_i) = o_{s_i} + f_i.$$

We cast fingering decision as a decoding problem of HMM whose output symbols are musical notes and hidden states are left hand forms, where a fingering is obtained as a sequence of hidden states given a monophonic phrase as a sequence of output symbols.

The difficulty levels of the moves from forms to forms are implemented in the probabilities of the transitions from hidden states to hidden states; a small value of the transition probability means the corresponding move is difficult and a large value means easy. We assume that the four fingers of the left hand are always put on consecutive frets in this section for simplicity. This lets us calculate the *index finger position* (the fret number the index finger is put on) of form q_i as $g(q_i) = f_i - h_i + 1$. Using the index finger position, we set the transition probability from hidden state q_i to hidden state q_j as

$$a_{ij}(d_t) \propto \frac{1}{2d_t} \exp\left(-\frac{|g(q_i) - g(q_j)|}{d_t}\right) \times P_H(h_j) \quad (1)$$

where $\propto$ means proportional and the left hand side is normalized so that the summation with respect to j equals 1 for all i. The first term of the right hand side is taken from the probability density function of the Laplace distribution that concentrates on the center and its variance d_t is set to the time interval between the onsets of the $(t-1)$-th note and the t-th note. The second term $P_H(h_j)$ corresponds to the difficulty level of the destination form q_j defined by the finger number h_j.

As for the output probability, because all the hidden states have unique output symbols in our HMM for fingering decision, it is one if the given output symbol n_k is the one that the hidden state q_i outputs and zero if the given output symbol is not,

$$b_{ik} = \left\{ \begin{array}{ll} 1 & (n_k = n(q_i)) \\ 0 & (n_k \neq n(q_i)) \end{array} \right. . \quad (2)$$

2.2 Note-tablature-form tree

To perform fingering decision as a decoding problem of HMM described in the previous section, we need to enumerate left hand forms for each note in a given sequence of notes, which is done by drawing note-tablature-form tree

Figure 1. Note-tablature-form tree for guitar (left) and corresponding diagram for piano (right) illustrating difference between fingering decision of string instruments and other instruments

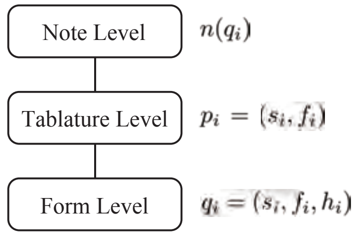

Figure 2. Three-level model for fingering decision of string instruments

that describes the difference between fingering decision of string instruments and other instruments as follows.

For example, on the piano, there is only one key on the keyboard to press for each note, and therefore fingering decision for a given sequence of notes is a matter of deciding which finger to press on the key for each note (Fig. 1, right). On the other hand, with the guitar, each note corresponds to several string-fret pairs that play it, and in addition, we have a matter of which finger to press for each string-fret pair (Fig. 1, left). In other words, fingering decision for the piano is simply a matter of finger assignments while fingering decision for the guitar consists of string assignments followed by finger assignments. This situation with the guitar is illustrated in a tree diagram (Fig. 1, left) which we call "note-tablature-form tree." While a note-tablature-form tree for a monophonic case in Fig. 1 is easy to draw, we tackle the problem of drawing corresponding trees for polyphonic cases in the following section.

The above-explained situation with fingering decision of string instruments is described by a three-level model for string instruments that consists of (1) note level, (2) tablature level, and (3) form level (Fig. 2). In relation to the notation introduced in Section 2.1, the note level contains the information of $n(q_i)$, the tablature level $p_i = (s_i, f_i)$, and the form level $q_i = (s_i, f_i, h_i)$, respectively. In guitar scores, the score and the tablature contains the information of the note level and the tablature level, respectively. From the viewpoint of fingering decision based on HMM, the hidden states correspond to the form level and the observed symbols to the note level.

3. NOTE-TABLATURE-FORM TREE FOR POLYPHONIC CASES

We discussed fingering decision method based on HMM and note-tablature-form tree limiting our attention to monophonic cases in the previous section for the sake of simplicity. However, in order to prove our fingering decision method practical, we need to consider polyphonic cases where multiple notes are played simultaneously. To extend our fingering decision method for polyphonic cases, it is enough to extend note-tablature-form tree to one for polyphonic cases which we construct in this section, and then we can perform fingering decision for polyphonic cases in the same manner for monophonic cases. (See [5] for HMM for polyphonic cases.) The construction of note-tablature-form tree for a given chord consists of searching for tablatures for a given chord followed by searching for left hand forms for each of the obtained tablatures.

3.1 From chord to tablature

Searching for tablatures for a given chord is relatively easy compared to searching for forms for a given tablature. It consists of assigning the chord notes to the strings followed by confirming that the notes are within the pitch ranges of the assigned strings. If we are given a chord consisting of n notes ($n \leq 6$), we have ${}_6P_n$ permutations of n out of 6 that give possible assignments of the n notes to the 6 strings. (If we can assume that the pitches of the notes in the chord is monotonic with respect to the string numbers, we can reduce the search to a number of combinations rather than permutations, but this is not the case because we consider left hand forms including open strings that can break the monotonicity of the pitches.) In Algorithm 1, $chord$ is a variable length array of MIDI note numbers while tab is a fixed length array of fret numbers, 0 or $None$ with length 6, the number of the strings, where 0 means an open string while $None$ means that the string is not played. The array $open$ keeps the MIDI note numbers of the open strings. Subtracting the MIDI note number of the open string from the MIDI note number of the note yields the fret number for playing the note on the string. If all the fret numbers obtained by such subtraction are between 0 and F, the number of the frets of the instrument, then the tablature is valid and is added to the list of tablatures $tabs$. Fig. 3 illustrates an example note-tablature tree for a simple C chord consisting of three notes, C, E and G, generated by Algorithm 1 for a standard guitar with $F = 21$ frets. Out of ${}_6P_3 = 120$ permutations, only 8 permutations give valid tablatures.

Algorithm 1

procedure CHORD2TABS($chord$)
  $tabs \leftarrow \emptyset$
  $n \leftarrow$ length of $chord$
  for $perm \leftarrow$ perms of n from $\{0,1,2,3,4,5\}$ **do**
    $tab \leftarrow [None, None, None, None, None, None]$
    for $i = 0$ to $n-1$ **do**
      $tab[perm[i]] \leftarrow chord[i] - open[perm[i]]$
    if $((f \geq 0 \wedge f \leq F) \vee f = None)$ for all $f \in tab$ **then**
      $tabs \leftarrow tabs \cup \{tab\}$
  return $tabs$

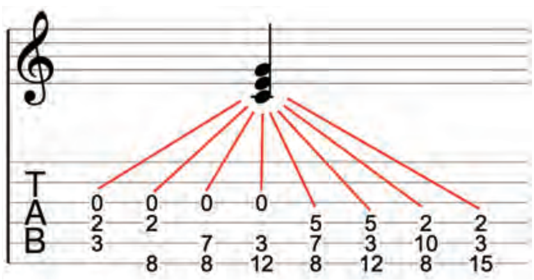

Figure 3. Note-tablature tree for C chord

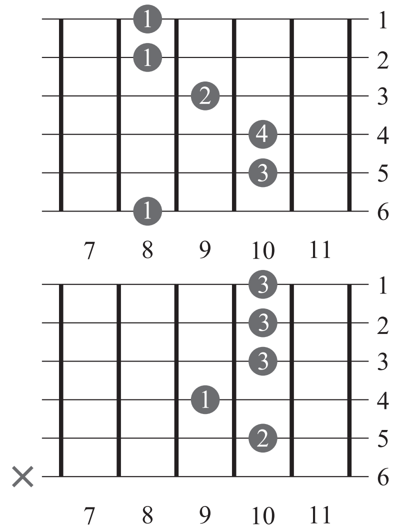

Figure 4. Left hand forms represented by finger numbers

3.2 From tablature to form

Searching for left hand forms for a given tablature is relatively difficult because the search space of left hand forms is huge and it is difficult to enumerate all the possible left hand forms in an orderly manner like possible tablatures as permutations. Two main strategies for searching for left hand forms for a given tablature can be considered here. One is to collect known forms from guitar chord books to build a database of left hand forms and then search for forms most suitable for a given tablature in the database. The other is to exhaustively search for all the possible left hand forms for a given tablature. They have their own pros and cons. The former is easy to implement and is guaranteed not to output strange forms but may miss out on some novel new left hand forms. The latter may not miss out on any forms but is difficult to implement because it is difficult to enumerate all the possible forms. We introduce a new idea for enumerating possible left hand forms to implement the latter approach in the following.

3.2.1 *Representing forms by finger numbers*

Fig. 4 shows two examples of standard guitar chords, C and G7(9), and their standard fingerings where the finger numbers 1, 2, 3 and 4 indicates the index, middle, ring and pinky fingers, respectively. The index finger (1) in the upper chord and the ring finger (3) in the lower chord hold down multiple strings. A method of playing in which one finger holds down multiple strings is called “barre” or “ceja.” In the way the chords and their fingerings are displayed in Fig. 4, left hand forms can be represented by assigning finger numbers to all the string-fret pairs to hold down.

3.2.2 *Numbering string-fret pairs*

We consider the numbering of the string-fret pairs on the fretboard as shown in Fig. 5 where a string-fret pair (f, s) is numbered by an integer $6f - s$ in hexadecimal where $f = 1, 2, \ldots$ is a fret number and $s = 1, 2, \ldots, 6$ is a string number (from the highest to the lowest). We show

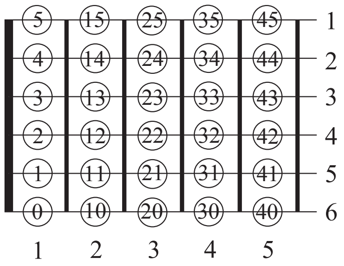

Figure 5. Numbering of string-fret pairs by integer $6f - s$ in hexadecimal

the numbering in hexadecimal notation in Fig. 5 just for ease of reading and it is not essential in the following discussion whether the numbering is expressed in decimal or hexadecimal.

3.2.3 Non-decreasing finger numbers

Then it holds with very few exceptions that the finger numbers representing a single left hand form are monotonically non-decreasing with respect to the numbering. This is because (i) a finger with a larger finger number is put on a higher or the same fret (and therefore a string-fret pair with a larger or equal digit in its 6's place), and (ii) a finger with a larger finger number is shorter (except that the middle finger (2) is longer that the index finger(1)) and is put on a string with a smaller string number (and therefore a string-fret pair with a larger digit in its first place). Fig. 6 illustrates that the finger numbers representing a single left hand form are monotonically non-decreasing with respect to the numbering using two cases of the chords in Fig. 4. The upper chord in Fig. 6 has six string-fret pairs to hold down with numberings 70, 74, 75, 83, 91 and 92 that are assigned with the finger numbers 1, 1, 1, 2, 3 and 4 in Fig. 4, which are monotonically non-decreasing with respect to the numberings. The lower chord in Fig. 6 has five string-fret pairs to hold down with numberings 82, 91, 93, 94 and 95 that are assigned with the finger numbers 1, 2, 3, 3 and 3 in Fig. 4, which are again monotonically non-decreasing with respect to the numberings.

3.2.4 Enumerating left hand forms

Having observed that the finger numbers are non-decreasing with respect to the numbering of the string-fret pairs, we can enumerate left hand forms for a given tablature without any leaks by inserting three separators separating four fingers into the sequence of the string-fret pairs representing the given tablature. Fig. 7 represents two left hand forms of Fig. 4 using string-fret pairs and separators, where long and short lines indicate mandatory and optional separators which are explained in the following sections. The separators are inserted between string-fret pairs where the finger numbers change. The index finger is assigned to the string-fret pairs to the left of the left separator, the middle finger to ones between the left and the middle separators, the ring finger to ones between the middle and the right separators, and the pinky finger to ones to the right of the right separator. If a separator is at the leftmost or the rightmost as in the right form of Fig. 7 or two separators are next to each other, it means that some fingers are not used in the form. When we have n string-fret pairs to hold down, there are a total of ${}_{n+3}C_3$ ways to insert three separators, but only a small fraction of these are actually available.

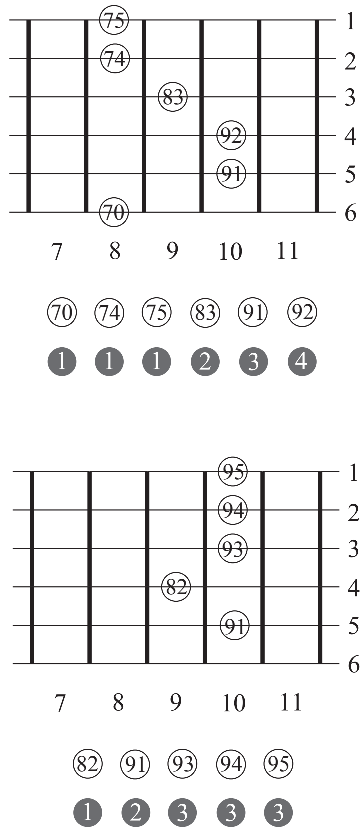

Figure 6. Non-decreasing finger numbers representing left hand forms

3.2.5 Inserting mandatory separators

When we insert three separators into a sequence of string-fret pairs, there are positions where separators must first be inserted. First, because it is impossible to hold down different frets with one finger, a separator must be inserted between two string-fret pairs with different fret numbers, that is, with numberings with different digits in their 6's places. Second, because it is impossible for one finger to hold down string-fret pairs on a single fret separated by a string played at a lower fret, such as 91 and 93 sepa-

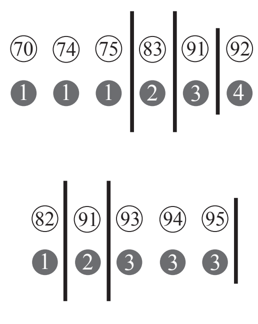

Figure 7. Left hand forms represented by string-fret pairs and separators

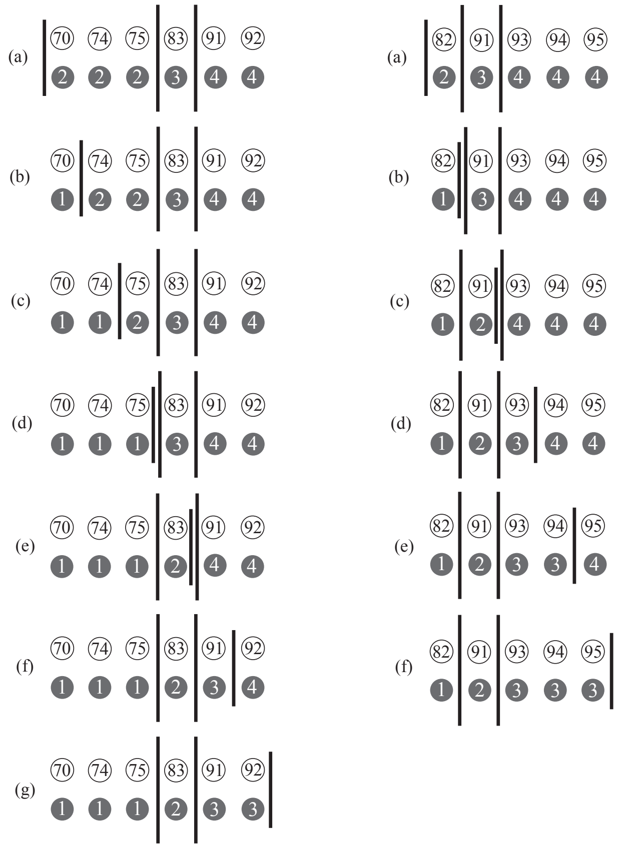

Figure 8. Exhaustive enumeration of left hand forms by inserting optional separators

rated by the fourth string played at 82 in the lower form of Fig. 7, a separator must be inserted between such two separate string-fret pairs. We call separators inserted to such positions "mandatory separators." In Fig. 7, long lines indicate mandatory separators while short lines indicate optional separators which are explained in the following section.

3.2.6 Inserting optional separators

When we have less than three mandatory separators, we insert optional separators until we have three separators in total. For example, both forms of Fig. 7 have two mandatory separators, which make us insert one more optional separator for each. The upper form of Fig. 7 has six string-fret pairs thus seven positions to insert an optional separator while the lower form has six positions to insert as shown in Fig. 8, where mandatory and optional separators are indicated by long and short lines. As the positions of the separators change, so do the finger numbers under the string-fret pairs and thus the left hand forms. We note that some forms of Fig. 8 are very difficult or impossible to play and not all of those forms are available. Here we have introduced a new idea for exhaustive enumeration of left hand forms for a given tablature and we still need to discuss how to eliminate the unplayable ones, which we leave to our future study.

4. CONCLUSION

We have reviewed a guitar fingering decision method based on HMM and note-tablature-form tree for monophonic cases and tried to extended the tree diagram to polyphonic cases. For that purpose, we have introduced an enumeration method of tablatures as permutations for a given chord. Furthermore, we have introduced a new idea for exhaustive enumeration of left hand forms for a given tablature based on non-decreasing finger numbers and two kinds of separators that assign fingers to string-fret pairs. We have noted that some left hand forms enumerated by our proposed method are very difficult or impossible to play and need to be eliminated. We leave elimination of such unplayable forms to our future study.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 21H03462.

5. REFERENCES

[1] S. I. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989.

[2] D. Radicioni, L. Anselma, and V. Lombardo, "A segmentation-based prototype to compute string instruments fingering," in *Proceedings of the Conference on Interdisciplinary Musicology (CIM04)*, vol. 17, Graz, Austria, 2004, pp. 97–104.

[3] A. Radisavljevic and P. F. Driessen, "Path difference learning for guitar fingering problem," in *Proceedings of International Computer Music Conference (ICMC2004)*, vol. 28, Miami, USA., 2004.

[4] D. R. Tuohy and W. D. Potter, "A genetic algorithm for the automatic generation of playable guitar tablature," in *Proceedings of International Computer Music Conference (ICMC2005)*, 2005, pp. 499–502.

[5] G. Hori, H. Kameoka, and S. Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *Journal of Information Processing*, vol. 21, no. 2, pp. 264–271, 2013.

[6] W. Nagata, S. Sako, and T. Kitamura, "Violin fingering estimation according to skill level based on hidden Markov model," in *Proceedings of International Computer Music Conference and Sound and Music Computing Conference (ICMC/SMC2014)*, Athens, Greece, 2014, pp. 1233—-1238.

[7] E. Nakamura, N. Ono, and S. Sagayama, "Merged-output HMM for piano fingering of both hands," in *Proceedings of International Society for Music Information Retrieval Conference (ISMIR2014)*, Taipei, Taiwan, 2014, pp. 531–536.

[8] G. Hori and S. Sagayama, "Minimax Viterbi algorithm for HMM-based guitar fingering decision," in *Proceedings of International Society for Music Information Retrieval (ISMIR2016)*, New York City, U.S.A., 2016, pp. 448–453.

[9] G. Hori, "Extension of decoding problem of HMM based on L^p-norm," in *Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2018)*, 2018, pp. 3989–3993.

[10] ——, "Three-level model for fingering decision of string instruments," in *Proceedings of the 15th International Symposium on CMMR*, 2021, pp. 93–98.