Evaluation of soprano donnée system for the theory of harmony

Masanobu Miura¹, Mitsuru Obana² and Masuzo Yanagida³

¹Graduate School of Engineering, Doshisha University,
1-3, Tatara-Miyakodani, Kyo-Tanabe, 610-0321 Japan

²Department of Visual Image Design, Takarazuka University of Arts and Design,
7-27, Hanayashiki-Tsutsujigaoka, Takarazuka, 665-0803, Japan

³Department of Knowledge Engineering and Computer Sciences, Doshisha University,
1-3, Tatara-Miyakodani, Kyo-Tanabe, 610-0321 Japan

(Received 25 April 2002, Accepted for publication 10 July 2002)

Keywords: Theory of harmony, Soprano donnée, Depth-first search, MIDI
PACS number: 43.58.Ta

1. Introduction

Theory of harmony is the main subject in music education for traditional music and even for most of contemporary music. It was established in 17–18 centuries and is now the basis of western music including not only classical but also popular music. In most music courses in universities, students are required to learn the theory of harmony at the beginning stage of their academic courses. The theory of harmony consists of various kinds of rules, and some of them are called inhibition rules as they limit the freedom of note assignment. So, students are required to understand individual rules together with inter-relations among them.

Educational training concerning the theory of harmony consists of two types of exercises: “basse donnée” and “soprano donnée.” Basse donnée is a task to assign upper three voices, i.e. soprano, alto and tenor, to a given bass sequence observing the inhibition rules, while soprano donnée is a task to assign lower three voices, i.e. alto, tenor and bass, to a given soprano sequence.

The authors have developed “Basse Donnée System (BDS)” [1,2] that yields all the allowable solutions for given bass sequences. Proposed here is a system that yields all the allowable solutions for given soprano sequences within triads, and is named “Soprano Donnée System (SDS),” forming a “Harmony System” together with BDS. It can judge if student’s answers violate any of the inhibition rules and it enables students to study the theory of harmony by themselves through combined use of them.

2. Soprano donnée tasks in the theory of harmony

Figure 1 shows an example of soprano donnée and an example of its allowable solutions. Students’ answers are regarded to be correct if the answer sequences do not violate any inhibition rules. There, however, can be so many allowable solutions for a given sequence of bass or soprano part. There is no exercise book that gives all the allowable solutions because thousands of correct answers are possible for a given sequence. This makes self-studying of the harmony theory so difficult. The rules are common for these two twin harmony tasks, basse donnée and soprano donnée. To express rules in the theory of harmony, the conventional rule-based scheme does not seem to be efficient for this case

Figure 1 consists of two musical staves. (a) A given soprano sequence: A single melodic line in G major, 4/4 time, consisting of 16 eighth notes. (b) An example of possible solutions for (a): The same soprano sequence is shown, but with three additional staves below it representing the alto, tenor, and bass parts. Above the staves, Roman numerals (I, VI, II, I⁶, V, VI, V, I, V⁶, I, V, VI, V, VI, V, I) and figured bass notation (T, S, T, D, T, D, T, D, T, D, T, D, T, D, T) are provided for each measure. Below the staves, Roman numerals (R3, R1, R1, R1, R1, R1, R1, R1) indicate the root position for each measure.

Fig. 1 An example of soprano donnée (a) and its solution (b).

as the total number of rules is not so large. So, the system is realized in an algorithmic implementation like BDS. The present paper describes the facilities and performances of SDS and then, quantitatively compares the difficulties of basse donnée tasks and soprano donnée tasks.

Soprano donnée differs from basse donnée in the point that the given sequence is the soprano part while the given sequence is the bass part for basse donnée. Though, the rules and inhibition rules are completely common for basse donnée and soprano donnée, there is a huge gap in difficulty degree between these two tasks. Figure 2 shows possible chord transitions within triads excluding III and VII (chords on the 3rd and 7th). Possible chord progressions are obtained as a sequence of chords allowed in this figure. In basse donnée tasks, once a chord is assigned to a bass note, the inversion type is forcedly determined without freedom as the lowest note is given as the bass note in the given bass sequence. In soprano donnée, however, the inversion type cannot be determined until the lowest note is fixed. There is, then, three possibilities of selecting one out of the three inversion types: root position, the first inversion and the second inversion for each note in a given soprano line. That makes the number of allowable solutions huge for soprano donnée compared with that of basse donnée. Furthermore, feasibility of assigning lower three voices for a given soprano note is not known until

Fig. 2 Allowable chord transitions within triads.

a number of checks referring to every inhibition rules is completed in local contexts of chord sequences. There are many cases that no lower three voice allocation is possible even if chord progression is superficially possible. This makes selection of chord progression difficult and time consuming.

3. Soprano donnée system

Proposed here is a system, named “Soprano Donnée System (SDS),” that generates all the possible sequences of lower three voices for given soprano sequences within triads. All the rules implemented in the current system are cited from a text book [3] widely used in Japan. The system is realized in an algorithmic scheme pursuing processing speed. An example of implementing the voice crossing inhibition rule in an algorithmic scheme is shown in Fig. 3. It is easy to implement inhibition rules in algorithmic forms in the beginning stage. The number of inhibition rules in introductory steps in the theory of harmony is about 40, so whole the system can be within a manageable size by algorithmic formulation. However it is difficult to add or modify rules to the existing set of rules. So, a new system based on a rule unit architecture is under development [4]. Figure 4 shows a

Fig. 3 Algorithm for checking the “voice crossing.”

Four simultaneous notes are fed to the rule # n , and checked whether they violate the inhibition rules. If they do not violate it, they are fed to the next rule $n + 1$.

display example of SDS for a given soprano sequence consisting of 14 soprano notes. SDS is implemented in C++ and is a stand-alone system on Windows platform. A main memory more than 1 GB is required for storing all the allowable solutions for a given soprano sequence of an ordinary length. The size of the object file of SDS is about 220 kB. A brief explanation is given in the figure caption.

3.1. Facilities

The current system has following facilities:

- Exhaustive search of possible chord progressions for given soprano sequences
- Rejection of chord progressions having no possible assignment for lower three voices
- Exhaustive search of allowable assignment for lower three voices under specified conditions
- Partial input of user’s answer for efficient reduction of the search space, for example input by initial notes
- Easy search using the wheel on a mouse, looking at score expression on display
- Indication of chord function from a view point of cadences
- Playing the solution on MIDI, saving the solution as a file and printing the score

3.2. Manipulation and system operation

Figure 5 shows a flowchart of manipulation and system operation at consulting the correctness of user’s answer to a given soprano sequence. For simple judgement, users are asked to input their answers through MIDI keyboard or computer keyboard. Input can be partial, such as one, two or three notes in lower three voices in each chord. Then the system first checks if the partial or complete chord sequence given by the user is possible from the view point of adjoining chord transition rules described in Fig. 2. If the system finds impossible chord transition in user’s answer, “No chord progression found” is displayed. If the system finds more than one possibilities of note assignment, users can continue searching his/her answer among the complete set of allowable solutions using additional conditions to reduce the search space or to make the retrieval efficient. As the allowable solutions are stored in the order of depth first search, user can look for his/her answer among the stored solutions by mouse operations without putting in their answers. User manipulations and system operations are almost the same as those of BDS.

4. Performance evaluation

Performance of SDS were evaluated using 4 soprano sequences employing five students majoring in composition. Subjects were asked to prepare their own answers by themselves for 4 soprano sequences before the experiment. Then, subjects were asked to exercise for one hour to get to be able to command SDS using other given soprano sequences. After the training, subjects were asked to confirm if their answers to the given 4 soprano sequences were correct using SDS. Measured is the required time from starting search until reaching the target solution or being sure that his/her answer is not included in the allowable solutions which SDS generates.

Table 1 shows the results. CP, Inv, and L3V represent the

Fig. 4 Displayed is solution #27983 among 31378 allowable solutions under chord progression #86 among 108 actually possible chord progressions in 312 feasible chord progressions in this particular case. Note assignment type (C = Close, O = Open, Oct = Octave or Others) is shown for each chord in the top line, and chord name (I, V, IV, II or VI, with small scripts representing inversion type) with its function (T, D, S) and cadence type (K1, K2 or K3) is displayed between the score and the sequential note ID in the top window. Inversion types are represented as X^6 (first inversion), X^4_4 (second inversion), whereas n -th inversion types are represented as X^n ($n = 1, 2$) in the Japanese standard textbook [3]. The solution ID 27983 is selected in the “Note Assignment ID” box in the left side in the lower half. Selected chord progression ID is displayed on a tree representation in the center window in the lower half. Detailed description of each note assignment can be seen in the right windows using MIDI notations.

Fig. 5 Flowchart of the processing.

average time required until the specified stage is completed as shown in Fig. 5. As the soprano donnee tasks seem rather difficult to subjects, the correct answer ratio was only one eighth. So, in order to evaluate the time required for reaching the target solution even in case they realized their answers were wrong, subjects were given a new correct answer and asked to check if the given answer is included in the set of

Table 1 Average-time required for self-evaluation of students' answers and numbers of allowable chord progressions and solutions (note assignments).

ID	Allowable			Notallowable		
	CP	Inv	L3V	CP	Inv	L3V
1	18	53	187	19	49	—
2	20	56	173	28	62	80
3	21	57	161	18	38	—
4	20	61	151	36	118	—

(s)

CP: Allowable assignment for the specified chord prgession.

Inv: Allowable assignment for a possible chord progression considering inversion.

L3V: Allowable assignment for lower three voices.

allowable solutions. Figures under “Allowable” denote the required time for completion of checking chord progression (represented as CP) and inversion type (Inv), while those of lower three voices (L3V) represent the total required time for reaching the target solution. Those under “Not Allowable” represent the same as those for “Allowable” but those for the case where users' answers were wrong. The symbol “-” in column L3V means that the user realized that their answer is wrong before entering the stage for checking validity of note assignment, i.e. at some time point of checking inversion type.

It is confirmed that they can check correctness of their answers within about 3 minutes.

5. Conclusions

A system that can produce all the allowable solutions to given soprano sequences is described together with its performance evaluation by five subjects on four given soprano sequences. Though the number of subjects and number of tested soprano sequences are small, it can be said that users can judge the correctness of their answers to given soprano sequences without putting all the notes in their answers. For higher level education, however, the system is required to have ability of explaining what part in user's answer is wrong by what reason. This requirement will be fulfilled in the future version of an integrated system designed on the rule unit model [4].

Acknowledgement

This work was partly supported by Grant-in-Aid for Scientific Research (No. C-10680395), Ministry of Education and Science, Japan, and by the Academic Frontier Research Project of Doshisha University.

References

- [1] M. Miura, T. Shimoishizaka, Y. Saiki and M. Yanagida, "A conceptual design of a CAI system for basse donnée in harmony theory," *Int. Conf. Music Perception and Cognition (ICMPC-2000)*, P-1-14 (2000).
- [2] <http://miguel.doshisha.ac.jp/~miura/BDS-e.html>.
- [3] T. Ikenouchi, *Harmony* (Ongaku-no-Tomosha, Tokyo, 1964), Vol. 1, up to p. 67.
- [4] M. Miura, T. Shimoishizaka, Y. Saiki and M. Yanagida, "A study for reconstructing the basse donnée system using a rule unit model," *Acoust. Sci. & Tech.*, **22**, 299–301 (2001).