

Describing Global Musical Structures by Integer Programming on Musical Patterns

Tsubasa Tanaka, Koichi Fujii

▶ **To cite this version:**

Tsubasa Tanaka, Koichi Fujii. Describing Global Musical Structures by Integer Programming on Musical Patterns. Mathematics and Computation in Music , Jun 2015, London, United Kingdom. pp.52-63, 10.1007/978-3-319-20603-5_5 . hal-01264006

HAL Id: hal-01264006
https://hal.science/hal-01264006

Submitted on 14 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Describing Global Musical Structures by Integer Programming on Musical Patterns

Tsubasa Tanaka[1] and Koichi Fujii[2]

[1]Institut de Recherche et Coordination Acoustique/Musique, Paris, France
tsubasa.tanaka@ircam.fr
[2]NTT DATA Mathematical Systems Inc., Tokyo, Japan
fujii@msi.co.jp

Abstract. Music can be regarded as sequences of localized patterns, such as chords, rhythmic patterns, and melodic patterns. In the study of music generation, how to generate musically adequate sequences is an important issue. In particular, generating sequences by controlling the relationships between local patterns and global structures is a difficult and open problem. Whereas the grammatical approaches that represent global structures are suitable to analyze how the pieces are constructed, they are not necessarily designed to generate new pieces with controlling their characteristics of global structures such as the redundancy of the sequence and the statistical distribution of specific patterns. To achieve this, we must overcome the difficulty of solving computationally complex problems. In this paper, we take an integer-programming-based approach and show that some important characteristics of global structures can be described only by linear equalities and inequalities, which are suitable for the integer programming.

Keywords: musical patterns, global structure, hierarchy, redundancy, integer programming

1 Introduction

Music can be regarded as sequences of localized musical patterns such as chords, rhythmic patterns, and melodic patterns. In the study of music generation, it is important to know the characteristics of these sequences. For example, a Markov model is used to learn transition probabilities of localized musical elements in existing pieces or real-time performances [1], and pieces that imitate the original styles are expected to be generated from the model.

However, learning local characteristics is not sufficient to understand or generate music. Global musical structures or musical forms are necessary to be considered. Contrary to Markov models, grammatical approaches such as generative theory of tonal music [2] (GTTM, hereafter), are used to analyze the global structures of musical pieces. In GTTM, a musical piece is abstracted step by step by discarding less important elements and the whole piece is understood as a hierarchical tree structure. For example, Hamanaka et al. have implemented

GTTM on a computer and analyzed musical pieces [3]. However, this model does not have a strategy for composing new pieces. We should note that the grammatical models are not designed to generate new pieces with specifying their characteristics of global structures such as the redundancy of the sequence and the statistical distribution of specific patterns that actually appear in the generated pieces.

Therefore, the objective of this paper is to propose a model that can give specifications of characteristics of global musical structures in the context of music generation. We expect the proposed model to be applied to help users generate new pieces of music and/or obtain desired musical structures.

Compared to the problem of analysis, the problem of generation has a difficulty of a combinatorial explosion of possibilities. When we compose a new piece or sequence, we have to choose a sequence from all the possible combinations of patterns, whose number increases exponentially depending on the number of basic patterns. To deal with such combinatorial problems, we propose an integer-programming-based approach. In order to apply the integer programming, the structural specifications have to be described by linear equalities and inequalities. This is the main challenge for our study.

Integer programming is a framework to solve linear programming problems whose variables are restricted to integer variables or 0–1 variables [4]. Although it shares something in common with constraint programming, which has been often used in the realm of music, one of the advantages of integer programming is that it has an efficient algorithm to find the optimum solution by updating the estimations of the lower and upper bounds of the optimum solution based on the linear programming relaxation technique. Integer programming has been applied to various problems. In our previous study [5], musical motif analysis was formulated as a set partitioning problem, which is a well-known integer programming problem. Thanks to the recent improvements of integer programming solvers such as Numerical Optimizer [6], more and more practical problems have been solved within a reasonable time. We expect that integer programming may also play an important role in the generation of music.

Other possible generation methods than integer programming and constraint programming include the use of metaheuristics. In the study of solving counterpoint automatically [7], counter melodies were generated based on the local search whose objective function is defined as how well the generated counter melody satisfies the rules of counterpoint. In such a search algorithm, there is no guarantee to find the optimum solution. Especially, in the case where there are conflicts between many rules, some rules might be violated. This is not preferable for us because we think that the structural specification should be strictly respected. Therefore, we take an integer-programming approach, which we think is suitable to find strict solutions for the discrete optimization problems.

There are several limitations about what we can describe in this paper. Three main limitations are as follows: (1) we do not describe the objective function and only focus on the constraints to be satisfied strictly (however, we are planning to introduce the objective function in the future work). (2) We do not propose

a comprehensive formulation that covers various situations. We would rather explain our point of view based on a concrete examples of typical musical structure. (3) We only focus on the aspect of formulation and do not proceed to the steps of generation and evaluation.

The rest of this paper is organized as follows: In section 2, we show the relationship between localized musical patterns and hierarchical global structure referring to the chord progression of a piece that has typical phrase structures. In section 3, we describe how to implement this relationship by linear equalities and inequalities. In section 4 and section 5, we describe the similar relationships for the structures of rhythmic pattern sequence and intervallic pattern sequence by linear constraints. Then, these are combined to formulate the constraints for generating a melody. Section 6 gives some concluding remarks.

2 Hierarchy and Redundancy

From the micro-level point of view, composing a piece of music can be regarded as determining how the sequences of localized musical patterns are arranged. If such sequences are seen from the macro-level point of view, they can be regarded as the musical forms.

Then, a question arises: what are the global characteristics that a musical sequence should have to construct a global musical form and to generate well-organized music? In this section, we study the piece Op. 101 No. 74 by Ferdinand Beyer to consider this problem. The chord sequence of this piece is clearly related to the phrase structures and the musical form, and it will provide a good clue for this problem.

The chord sequence of this piece is TTSTTTDTSTSTTTDTDTDTDTDT, using T (tonic), S (subdominant), and D (dominant). Each chord corresponds to one measure. To see the phrase structures more clearly, let's combine each two bars, four bars, and eight bars. Then these four levels can be represented as the following sequences:

- 1^{st} level: $A_1A_1A_2A_1A_1A_1A_3A_1A_2A_1A_2A_1A_1A_1A_3A_1A_3A_1A_3A_1A_3A_1A_3A_1$ (3/24)
- 2^{nd} level: $B_1B_2B_1B_3B_2B_2B_1B_3B_3B_3B_3B_3$ (3/12)
- 3^{rd} level: $C_1C_2C_3C_2C_4C_4$ (4/6)
- 4^{th} level: $D_1D_2D_3$ (3/3)

Here, the same indices indicate the same patterns. For example, A_1 =T, A_2 =S and A_3 =D. There are relationships between the consecutive levels such as $B_1 = A_1A_1$, $C_1 = B_1B_2$, $D_1 = C_1C_2$, and so on (The notations like "A_1A_1" represent the concatenations of the patterns and do not indicate multiplications). The numerators of the fractions indicated between parentheses after each sequence indicate the numbers of the variety of patterns that exist in the sequences in each level, and the denominators indicate the lengths of each level. We call the inverse numbers of these fractions redundancies[1]. Fig. 1 visualizes

[1] The study [8] focuses on the redundancy of musical sequence, and models the musical style by referring to the Lempel-Ziv compression algorithm. We extend this perspective and pay attention to the redundancies of multiple levels simultaneously.

this hierarchical structure. Because the fourth level has three different elements,

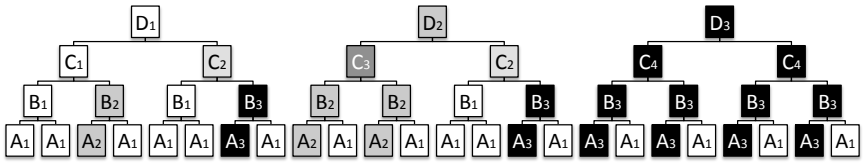


Fig. 1. Hierarchical structure of chord sequence of Beyer No. 74.

we see that this piece consists of three different parts. When observing the relationship between the third level and fourth level, we find that the latter parts of D_1 and D_2 are the same and D_3 consists of a repetition of C_4.

We can observe that there are many repetitions of the same patterns and that the lower the level, the higher the redundancy. In addition, T is more than twice as frequent as D, and D is twice as frequent as S. These biases of frequency are also notable characteristics of this sequence. Thus, the global structure of this piece can be regarded to be, at least, constrained by hierarchy, redundancies, and frequencies of patterns. Without repetitions in lower levels, the music will become unmemorable and lose the attention of human listeners. Moreover, repetitions in higher levels are related to known musical forms such as A-B-A ternary form or rondo form. Thus, frequency and redundancy are important fearures for describing musical structures.

Focusing on each level, state transition diagrams can be depicted per level (Fig. 2). From the global point of view, state transitions of multiple levels should

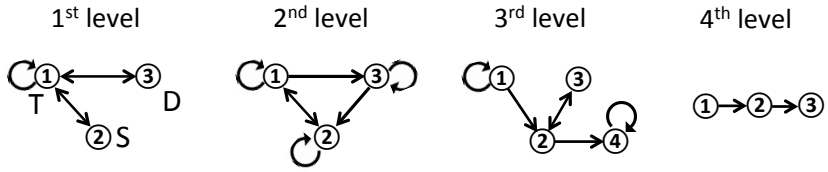


Fig. 2. Transition diagram of chord patterns in each level.

be considered simultaneously. In these diagrams, the lowest level (first level) represents the rules of chord progression proper to this piece. Although the transition from S to D is possible in ordinary rules of harmony, this transition is not used in this piece.

3 Implementation of Constraints on Chord Progression

It is difficult to find new sequences that have specific redundancies in respective levels, because the constraints for different levels may cause conflicts during the process of searching for the solutions. Also, the naive method of enumerating possible sequences one by one and checking whether or not they satisfy specific

redundancies in each level won't be realistic because the number of possible sequences increases exponentially with the length of the sequence based on the number of the states in the lowest level.

To deal with this problem, we take an integer-programming-based approach, which has good search algorithms to practically solve computationally complex problems. In order to use integer programming, the characteristics of the sequence should be expressed by linear constraints (equalities and inequalities). In this section, we demonstrate that some important characteristics of the sequence observed in the Beyer's piece can be expressed by linear constraints.[2]

Let A, B, C, D be the set of patterns that can appear in respective levels. The number of elements in each level is denoted by K_1, K_2, K_3, and K_4. For explanation, we set these possible patterns as the patterns that actually appear in the Beyer's piece (i.e., $A = \{A_1, A_2, \cdots A_3\}$, $B = \{B_1, B_2, \cdots, B_3\}$, $\cdots$.). [3] Let a_t be a variable that corresponds to the t-th element in the sequence of variables for the patterns of the first level, b_t for those of the second level, c_t for those of the third level, and d_t for those of the fourth level. For example, in the original piece, $a_1 = A_1$, $a_2 = A_1$, $a_3 = A_2$, $a_4 = A_1$, $\cdots$, $a_{24} = A_1$, $b_1 = B_1$, $b_2 = B_2$, $b_3 = B_1$, $b_4 = B_2$, $\cdots$, $b_{12} = B_3$, and so on.

The followings are important characteristics of the original sequence explained in Section 2:

1. Hierarchy of phrase structures (relationships between neighboring levels): $b_t = a_{2t-1}a_{2t} (1 \leq t \leq 12)$, $c_t = b_{2t-1}b_{2t} (1 \leq t \leq 6)$, $d_t = c_{2t-1}c_{2t} (1 \leq t \leq 3)$.
2. Variety of patterns that appear in each level: the first level has three patterns, the second has three patterns, the third has four patterns, and the fourth has three patterns.
3. Frequency of each state in each level (in the first level, these are the frequency of each chord).
4. State transition rules (in the first level, these are the chord progression rules).

In this section, we show how to formalize these rules by variables and linear constraints on the variables.

3.1 Constraints of Hierarchical Phrase Structures

Let $x_{t,i}$, $y_{t,i}$, $z_{t,i}$, and $w_{t,i}$ be 0–1 variables that represent whether or not the t-th element of the sequence for each level is the pattern i (For example, the pattern

[2] In this paper, in order to avoid the explanation from being complicated, we only treat the stereotype examples of musical pieces whose groupings are always combinations of two consecutive elements in every level. However, in practice, such structures should vary depending on the specifications of the pieces that the user wants to create. For example, we can think of the case where the number of combination in the groupings are different between the levels. We can also think of the case where the consecutive patterns can be overlapped as is mentioned in [9]. How to formulate such cases is an important future issue.

[3] In practice, it is not necessary to stick to existing pieces.

i indicates A_i, in the case of the first level). $x_{t,i}$, $y_{t,i}$, $z_{t,i}$, and $w_{t,i}$ correspond to the first, second, third, and fourth levels respectively. For example, $x_{t,i} = 1$ if $a_t = A_i$, and otherwise $x_{t,i} = 0$. $y_{t,i} = 1$ if $b_t = B_i$ and otherwise $y_{t,i} = 0$. $z_{t,i}$ and $w_{t,i}$ are also defined similarly. a_t, b_t, c_t, and d_t take one of the values in A, B, C, and D, respectively. The numbers of elements of A, B, C, and D are denoted by K_1, K_2, K_3, and K_4, respectively. Because a_t, b_t, c_t, and d_t take only one value at a location t, respectively (here, location t means t-th element in the sequence of each level), they satisfy the following constraints:

$$\sum_{i=1}^{K_1} x_{t,i} = 1 \ (\forall t, 1 \le t \le 24), \quad \sum_{i=1}^{K_2} y_{t,i} = 1 \ (\forall t, 1 \le t \le 12),$$
$$\sum_{i=1}^{K_3} z_{t,i} = 1 \ (\forall t, 1 \le t \le 6), \quad \sum_{i=1}^{K_4} w_{t,i} = 1 \ (\forall t, 1 \le t \le 3). \tag{1}$$

Now the hierarchy can be expressed by constraints on $x_{t,i}$, $y_{t,i}$, $z_{t,i}$, and $w_{t,i}$. For example, if $B_i = A_{j_1}A_{j_2}$, the statement "$b_t = B_i$ (i.e. $y_{t,i} = 1$)" must be equivalent to "$a_{2t-1} = A_{j_1}$ and $a_{2t} = A_{j_2}$ (i.e., $x_{2t-1,j_1} = 1$ and $x_{2t,j_2} = 1$)", and this equivalence can be expressed by the three constraints: $y_{t,i} \le x_{2t-1,j_1}$, $y_{t,i} \le x_{2t,j_2}$, and $x_{2t-1,j_1} + x_{2t,j_2} - 1 \le y_{t,i}$. Therefore, the constraints that correspond to the whole hierarchy are described as follows:

for all (i, j_1, j_2) that satisfy $B_i = A_{j_1}A_{j_2} (1 \le i \le K_1)$,

$$y_{t,i} \le x_{2t-1,j_1}, \ y_{t,i} \le x_{2t,j_2}, \ x_{2t-1,j_1} + x_{2t,j_2} - 1 \le y_{t,i} \ (1 \le t \le 12), \tag{2}$$

for all (i, j_1, j_2) that satisfy $C_i = B_{j_1}B_{j_2} (1 \le i \le K_2)$,

$$z_{t,i} \le y_{2t-1,j_1}, \ z_{t,i} \le t_{2t,j_2}, \ y_{2t-1,j_1} + y_{2t,j_2} - 1 \le z_{t,i} \ (1 \le t \le 6), \tag{3}$$

for all (i, j_1, j_2) that satisfy $D_i = C_{j_1}C_{j_2} (1 \le i \le K_3)$,

$$w_{t,i} \le z_{2t-1,j_1}, \ w_{t,i} \le z_{2t,j_2}, \ z_{2t-1,j_1} + z_{2t,j_2} - 1 \le w_{t,i} \ (1 \le t \le 3). \tag{4}$$

Finally, the statement that every element in the last level is different can be expressed by the following constraints:

$$w_{1,i} + w_{2,i} + w_{3,i} \le 1 \ (\forall i, 1 \le i \le K_4). \tag{5}$$

3.2 Constraints on Frequencies of Each Patterns

The constraints to control the frequency (or the range of frequency) of each state can be described by the following inequalities using 0–1 variables α_i, β_i, γ_i, and δ_i for each level:

$$L_{1,i}\alpha_i \le \sum_{t=1}^{24} x_{t,i} \le H_{1,i}\alpha_i \ (\forall i, 1 \le i \le K_1), \tag{6}$$

$$L_{2,i}\beta_i \leq \sum_{t=1}^{12} y_{t,i} \leq H_{2,i}\beta_i \ (\forall i, 1 \leq i \leq K_2), \tag{7}$$

$$L_{3,i}\gamma_i \leq \sum_{t=1}^{6} z_{t,i} \leq H_{3,i}\gamma_i \ (\forall i, 1 \leq i \leq K_3), \tag{8}$$

$$L_{4,i}\delta_i \leq \sum_{t=1}^{3} w_{t,i} \leq H_{4,i}\delta_i \ (\forall i, 1 \leq i \leq K_4), \tag{9}$$

where $L_{1,i}$, $L_{2,i}$, $L_{3,i}$, $L_{4,i}$, $H_{1,i}$, $H_{2,i}$, $H_{3,i}$, and $H_{4,i}$ are the constants[4] for lower and upper bounds in the case that the i-th element of each level appears. The statement "$\alpha_i = 0$" is equivalent to "$x_{t,i} = 0$ for all $t(1 \leq t \leq 24)$". This means that α_i represents whether or not the state a_i appears in the sequence. β_i, γ_i, and δ_i also have such meanings in their own levels.

Also, relative differences of frequencies that T is more than twice or twice as frequent as D and D is twice as frequent as S can be expressed by the following constraints:

$$\sum_{t=1}^{24} x_{t,1} \geq 2\sum_{t=1}^{24} x_{t,3}, \quad \sum_{t=1}^{24} x_{t,3} = 2\sum_{t=1}^{24} x_{t,2} \tag{10}$$

3.3 Constraints on Varieties of Patterns

Using the variables α_i, β_i, γ_i, and δ_i, the number of variety of patterns in each level can be described. For example, $\sum_i \alpha_i$ indicates the number of variety of patterns in the first level. Similarly, the statement that the numbers of variety of patterns in respective levels are 3, 3, 4, and 3 can be represented as:

$$\sum_{i=1}^{K_1} \alpha_i = 3, \quad \sum_{i=1}^{K_2} \beta_i = 3, \quad \sum_{i=1}^{K_3} \gamma_i = 4, \quad \sum_{i=1}^{K_4} \delta_i = 3. \tag{11}$$

3.4 Constraints on State Transitions

The possibility of state transitions in the first level can be controlled by posing the following inequality for all of the combinations of (t, i, j) whose transition from i to j at the location t is prohibited:

$$x_{t,i} + x_{t+1,j} \leq 1. \tag{12}$$

The same is true in other levels. One way to determine the allowed transitions is to prohibit the transitions that do not occur in the original piece.

Thus, linear constraints can describe both global structures and local state transitions.

[4] For example, they can be set depending on user's preference or statistics of the original piece. If $L_{j,i} \leq 1$ and $H_{j,i}$ is larger than or equal to the length of the sequence, these equations give no limitation to the number of each pattern that appears in the sequence.

4 Constraints on Rhythmic Patterns

In this section, the hierarchical structure of rhythmic patterns is illustrated in a similar way to the hierarchical structure of chords. Beethoven's famous melody *Ode to Joy* is used as an model example of monophony (Fig. 3).
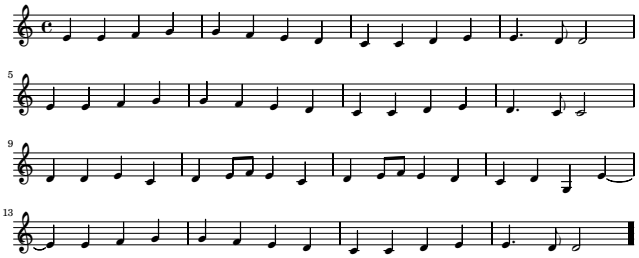


Fig. 3. Beethoven's melody *Ode to Joy*.

Adopting a beat as a unit length, the different elements of the first level are $A_1 = [f]$, $A_2 = [f\sim]$, $A_3 = [\sim e, e]$, $A_4 = [\sim f]$, and $A_5 = [e, e]$, where e, f, and s represent 8th note, 4th note, 2nd note, respectively, and "." and "$\sim$" represent a "dot" and a "tie," respectively. Each A_i represents one beat and does not necessarily correspond to the actual notations (the "ties" and "dots" actually correspond to the ties or prolongations of a note beyond the beat). The second level consists of $B_1 = [f, f]$, $B_2 = [f., e]$, $B_3 = [s]$, $B_4 = [f, e, e]$, $B_5 = [f, f\sim]$, and $B_6 = [\sim f, f]$. The entire hierarchy is as follows:

- a_t (1^{st} level): $||A_1A_1A_1A_1|A_1A_1A_1A_1|A_1A_1A_1A_1|A_2A_3A_2A_4|| \cdots$ (5/64)
- b_t (2^{nd} level): $||B_1B_1|B_1B_1|B_1B_1|B_2B_3|| \cdots$ (6/32)
- c_t (3^{rd} level): $||C_1|C_1|C_1|C_2||C_1|C_1|C_1|C_2||C_1|C_3|C_3|C_4||C_5|C_1|c_1|c_2||$ (5/16)
- d_t (4^{th} level): $||D_1D_2||D_1D_2||D_3D_4||D_5D_1||$ (5/8)
- e_t (5^{th} level): $||E_1||E_1||E_2||E_3||$ (3/4)
- f_t (6^{th} level): $||F_1\ F_2||$ (2/2)

where, "|" is a bar line and "||" is a four-bar partition. This hierarchy can be described by linear constraints in a similar way to the previous section. However, this case has more levels than the previous case. In general, the number of possible patterns increases drastically when looking at a higher level. If the number of levels is too large, the number of variables for the level may become too large.

Therefore, we introduce an alternative way to describe the constraints for redundancies of higher levels using the variables for the first level $x_{t,i} (1 \leq i \leq K_1)$ (without using $y_{t,i}$, $z_{t,i}$, and $w_{t,i}$).[5]

[5] However, at the current moment, we do not know an alternative way to implement the constraints for state transitions and frequencies of each pattern in the high levels.

4.1 Alternative Way to Control Redundancies of High Levels

Let us consider a 0–1 variable $s_{n,t}$ whose value is 1 if and only if an element in a level n first appear at the location t. Then, $\sum_t s_{n,t}$ represents how many different elements appear in level n. This offers an alternative way to specify the redundancies. Our purpose here is to construct the variables $s_{n,t}$ that have such meaning. In order to do so, we introduce other subsidiary variables q_{n,t_1,t_2} and $r_{n,i,u1,u2}$.

q_{n,t_1,t_2} is a 0–1 variable that represents whether the sections t_1 and t_2 ($t_1 < t_2$) in level n are the same or not (we call each content of the time span of an element such as b_t, c_t, $\cdots$ a section). $q_{n,t_1,t_2} = 1$ means that all of corresponding elements a_{u_1} and a_{u_2} in the sections t_1 and t_2 in level n are the same. Therefore, it is necessary to introduce the constraints that make the statement "$q_{n,t_1,t_2} = 1 \Longleftrightarrow x_{u_1,i} = x_{u_2,i}$ $(\forall(i, u_1, u_2)$ s.t. $i \in [1, K_n]$, $(u_1, u_2) \in D(n, t_1, t_2))$" true, where $D(n, t_1, t_2)$ represents the range of the combinations (u_1, u_2) where a_{u_1} and a_{u_2} are all of the pairs of corresponding elements in the sections t_1 and t_2 in level n. This statement can be replaced by the statement "$r_{n,i,u_1,u_2} = 1$ $(\forall(i, u_1, u_2)$ s.t. $i \in [1, K_n]$, $(u_1, u_2) \in D(n, t_1, t_2)) \Longleftrightarrow q_{n,t_1,t_2} = 1$," where the 0–1 variable r_{n,i,u_1,u_2} means whether $x_{u_1,i} = x_{u_2,i}$ or not.

Here, the statement "$r_{n,i,u_1,u_2} = 1 \Longleftrightarrow x_{u_1,i} = x_{u_2,i}$," can be expressed by the following constraints:

$$r_{n,i,u_1,u_2} \leq 1 + x_{u_1,i} - x_{u_2,i}, \tag{13}$$

$$r_{n,i,u_1,u_2} \leq 1 - x_{u_1,i} + x_{u_2,i}, \tag{14}$$

$$1 - x_{u_1,i} - x_{u_2,i} \leq r_{n,i,u_1,u_2}, \tag{15}$$

$$-1 + x_{u_1,i} + x_{u_2,i} \leq r_{n,i,u_1,u_2}. \tag{16}$$

Also, the statement "$r_{n,i,u_1,u_2} = 1$ $(\forall(i, u_1, u_2)$ s.t. $i \in [1, K_n]$, $(u_1, u_2) \in D(n, t_1, t_2)) \Longleftrightarrow q_{n,t_1,t_2} = 1$," can be expressed by the following constraints:

$$1 - \sum_{i \in [1,K_n]} \sum_{(u_1,u_2) \in D(n,t_1,t_2)} (1 - r_{n,i,u1,u2}) \leq q_{n,t_1,t_2}, \tag{17}$$

$$q_{n,t_1,t_2} \leq r_{n,i,u_1,u_2} \quad (\forall(i, u_1, u_2)\ s.t.\ i \in K_n,\ (u1, u2) \in D(n, t_1, t_2)). \tag{18}$$

Let $v_{n,t}$ be $\sum_{t_1<t} q_{n,t_1,t}$, then $v_{n,t}$ represents how many the same sections as t there are before the section t in level n. $v_{n,t} = 0$ means that the content of section t first appears. Therefore, the statement "$v_{n,t} = 0 \Longleftrightarrow s_{n,t} = 1$" must be true to let $s_{n,t}$ have the proper meaning. This statement can be expressed by a constraint:

$$1 - s_{n,t} \leq v_{n,t} \leq M \cdot (1 - s_{n,t}), \tag{19}$$

where M is a sufficiently large constant number. Under the constraints above, we can control the redundancy of each level n by a constraint:

$$\sum_t s_{n,t} = Constant. \tag{20}$$

5 Constraints on Melodic Patterns

In this section, hierarchical structure of the melodic patterns, which has information of pitches and rhythms, is illustrated using the same melody *Ode to Joy*. Although pitch information and rhythmic information in melodic patterns are not completely independent of each other, the hierarchical structure of pitch patterns and rhythmic patterns are different in general. Therefore, we treat these two hierarchical structures separately. After that, we consider compatibility between them and combine them. Considering that the same pitch patterns can occur in different transpositions on the scale, we treat pitch information as the series of intervals. In section 5.1, we introduce constraints on structure of intervallic patterns. In section 5.2, constraints for controlling pitch range is introduced. Then, in section 5.3, we describe how to combine the rhythmic structure and the intervallic structure to generate melodies.

5.1 Constraints on Intervallic Patterns

There are nine different one-beat intervallic patterns that appear in the piece *Ode to Joy*. The intervals are denoted by the interval numbers on the scale - 1, based on the diatonic scale[6]. These are [0], [1], [−1], [∼], [−1, 0], [−2], [1, −1], [−4], [5], which are denoted by $A_1 \sim A_9$, respectively. These are series of intervals that start from the interval between the first pitch and the second pitch of the beat to the interval between the last pitch to the first pitch of the next beat. [∼] means the interval 0 by a tie or a prolongation of a pitch to the next pattern. Though there is no beat after the last beat, the last beat obviously corresponds to the last beats of the bar 4 and bar 8. Therefore, we regard the last beat as the same intervallic pattern as the last beats of the bar 4 and bar 8, namely [1]. The sequences of each level is as follows:

- a_t (1^{st} level): $||A_1A_2A_2A_1|A_3A_3A_3A_3|A_1A_2A_2A_1|A_4A_5A_4A_1|| \cdots$ (9/64)
- b_t (2^{nd} level): $||B_1B_2|B_3B_3|B_1B_2|B_4B_5|| \cdots$ (10/32)
- c_t (3^{rd} level): $||C_1|C_2|C_1|C_3||C_1|C_2|C_4|C_3||C_5|C_6|C_7|C_8||C_1|C_2|C_1|C_3||$ (8/16)
- d_t (4^{th} level): $||D_1D_2||D_1D_3||D_4D_5||D_1D_2||$ (5/8)
- e_t (5^{th} level): $||E_1||E_2||E_3||E_1||$ (3/4)
- f_t (6^{th} level): $||F_1\ F_2||$ (2/2)

Comparing, for example, the sequence of the fifth level of the rhythmic patterns $||E_1||E_1||E_2||E_3||$) and the sequence of the intervallic patterns ($||E_1||E_2||E_3||E_1||$), we see that the first two elements are the same and the first and last elements are not the same in the former sequence. On the other hand, the first two elements are not the same and the first and the last elements are the same in the latter sequence. This difference adequately represents that although the first, second, and fourth 4-bars phrases are almost the same, the first and second have

[6] Here, we can also represent the pitches and the intervals based on the chromatic scale. However, we use the scale degrees and the interval numbers based on the diatonic scale because that is more efficient.

the same rhythms but slightly different intervals and the first and last 4-bars phrases have the same intervals but slightly different rhythms. This is the benefit of treating the rhythmic structure and the intarvallic structure separately.

5.2 Constraints for Pitch Range

The intervallic structures described in the previous section do not have the limitation of the pitch range. This problem occurs because the intervallic patterns are based on relative intervals instead of absolute pitches. Therefore, in this subsection, we introduce extra variables and constraints for bounding the pitch range.

Let p_t be the first pitch of the pattern at the location t of the sequence. The pitch is counted on the scale, where the starting pitch p_1 is set as 0. In the case of *Ode to Joy*, C $= -2$, D $= -1$, E $= 0$, etc. The scale of the piece is {-5, -4, -3, -2, -1, 0, 1, 2}. Let $Intvl(A_i)$ be the total interval of the intervallic pattern A_i (e.g., $Intvl(A_7) = 0$, because the total interval of $A_7(= [1, -1])$ is 0 $(= 1 - 1)$). Then, p_t is equal to the accumulation of the total intervals from the first pattern to the $(t-1)$-th pattern as in the following equation:

$$p_t = \sum_{t_1<t} Intvl(a_{t_1}) = \sum_{t_1<t} \sum_i Intvl(A_i) \cdot x_{t_1,i} \tag{21}$$

Let $Int(i, j)$ be the total interval from the beginning of A_i to the end of jth interval. If $a_t = A_i$, the pitch after the jth interval of A_i in a_t is $p_t + Int(i, j)$. If the next constraint:

$$LB_t \leq p_t + Int(i, j)x_{t,i} \leq UB_t \tag{22}$$

is satisfied for all (t, i, j), the upper bound UB_t and lower bound LB_t for pitches in t-th element of the sequence can be set.

5.3 Compatibility of Rhythmic Pattern and Intervallic Pattern

Although the sequences of rhythmic patterns and intervallic patterns have been independently introduced in the previous subsections, it is necessary to combine these two sequences to complete a melody, which contains both rhythms and intervals. To combine both of the sequences, compatibility between rhythmic patterns and intervallic patterns must be assured. For example, [s] and [-1,0] are not compatible because the number of elements differs ([s] indicates that there is only one note in this rhythmic pattern, and [-1,0] indicates that there are two notes in the intervallic pattern).

Therefore, we introduce constraints to assure that both sequences can coexist and propose a formulation to generate both sequences simultaneously. Let's discriminate variables and constants for intervallic patterns from those of rhythmic patterns by adding a dash on the shoulder of the variable names. The compatibility between both sequences depends on whether a_t and a'_t are compatible in

every location t or not. Therefore, the compatibility can be expressed by the following inequality for all combination A_{i_1} and A'_{i_2} that are not compatible:

$$x_{t,i_1} + x'_{t,i_2} \leq 1. \tag{23}$$

If we find a solution for the problem that is a combination of the problems for rhythmic patterns and intervallic patterns with these inequalities, we will be able to obtain a complete melody.

6 Conclusion

In this paper, we proposed a formulation to generate sequences of musical patterns controlling some global structures of the sequences especially focusing on hierarchy and degree of redundancy in each level. We showed that such structures can be expressed only by linear constraints, which are necessary to apply integer programming. Future tasks include describing global structures more comprehensively, implementation of the constraints and actually generating new pieces, defining constraints on the relationships between melody and chords, and defining the constraints for polyphonic relationships.

Acknowledgments. This work was supported by JSPS Postdoctoral Fellowships for Research Abroad.

References

1. Pachet, F.: "The Continuator: Musical Interaction With Style," Journal of New Music Research, Vol.32, No.3, pp.333-341 (2003).
2. Lerdahl, F., and Jackendoff, R.: A Generative Theory of Tonal Music, MIT Press (1983).
3. Hamanaka, M., et al.: "Implementing "A Generative Theory of Tonal Music,"" Journal of New Music Research, Vol.35, No.4, pp.249-277 (2006).
4. Nemhauser, G.L. and Wolsey, L.A.: Integer and Combinatorial Optimization, Wiley (1988).
5. Tanaka, T. and Fujii K.: "Melodic Pattern Segmentation of Polyphonic Music as a Set Partitioning Problem," Proceedings of International Congress on Music and Mathematics (to be published).
6. http://msi.co.jp/nuopt/
7. Herremans, D. and Srensen, K.: "A variable neighbourhood search algorithm to generate first species counterpoint musical scores," Working Paper, University of Antwerp Faculty of Applied Economics Operations Research Group ANT/OR, (2011).
8. Lartillot, O., et al.: "Automatic Modeling of Musical Style," 8èmes Journées d'Informatique Musicale, pp.113-119 (2001).
9. Mazzola, G., et al.: The Topos of Music: Geometric Logic of Concepts, Theory, and Performance, Birkhäuser (2002).