Learning of Hierarchical Temporal Structures for Guided Improvisation

Ken Déguernel,* Emmanuel Vincent,† Jérôme Nika,* Gérard Assayag,* and Kamel Smaïli†

*Institut de Recherche et Coordination Acoustique/Musique
1 Place Igor Stravinsky
75004 Paris, France
†Inria
Université de Lorraine
615 Rue du Jardin Botanique
54600 Villers-lès-Nancy, France
{ken.deguernel, jerome.nika, gerard.assayag}@ircam.fr,
emmanuel.vincent@inria.fr,
kamel.smaili@loria.fr

Abstract: This article focuses on learning the hierarchical structure of what we call a "temporal scenario" (for instance, a chord progression) to perform automatic improvisation consistently over several different time scales. We first present a way to represent hierarchical structures with a phrase structure grammar. Such a grammar enables us to analyze a scenario at several levels of organization, creating a "multilevel scenario." We then develop a method to automatically induce this grammar from a corpus, based on sequence selection with mutual information. We applied this method to a corpus of transcribed improvisations based on the chord sequence, also with chord substitutions, from George Gershwin's "I Got Rhythm." From these we obtained multilevel scenarios similar to the analyses performed by professional musicians. We then present a novel heuristic approach, exploiting the multilevel structure of a scenario to guide the improvisation with anticipatory behavior in an improvisation paradigm driven by a factor oracle. This method ensures consistency of the improvisation with regard to the global form, and it opens up possibilities when playing on chords that do not exist in memory. This system was evaluated by professional improvisers during listening sessions and received excellent feedback.

In idiomatic music, when improvising on a chord progression, musicians use knowledge of the form of the piece being played to build their musical discourse in a consistent way upon several levels of organization. For instance, in a jazz tune the chord progression is often structured upon different time scales; some groups of chords (short scale) can create tonal or modal functions (medium scale), and these functions can be organized into several sections (long scale). In the following, we use the word "level" rather than "scale" to describe the temporal organization to avoid confusion with the notions of tonality and modality. Our goals are, first, to perform an automatic analysis of the hierarchical structure of a chord progression, and then to use this structure to guide an automatic improvisation system.

Over the years, several generative models have been developed for machine improvisation, such as statistical sequence models (Dubnov, Assayag, and El-Yaniv 1998), Markov models (Pachet 2002), deep neural networks (Bickerman et al. 2010), and factor oracles (Assayag et al. 2006; Déguernel, Vincent, and Assayag 2018). More recently, research has focused on "guided improvisation" in which the improvisation system relies both on a generative model representing the musical style and on prior knowledge of a sequential structure we will call a *scenario*.

Gillick, Tang, and Keller (2010) used inference of a probabilistic context-free grammar to generate melodies over a given chord progression. Pachet and Roy (2011) used a set of constraints to generate blues chord progressions and to generate melodies using scales specific to a given musical style. A similar method was used by Roy and Pachet (2013) to force an improvisation to comply with the bars of a specific tune. Papadopoulos, Roy, and Pachet (2016) developed a computer program, FlowComposer, in which the user can choose a stylistic corpus and set structural constraints on melody and harmony; the system then generates

Computer Music Journal, 43:2/3, pp. 109–124, Summer/Fall 2019
doi:10.1162/COMJ_a_00521

a melody and a chord progression following the given style and constraints. Donzé et al. (2014) built harmonic and rhythmic constraint automata given a chord progression and melodic and rhythmic factor oracles to model the style of a musician. Improvising in this system consists of finding a path in the factor oracles that also satisfies the constraints enforced by the automata.

Although these systems describe a global structure, they only enforce structures on a local level when generating an improvisation and do not benefit from anticipatory behavior.

Keller et al. (2012) proposed the use of idiomatic harmonic bricks to guide improvisation. Chord progressions are analyzed and organized in harmonic functions, creating a scenario with "harmonic-brick" hierarchies, upon which melodic improvisations are generated. Although this method takes higher-level structure into account, it still does not benefit from anticipatory behavior.

In their software ImproteK, Nika, Chemillier, and Assayag (2017a) introduced anticipatory behaviors using prior knowledge of the scenario by generating improvisations taking the future of the scenario into account while ensuring consistency with the past of the improvisation. The scenarios used in ImproteK are mere sequences of symbols, however, and global form is not considered in the generative process. This can lead to a feeling of inconsistency in the long run.

Form analysis is a major topic in music information retrieval and has been studied with methods from different fields. Giraud et al. (2015) used dynamic programming to perform fugue analysis by detecting the occurrences of each motif (subject, countersubject, etc.). A similar method was used for analysis of sonata form by Bigo et al. (2017). Bimbot et al. (2016) introduced a "System-and-Contrast" model, in which a musical segment is divided into morphological elements. Relations between these elements are represented as a regular polytope.

Generative grammars have been used to model the form of a tune. Mark Steedman (1996) proposed a grammar based on rewrite rules for jazz music. This grammar was used in ImproteK to create new instances of a given progression, but only at a local level. De Haas et al. (2009) used a context-free grammar of tonal harmony to create a hierarchical analysis of chord progressions at four different temporal levels. This grammar was used to compute harmonic similarity between two chord progressions. Keller et al. (2013) used idiomatic harmonic bricks to describe some hierarchical aspects of a chord progression using a modified version of the Cocke-Younger-Kasami algorithm. Whereas these methods are partially based on empirically created grammars, Corentin Guichaoua (2017) proposed a method of grammar induction using only the leaves of a grammar tree, based on the minimal-grammar problem (Gallé 2011). Unfortunately, this method is purely symbolic and, as a result, is not robust when encountering variations.

In this article, we present two contributions.

First, we present the automatic inference of a phrase structure grammar (Chomsky 1957) to represent the hierarchical temporal structure of a scenario, creating what we call a *multilevel scenario*. We put forward a probabilistic grammar-induction process for musical sequences based on the sequence selection method developed by Zitouni, Smaili, and Haton (2000), originally designed for written text. This method is used on a corpus of scenarios of a given form, enabling us to find common structures and to prevent the modeling problems encountered with local variations of a given motif. Once the grammar has been trained, any sequence of symbols representing a scenario of the given form can be reformulated as a multilevel scenario represented by multilevel labels. We applied this method to a corpus of tunes using the *Rhythm Changes* and compared the generated grammar with a ground-truth grammar created in collaboration with a professional musician. (The Rhythm Changes are a well-known chord progression used in many jazz compositions, based on the chords used by George Gershwin in "I Got Rhythm," and the progression is widely used in bebop.)

Second, we propose new heuristics that extend the work by Nika et al. (2017b) on ImproteK to exploit a multilevel scenario to guide improvisation in an improvisation paradigm driven by a factor oracle. These heuristics were first introduced by Déguernel et al. (2017) and consist in performing anticipation of the scenario and navigating in memory over several temporal levels by considering *equivalent labels*. These are multilevel labels sharing common information with the target scenario at one or more levels. In this article, we extend this preliminary study with formalization of these heuristics at a greater depth, and we conduct an evaluation of our system by means of listening sessions with professional improvisers.

In the first main section of this article, we present how a grammar based on a constituent analysis can represent the hierarchical structure of a temporal scenario to create multilevel scenarios. We create such a grammar in collaboration with a professional musician for the Rhythm Changes chord progression, considered here as a multilevel scenario. In the second section we present a method for using machine learning for grammar induction based on probabilistic methods of word sequence selection with mutual information. We apply this method on a corpus of compositions based on the Rhythm Changes and compare the generated grammar with our ground truth. In the third section, we present a new heuristic for generating improvisations on multilevel scenarios based on the notion of scenario anticipation. The generative model is evaluated through listening sessions and interviews with professional jazz musicians.

Using a Grammar to Model a Multilevel Structure

In this section, we present the use of a grammar to represent the hierarchical structure of a chord progression on multiple time levels. First, we present the type of grammars we are going to use: phrase structure grammars. Then we create, in collaboration with a professional musician, a phrase structure grammar representing the multilevel aspect of the Rhythm Changes.

Phrase Structure Grammar

Following Hopcroft and Ullman (1979), we consider a set of symbols X, denoting the set of finite sequences of these symbols as X^*. A grammar $G = (\Sigma, N, R, s)$ is defined by a set Σ of terminal symbols called an alphabet, a set N of nonterminal symbols that is disjoint from Σ, a singular element $s \in N$ that is the start symbol of the grammar, and a finite set of rewrite rules $R \subset (N \cup \Sigma)^* N (N \cup \Sigma)^* \to (N \cup \Sigma)^*$. A rewrite rule, usually denoted $u \to v$, can be seen as an instruction meaning "rewrite u as v."

Figure 1. Example of a phrase structure grammar (adapted from Chomsky 1957, p. 26).

$$\begin{aligned}
&1:\ Sentence \to NP\ \ VP\\
&2:\ NP \to Article\ \ Noun\\
&3:\ VP \to Verb\ \ NP\\
&4:\ Article \to \text{a} \mid \text{the} \ldots\\
&5:\ Noun \to \text{man} \mid \text{ball} \ldots\\
&6:\ Verb \to \text{hit} \mid \text{took} \ldots
\end{aligned}$$

A *phrase structure grammar* is a particular type of grammar based on a linguistic description of a language at a syntactic level, using a constituent analysis (i.e., a breakdown of the linguistic functions following a hierarchical structure). Noam Chomsky (1957) presented an example of a phrase structure grammar for constructing simple sentences in English, displayed in Figure 1. In this example, the nonterminal symbols are written in italics and the terminal symbols are written in roman type. For instance, Rule 1 can be read as "rewrite *Sentence* as '*NP VP*' " (i.e., a sentence consists of a noun phrase followed by a verb phrase). Similar interpretations can be made for the other rules.

A *derivation* is the sequence of rewrite rules used to obtain a particular sentence. It can be represented with a diagram. Figure 2 shows a diagram for the derivation of the phrase "the man hit a ball." This diagram conveys less information than the derivation itself because the order in which the rewrite rules have been applied does not appear. This diagram retains only the information that is necessary to determine the phrase structure from the derivation. As such, it clearly shows the hierarchical syntactic structure of the sentence and its constituent analysis.

In this article, we will only consider a particular type of phrase structure grammar called a *context-free grammar*, for which the rewrite rules R are a subset of $N \to (N \cup \Sigma)^*$.

Figure 2. Diagram for the derivation of the phrase "the man hit a ball" with the grammar from Figure 1 (adapted from Chomsky 1957, p. 27).

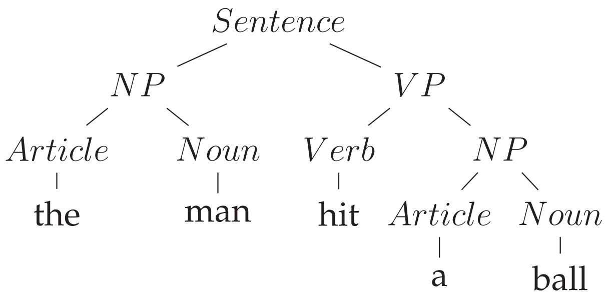


Figure 3. Rhythm Changes: chord progression of the tune "I Got Rhythm" by George Gershwin. The chords are shown as degrees with respect to the main tonality of the tune using jazz notation (dashes indicate minor chords).

$\boxed{A}$	I VI–	II– V^7	I VI–	II– V^7
	I I^7	IV IV–	I VI–	II– V^7
$\boxed{A}$	I VI–	II– V^7	I VI–	II– V^7
	I I^7	IV IV–	I V^7	I
$\boxed{B}$	III^7	III^7	VI^7	VI^7
	II^7	II^7	V^7	V^7
$\boxed{A}$	I VI–	II– V^7	I VI–	II– V^7
	I I^7	IV IV–	I V^7	I

A Phrase Structure Grammar for Rhythm Changes

To show how a phrase structure grammar can be used to create multilevel scenarios, we decided to create such a grammar for the Rhythm Changes. To do so, we took a set of jazz compositions based on Rhythm Changes from the *Charlie Parker Omnibook* (Parker and Aebersold 1978). We then analyzed this corpus with Pascal Mabit, a professional jazz musician and teacher. We took advantage of earlier work, in which we had created MusicXML representations of the compositions in the *Omnibook* (cf. Déguernel, Vincent, and Assayag 2016, the MusicXML files are available at members.loria.fr/evincent/files/omnibook). Our corpus consists of the chord progressions on the theme and the transcribed voicings played during Parker's solos of the ten tunes based on Rhythm Changes found in the *Omnibook*, leading to 26 variations. The tunes from the *Omnibook* based on Rhythm Changes are "An Oscar for Treadwell," "Anthropology," "Celerity," "Chasing the Bird," "Kim" (two versions), "Moose the Mooche," "Passport," "Red Cross," "Steeplechase," and "Thriving from a Riff." The chords are each two beats long and are annotated with their harmonic function relative to the tonic of the tune's main key.

Rhythm Changes tunes follow the 32-bar chord progression displayed in Figure 3.

The global form of the Rhythm Changes is *AABA*, with a *B* section (called the "bridge") contrasting with the *A* sections. The *A* sections are eight bars long, with fast-changing chords (usually two chords per bar) staying close to the initial tonality. These sections consist of:

1. A series of two *turnarounds*, one on bars 1 and 2 and the other on bars 3 and 4. The turnaround is a two-bar tonal function based on an arc of the circle of fifths. Its most characteristic form is `I VI– II– V` (dashes indicate minor chords) but it has many variations based on common chord substitutions (for instance, I I II– V^7, III– VI^7 II– V^7, I VI^7 II– ♭II, etc.). The first turnaround always starts on the first degree of the key, to highlight the tonality at the beginning of the section. We denote the set of all turnarounds as τ, and with τ_{I} we denote the subset of turnarounds starting on the first degree ($\tau_{\mathrm{I}} \subset \tau$).
2. A temporary tonicization of the subdominant (degree IV) on bars 5 and 6. Its characteristic form is I I^7 IV IV– but it also has many variations based on the usual substitutions (for instance, V– I^7 IV ♯IVo, I I^7 IV^7 ♭VII^7, etc.). We denote the set of all variations of this harmonic function by σ.
3. A final turnaround on bars 7 and 8. Except for the first *A*, this turnaround can be replaced with a *loop cadence* to either end on a tonic chord or to anticipate the following chord function. For instance, a loop cadence can be II– V^7 I I. We denote the set of all variations of loop cadences by ω.

Figure 4. Phrase structure grammar for the Rhythm Changes.

1: $RhythmChanges \rightarrow A_1 \ \ A \ \ B \ \ A$
2: $A_1 \rightarrow \tau_{\mathtt{I}} \ \ \tau \ \ \sigma \ \ \tau$
3: $A_2 \rightarrow \tau_{\mathtt{I}} \ \ \tau \ \ \sigma \ \ \omega$
4: $A \rightarrow A_1 \mid A_2$
5: $B \rightarrow \delta_{\mathtt{III}} \ \ \delta_{\mathtt{VI}} \ \ \delta_{\mathtt{II}} \ \ \delta_{\mathtt{V}}$
6: $\tau_{\mathtt{I}}, \tau, \sigma, \omega, \delta_{\mathtt{III}}, \delta_{\mathtt{VI}}, \delta_{\mathtt{II}}, \delta_{\mathtt{V}}$ are sets of harmonic functions that depend on the corpus.

The *B* section is eight bars long, with a slower chord progression (usually, each chord is played over two bars) based on dominant-seventh chords following the circle of fifths ($\mathtt{III}^7$ $\mathtt{VI}^7$ $\mathtt{II}^7$ $\mathtt{V}^7$), giving a sensation of key shifting. Improvisers usually underline these progressions by focusing on "guide" notes (i.e., the third and the seventh) of these chords. Once again, the usual substitutions can be applied, for instance, the two bars of $\mathtt{V}^7$ can be replaced by a bar of $\mathtt{II}$– followed by a bar of either $\mathtt{V}^7$ or of $\flat\mathtt{II}^7$). With $\delta_{\mathtt{X}}$ we denote the tonal function of dominant on degree $\mathtt{X}$.

The Rhythm Changes is an interesting case study for our application because there are so many variations of this chord progression. This is of considerable interest for musicians who can change the progression on the fly, using different substitutions during the improvisation, as long as the global form and the different tonal functions are played. The chord progression can be different for each occurence of a chorus. Using a phrase structure grammar to analyze and generate Rhythm Changes therefore seems appropriate. Considering chords, tonal functions, and sections as the constituents, we can create a phrase structure grammar representing the hierarchical structure of the Rhythm Changes in which the chords are the terminal symbols. Figure 4 shows the grammar for Rhythm Changes created by Mabit, which will be used as a ground truth in the next section. Figure 5 gives the diagram for the derivation of this grammar on one instance of Rhythm Changes from the *Omnibook* entitled "Celerity."

Learning Multilevel Structures

In this section we present a method of inducing an unsupervised grammar to automatically learn a multilevel structure from a training corpus. In our case, the training corpus is composed of several scenarios (here, chord sequences) that are expected to follow the same form. The idea is to infer the hierarchical structure of the scenarios, without prior knowledge of structure and without music theoretical input, by adapting the sequence selection method by Zitouni et al. (2000). The method was originally introduced for text analysis, we adapted it to the specific problems encountered in musical sequences.

Inducing a Context-Free Grammar with Sequence Selection

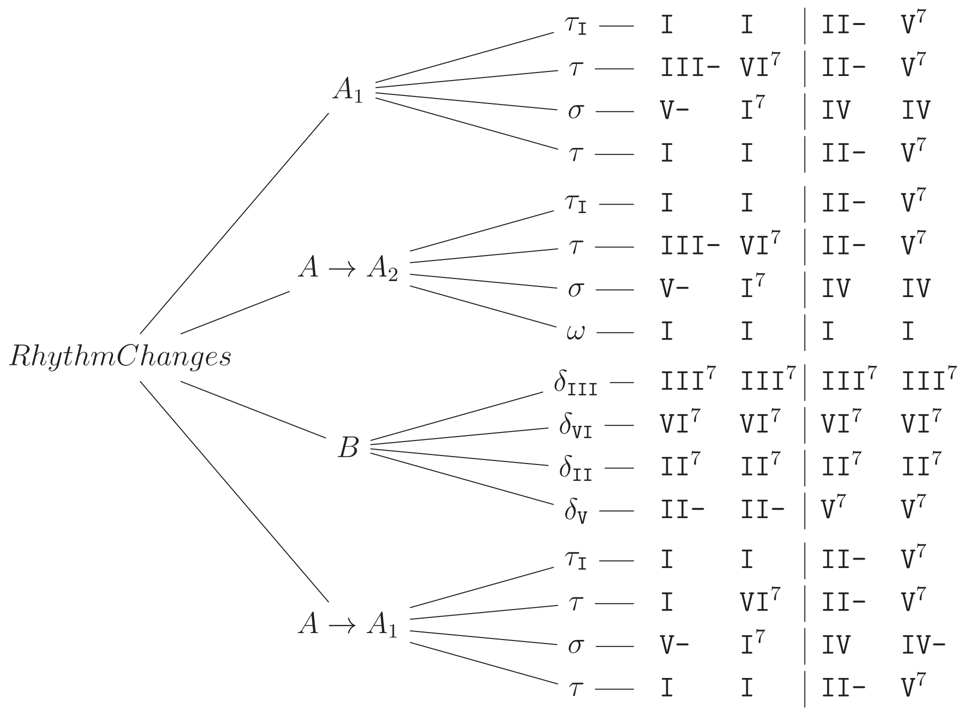
The main idea of this method is to simultaneously and iteratively segment all sequences in the corpus by unifying pairs of symbols sharing the highest mutual information. For two consecutive symbols *a* and *b*, the mutual information $J(a, b)$ is defined by

$$J(a, b) = \log \frac{\mathtt{count}(a\ b) N}{\mathtt{count}(a)\mathtt{count}(b)},$$

where $\mathtt{count}$ is the number of occurrences in the corpus ($\mathtt{count}(a\ b)$ is the number of occurences where *a* appears right before *b* in the corpus), and *N* is the corpus size. A high value of mutual information means that the symbols *a* and *b* occur as a sequence much more frequently than could be expected from pure chance. The iterative unification of symbols sharing the highest mutual information therefore leads to a hierarchical structure.

We applied Zitouni's original method on the *Omnibook* Rhythm Changes subcorpus. We obtained mixed results and identified two main limitations to this method. First, iterative unification leads

Figure 5. Diagram for the derivation of the Rhythm Changes on the tune "Celerity" from the Charlie Parker Omnibook. *Each chord lasts two beats.* A → A$_2$ *and* A → A$_1$ *denote the use of Rule 4 from the grammar shown in Figure 4.*



RhythmChanges
A_1
$A \to A_2$
B
$A \to A_1$
$\tau_{\mathtt{I}}$ — I I | II- V^7
τ — III- VI^7 | II- V^7
σ — V- I^7 | IV IV
τ — I I | II- V^7
$\tau_{\mathtt{I}}$ — I I | II- V^7
τ — III- VI^7 | II- V^7
σ — V- I^7 | IV IV
ω — I I | I I
$\delta_{\mathtt{III}}$ — III^7 III^7 | III^7 III^7
$\delta_{\mathtt{VI}}$ — VI^7 VI^7 | VI^7 VI^7
$\delta_{\mathtt{II}}$ — II^7 II^7 | II^7 II^7
$\delta_{\mathtt{V}}$ — II- II- | V^7 V^7
$\tau_{\mathtt{I}}$ — I I | II- V^7
τ — I VI^7 | II- V^7
σ — V- I^7 | IV IV-
τ — I I | II- V^7

to a clumping phenomenon around rare symbols appearing only in a single context. This is because of the limited amount of data. If a symbol appears in one context only, it will share a large amount of mutual information with its neighbors, and so it will be unified with one of them, creating a new symbol appearing in this one context only. This creates a vertical hierarchical structure centered on this symbol. Second, no relation other than the identity exists between symbols. For instance, all the variants of turnaround are considered strictly different (I VI– II– V^7 is different from III– VI^7 II– V^7 from a symbolic point of view, despite the fact that these sequences have the same tonal function). This results in poor high-level structures since, at higher levels, we obtain a set of symbols that is too large for too little data.

To deal with the first problem, we introduced the notion of symbol length, corresponding to the length, in beats, of the chords (or chord sequences) they represent. For a symbol a, we denote its length as $l(a)$. To avoid the clumping problem around rare symbols, we want to prioritize the unification of pairs of symbols of short length. To this end, we normalize the mutual information by the length of the symbols:

$$\widetilde{J}(a,b) = \frac{1}{l(a)+l(b)} \log \frac{\mathtt{count}(a\ b)N}{\mathtt{count}(a)\mathtt{count}(b)}.$$

To deal with the second problem, when creating a new symbol (by merging two other symbols), we check whether it is equivalent to another existing symbol of the same length using the sequential structure. We want to consider two symbols as equivalent in terms of mutual information, if they have the same length and have similar neighborhoods. Two symbols a and b are considered equivalent if

$$\Psi(a,b) = \frac{1}{K}\sum_{u,v}(J(u,a) - J(u,b))^2 + (J(a,v) - J(b,v))^2 \leq \xi,$$

where

K is the size of the vocabulary,

Figure 6. Algorithm for grammar induction from a corpus of scenarios.

```
   **Input :** Corpus of scenarios.
   **Output :** Set of rewrite rules.
1: **Repeat**
2:     Find $a$ and $b$ such that $\tilde{J}(a, b) = \max_{x,y} \tilde{J}(x, y)$.
3:     Create the rewrite rule $X_{ab} \rightarrow a\ \ b$.
4:     $l(X_{ab}) \leftarrow l(a) + l(b)$.
5:     Replace all occurrences of $a\ \ b$ with $X_{ab}$ in the corpus.
6:     **if** $\exists$ a symbol $Y$ such that $l(Y) = l(X_{ab})$ and $\Psi(Y, X_{ab}) < \xi$ **then**    $\triangleright$ If several
   symbols $y$ respect these conditions, we take $Y$ such that $\Psi(Y, X_{ab}) = \min_{y} \Psi(y, X_{ab})$.
7:         Create the rewrite rule $Y \rightarrow X_{ab}$.
8:         Replace all occurrences of $X_{ab}$ with $Y$ in the corpus.
9:     **end if**
```

u is the list of symbols on the left of *a* and *b*,
v is the list of symbols on the right of *a* and *b*, and
ξ is a threshold value.

The threshold value is attuned here manually for the corpus following the knowledge of the corpus acquired by its analysis by a professional musician.

This function computes the distance, in terms of mutual information between *a* and *b* and every neighbor of *a* or *b* (either on the left or on the right). Thus, $\Psi(a, b)$ is the sum of all these distances, normalized by the size of the vocabulary. The lower $\Psi(a, b)$ is, the more *a* and *b* are used in similar contexts. If *a* and *b* are used in sufficiently similar contexts, the two symbols are unified. Although we set the value manually here, the threshold ξ could also be set using machine learning on a validation corpus, either in a supervised manner, based on a ground truth, or in an unsupervised manner, using characteristics of the inferred grammar and derivation (minimum depth, minimum number of symbols, etc.).

In Figure 6 we show an algorithm for grammar induction from a corpus of scenarios.

Evaluation of the Induced Grammar

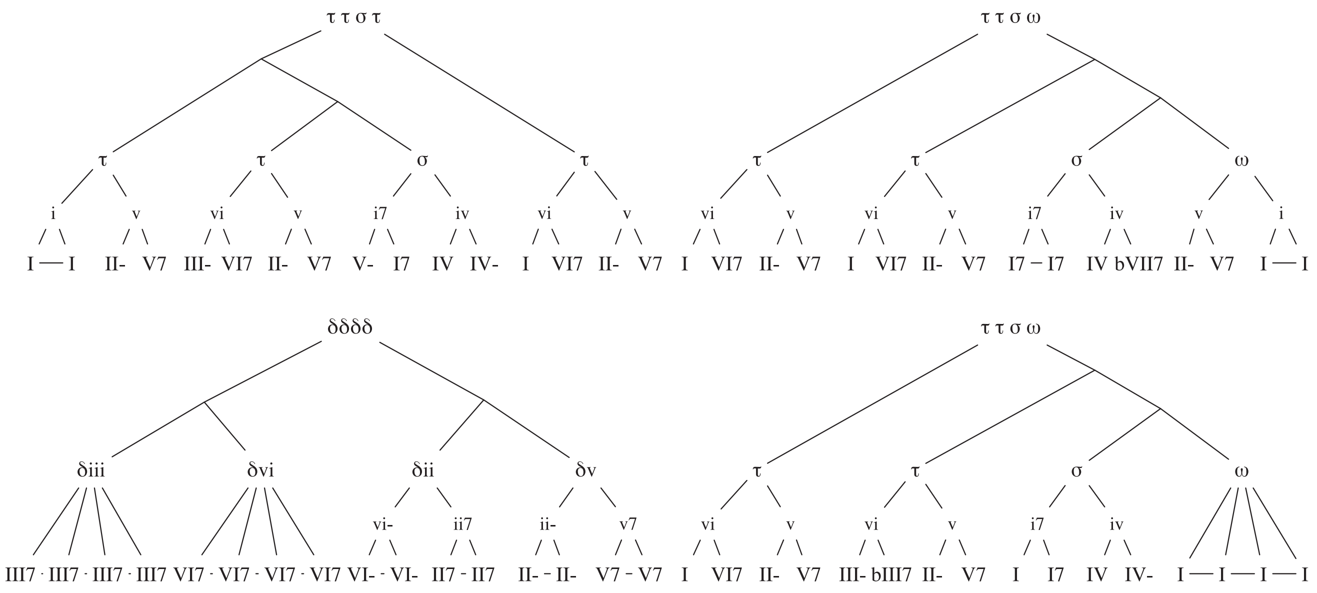
We applied the algorithm in Figure 6 to the subcorpus of Rhythm Changes from the *Omnibook*. Figure 7 shows the hierarchical analysis obtained on one example from this subcorpus. The automatic analyses of the 26 variations of Rhythm Changes from the corpus have all been studied and validated by Mabit.

First, the different tonal functions (turnaround, subdominant tonicization, etc.) have been properly segmented and the structural organization is correct on all desired levels (chords, tonal functions, and sections), for all variations of the Rhythm Changes. Second, the different variations of a given tonal function have been correctly identified as equivalent. For instance, `I VI`^{7} `II– V`^{7} and `III– ♭III`^{7} `II– V`^{7} are considered equivalent, as are `V– I`^{7} `IV IV–` and `I`^{7} `I`^{7} `IV ♭VII`^{7}. This analysis also reveals another level of organization between the chord level and the functional level that does not appear in the ground-truth grammar but which was deemed semantically meaningful by the musician.

This analysis is less accurate than the ground truth on a couple of points, however. First, the different types of *A* section (A_1 and A_2 from the grammar in Figure 4) are consistently considered strictly different. Second, all variants of turnaround are identified as equivalent, masking the fact that the first turnaround of an *A* section should start on the first degree (this specific variant of turnaround is called τ_I in the grammar in Figure 4).

Overall, the results of this method are promising: The multilevel structures we obtained are very close to those found by Mabit. It must be noted that we

Figure 7. Hierarchical analysis of Rhythm Changes obtained with the algorithm shown in Figure 6. The different sections are separated into four subtrees, and the last level of organization is not shown. Note that the lowercase roman numerals in the first level of analysis above chords represents here two-chord long harmonic functions not minor chords.


τ τ σ τ
τ τ σ ω
δδδδ
τ τ σ ω
I — I II- V7 III- VI7 II- V7 V- I7 IV IV- I VI7 II- V7
I VI7 II- V7 I VI7 II- V7 I7 – I7 IV bVII7 II- V7 I — I
III7 · III7 · III7 · III7 VI7 · VI7 · VI7 · VI7 VI- - VI- II7 - II7 II- - II- V7 - V7
I VI7 II- V7 III- bIII7 II- V7 I I7 IV IV- I — I — I — I

used a corpus of scenarios of the same form; this method could therefore be applied immediately to other corpora representative of another form or style, such as the blues. It would be interesting to compare this method with other grammar induction methods that are applicable to corpora of limited size. It would also be interesting to apply this method to larger and more-varied corpora such as the *Real Book*, which involves different forms, and to other genres (classical music, pop, Georgian chant, etc.). Early experiments on the *Real Book* indicate that this method may be able to detect the most common harmonic functions and their substitutions, and to recognize harmonic similarities across different forms, similar to work by de Haas et al. (2009).

Using a Multilevel Structure for Guided Improvisation

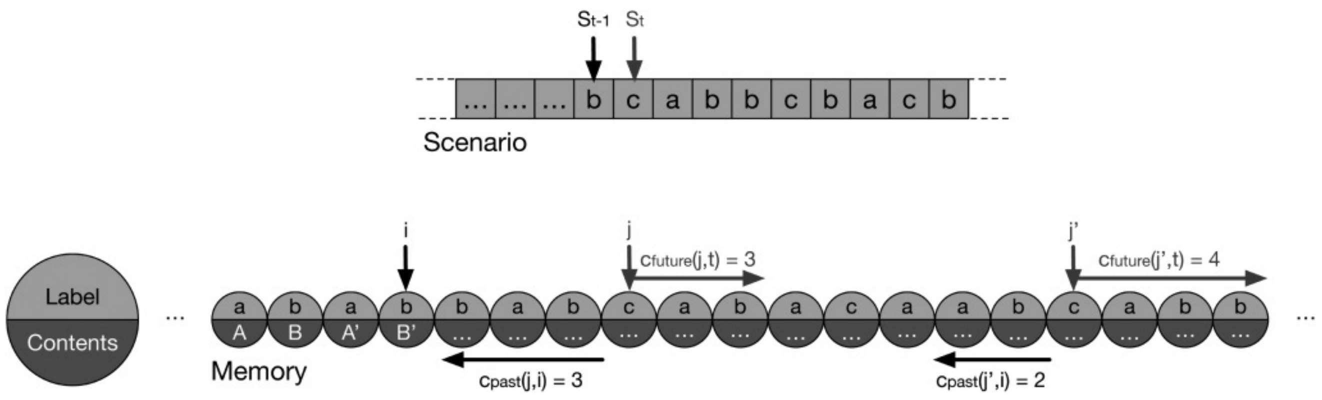
In this section, we first describe a method for guiding an automatic improvisation system using a scenario, following work by Nika, Chemillier, and Assayag. We then propose a new method using a multilevel scenario to enrich the improvisation in a more flexible manner. Finally, the models are compared in listening sessions and in interviews with professional musicians.

Improvisation on a Temporal Scenario

Our method is based on the work by Nika, Chemillier, and Assayag (2017a), who introduce a temporal scenario, modelled as a sequence of labels, to guide an improvisation. During training, the system's memory is constructed as a sequence of contents from the improvised dimension, organized using a factor oracle (Allauzen, Crochemore, and Raffinot 1999) from the scenario labels. In the case of jazz, the labels can be the chords of a chord progression and the contents can be the musical notes of the melody played by an improviser. During the generation process, for a given scenario, we try to ensure consistency both with the future of the scenario by using anticipatory behaviors, and with the memory's past by using the properties of the factor oracle (cf. Assayag et al. 2006). Musical contents from the chosen states are then played to generate the improvisation.

The generation process is divided into two successive steps: an *anticipation step* followed by a *navigation step* (for a more detailed description of

Figure 8. Creation of the set of states in memory sharing a common future with the current position in the scenario t *and a common past with the current event in memory* i_{t-1} *for the anticipation step, following Nika, Chemillier, and Assayag (2017a).*



the algorithm, see Nika, Chemillier, and Assayag 2017a). Let us denote the scenario by $S = S_1 \ldots S_s$, the current position in the scenario is t, and let us consider a memory represented by a factor oracle with states $0 \ldots m$ and labels $\Lambda_0 \ldots \Lambda_m$.

The anticipation step consists of looking for events in memory sharing a common future with the current position in the scenario, while ensuring consistency with the memory's past. This is achieved by indexing the prefixes of the suffixes of the current position in the scenario in memory. First, we build the set of states in memory sharing a common future with the current position in the scenario $S_t \ldots S_s$:

$$\text{Future}(t) = \{\, j \in [0 \ldots m] \mid \exists c_{\text{future}} > 0,\ \Lambda_j \ldots \Lambda_{j+c_{\text{future}}-1} \in \text{Pref}(S_t \ldots S_s)\} ,$$

where c_{future} is the length of the prefix. Then, we build the set of states in memory sharing a common past with the current state i in memory,

$$\text{Past}(i) = \{\, j \in [0 \ldots m] \mid \exists c_{\text{past}} \in [1, j],\ \Lambda_{j-c_{\text{past}}+1} \ldots \Lambda_j \in \text{Suff}(0 \ldots i)\} .$$

This set can be constructed using the properties of the factor oracle's suffix links.

For the anticipation step, we look for the positions $j \in [0 \ldots m]$ in memory such that

$$j \in \text{Future}(t) \quad \text{and} \quad j - 1 \in \text{Past}(i).$$

Figure 8 shows an anticipation step.

The navigation step consists of looking for events in memory sharing a common context with the current position in the scenario, while conforming to the scenario. We are looking for positions $j \in [0 \ldots m]$ in memory such that

$$\Lambda_j = S_t \quad \text{and} \quad j - 1 \in \text{Past}(i).$$

This step enables the system to follow nonlinear paths in memory to create new musical phrases, thereby generating more local variations, in a fashion similar to that of Assayag et al. (2006).

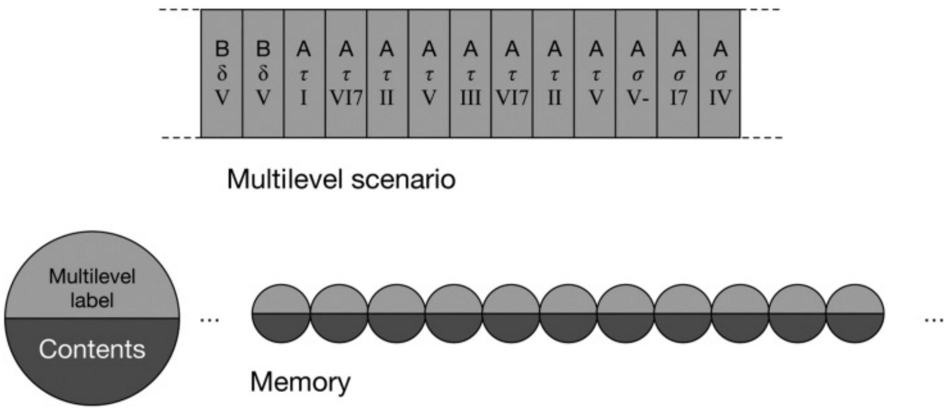
In practice, minimum and maximum values for c_{future} and c_{past} are chosen to avoid the use of fragments in memory that are either too short or too long while generating an improvisation.

Using Multilevel Information

Now we want to take a multilevel scenario into account in the generation process. In this case, the scenario is no longer merely a sequence of symbols, but a sequence of lists of symbols corresponding to each level. We call these sequences *multilevel labels*. Memory is therefore built from multilevel labels corresponding to the multilevel scenario. Figure 9 shows an example of multilevel scenario with three levels of temporal organization.

We adapted the generation method presented in the previous section to account for the information brought by the multilevel aspect in both the anticipation and navigation steps. For each step, we want to extend the possible positions in memory

Figure 9. Example of a multilevel scenario and a multilevel memory.



to states that share equivalent multilevel labels to those of the scenario. That is to say, labels sharing common information on one or more levels but possibly different information on other levels. For instance, in jazz we could accept positions in memory with a different chord label than the one in the scenario, as long as they share the same tonal function and the same section. This way, it would be possible to generate improvisations on a scenario with chords that do not exist in memory but share the same tonal function as a known chord. We would also like to prioritize positions in memory sharing a strong common information with the future of the scenario and the memory's past. Thus, the generation process would better account for the hierarchical structure of the scenario to guide the improvisation.

This process can be explained as follows. Let us denote the given multilevel scenario as $S = S_1 \ldots S_s$ (in our case, $S_1, \ldots, S_s$ are chords) of length s with Q levels, with $\forall p \in [1 \ldots s], S_p = \{S_p^1 \ldots S_p^Q\}$. Let t be the current position in the scenario and let us consider a memory built with a factor oracle with states $0 \ldots m$ and multilevel labels $\Lambda_0 \ldots \Lambda_m$, with $\forall p \in [0 \ldots m], \Lambda_p = \{\Lambda_p^1 \ldots \Lambda_p^Q\}$. We consider that two multilevel symbols u and v are equivalent (in the general sense rather than in the mathematical sense of an equivalence relation), and we write $u \simeq v$ if $\exists q; u^q = v^q$. For instance, with

$$u = \left\{\begin{matrix} A \\ \tau \\ \mathtt{I} \end{matrix}\right\}, \text{and } v = \left\{\begin{matrix} A \\ \tau \\ \mathtt{III-} \end{matrix}\right\},$$

we have $u \simeq v$. By extension, we consider that a sequence of multilevel symbols $u = u_1 \ldots u_n$ with $\forall p \in [1 \ldots n], u_p = \{u_p^1 \ldots u_p^Q\}$ is an equivalent prefix of a sequence of multilevel symbols $v = v_1 \ldots v_m$ with $\forall p \in [1 \ldots m], v_p = \{v_p^1 \ldots v_p^Q\}$, if

$$u = u_1 \ldots u_n \in \text{Pref\_eq}(v = v_1 \ldots v_m) \iff \forall p \in [1 \ldots n], u_p \simeq v_p.$$

For instance,

$$u = \left\{\begin{matrix} A \\ \omega \\ \mathtt{V7} \end{matrix}\right\}\left\{\begin{matrix} A \\ \omega \\ \mathtt{I} \end{matrix}\right\} \in \text{Pref\_eq}\left(v = \left\{\begin{matrix} A \\ \omega \\ \mathtt{I} \end{matrix}\right\}\left\{\begin{matrix} A \\ \omega \\ \mathtt{I} \end{matrix}\right\}\left\{\begin{matrix} B \\ \delta_{III} \\ \mathtt{III7} \end{matrix}\right\}\left\{\begin{matrix} B \\ \delta_{III} \\ \mathtt{III7} \end{matrix}\right\}\right).$$

Similarly, we consider that a sequence of multilevel symbols $u = u_1 \ldots u_n$ with $\forall p \in [1 \ldots n], u_p = \{u_p^1 \ldots u_p^Q\}$ is an equivalent suffix of a sequence of multilevel symbols $v = v_1 \ldots v_m$ with $\forall p \in [1 \ldots m], v_p = \{v_p^1 \ldots v_p^Q\}$, if

$$u = u_1 \ldots u_n \in \text{Suff\_eq}(v = v_1 \ldots v_m) \iff \forall p \in [1 \ldots n], u_p \simeq v_{p+m-n}.$$

Both generation steps are modified to take the multilevel information into account with equivalent labels as follows.

First, for the anticipation step, we replace the sets Future(t) and Past(i) with the sets Future_eq(t) and

Past_eq(i), respectively:

$$\begin{aligned}\text{Future\_eq}(t) &= \{\, j \in [0\ldots m] \mid \exists c_{\text{future}} > 0,\\ &\qquad \Lambda_j \ldots \Lambda_{j+c_{\text{future}}-1} \in \text{Pref\_eq}(S_t \ldots S_s)\},\\ \text{Past\_eq}(i) &= \{\, j \in [0\ldots m] \mid \exists c_{\text{past}} \in [1, j],\\ &\qquad \Lambda_{j-c_{\text{past}}+1} \ldots \Lambda_j \in \text{Suff\_eq}(0\ldots i)\}.\end{aligned}$$

We then consider the positions $j \in [0\ldots m]$ in memory such that

$$j \in \text{Future\_eq}(t) \quad \text{and} \quad j-1 \in \text{Past\_eq}(i).$$

Then, for the navigation step, we similarly extend the possibilities with equivalent labels. We look for positions $j \in [0\ldots m]$ in memory such that

$$\Lambda_j \doteq S_t \quad \text{and} \quad j-1 \in \text{Past\_eq}(i).$$

For each step, to prioritize the positions j sharing the most common information with the future of the scenario and with the memory's past, a similarity score between labels is created. For each time level q, the user can choose, a priori, a weight W_q such that $\sum_q W_q = 1$. For instance, in jazz we could consider that the level of tonal function is more important than the chord level or the section level.

The similarity between two multilevel labels Λ_i and Λ_j is defined by

$$\varsigma(\Lambda_i, \Lambda_j) = \sum_q W_q \delta_{\Lambda_i^q, \Lambda_j^q},$$

where δ is the Kronecker symbol.

For instance, if we consider the case with section level, tonal-function level, and chord level, and assign the weights of 0.1, 0.6, and 0.3, respectively, we get

$$\varsigma\left(\begin{Bmatrix} A \\ \tau \\ \texttt{I} \end{Bmatrix}, \begin{Bmatrix} A \\ \tau \\ \texttt{III-} \end{Bmatrix}\right) = 0.1 + 0.6 = 0.7 \text{ and}$$

$$\varsigma\left(\begin{Bmatrix} B \\ \delta_{VI} \\ \texttt{III-} \end{Bmatrix}, \begin{Bmatrix} A \\ \tau \\ \texttt{III-} \end{Bmatrix}\right) = 0.3.$$

Each element j considered in the anticipation step can be given the score

$$\sum_{k=0}^{c_{\text{future}}(j,t)} \varsigma(\Lambda_{j+k}, S_{t+k}) + \sum_{k=0}^{c_{\text{past}}(j,i)} \varsigma(\Lambda_{j-k}, \Lambda_{i-k})$$

and each element j considered in the navigation step can be given the score

$$\varsigma(\Lambda_j, S_t) + \sum_{k=0}^{c_{\text{past}}(j,i)} \varsigma(\Lambda_{j-k}, \Lambda_{i-k}),$$

where $c_{\text{future}}(j,t)$ and $c_{\text{past}}(j,i)$ are, respectively, the maximum length of c_{future} associated with $j \in \text{Future\_eq}(t)$ and the maximum weight of c_{past} associated with $j \in \text{Past\_eq}(i)$.

With these scores, several strategies can be applied to prioritize elements sharing more common information with the scenario's future and with the memory's past. The event with the highest score can be chosen, or we can only consider events with a higher score than a given threshold. In the following experiment, we normalized the scores with the sum of the scores of all possible events and then selected an event in a random fashion following the obtained pseudoprobabilities.

Evaluation and Musicians' Feedback

To evaluate the generated improvisations, we conducted listening sessions and interviews with three professional jazz musicians: Louis Bourhis, a jazz bassist who graduated from the Haute École de Musique de Lausanne; Joël Gauvrit, a jazz and classical music pianist and teacher who graduated from the Conservatoire National Supérieur de Musique et de Danse de Lyon; and Pascal Mabit, a jazz saxophonist and teacher who graduated from the Conservatoire National Supérieur de Musique et de Danse de Paris.

We focused on a corpus of tunes based on the Rhythm Changes, taken from the *Charlie Parker Omnibook*. For these interviews, multilevel scenarios of Rhythm Changes were generated using the ground-truth grammar created by Mabit. Three levels of organization were considered: chords, tonal functions, and sections. For each improvisation generated, the contents in memory are based on a theme and an improvisation from our corpus. We used the improvisations on "Thriving from a Riff," "An Oscar for Treadwell," and "Anthropology." Memory is divided per beat to more easily match the labels of memory to the labels of the scenario.

In these scenarios, we generated improvisations using two approaches. First, we used an approach following Nika, Chemillier, and Assayag (2017a), hereafter called the *base method*. The anticipation of the scenario and the navigation in memory use only the chord level of the scenario (other levels are not taken into account); when a chord in the scenario does not appear in memory, key transposition is applied. We then compared these with our multilevel approach. The anticipation of the scenario and the navigation in memory use all the multilevel information of the scenario.

Examples of improvisations generated with both methods are available at https://www.mitpressjournals.org/doi/suppl/10.1162/COMJ_a_00521. Each musician listened to a dozen improvisations for each method. Each time an improvisation was generated with both methods for a given chord progression generated by the phrase structure grammar.

All musicians immediately noticed a clear difference between the base method and the multilevel approach. The main improvement noticed by the musicians was the global construction of the improvisation on the whole chord progression. Improvisations generated with the multilevel method were deemed more realistic thanks to a better account of harmonic structures. Gauvrit commented:

> The difference is impressive. Here we feel like it's constructing. This is a great chorus. If I have a student who does that, I'm happy. Even rhythmically, the way it works with the harmonic structure . . . I do hear that. With the other, no, not at all.

Mabit elaborated this point, noticing that, with multilevel method, the improvisation is driven around the functions and the sections, creating phrases (or sequences of phrases) with a longer-term consistency:

> It's more realistic. You can really feel a will to develop material in the long term, using harmonic functions and all that. Not like before where it played more chord by chord. . . . For instance, at the beginning of a section, it starts with fewer notes and then it builds upon it. It's much more human in this sense.

The melodies generated with the multilevel method were also deemed less disjointed. This can be explained by the fact that this method opens up more possibilities in navigation, enabling the system to make fewer jumps in memory during complex local harmonic progressions. Bourhis said:

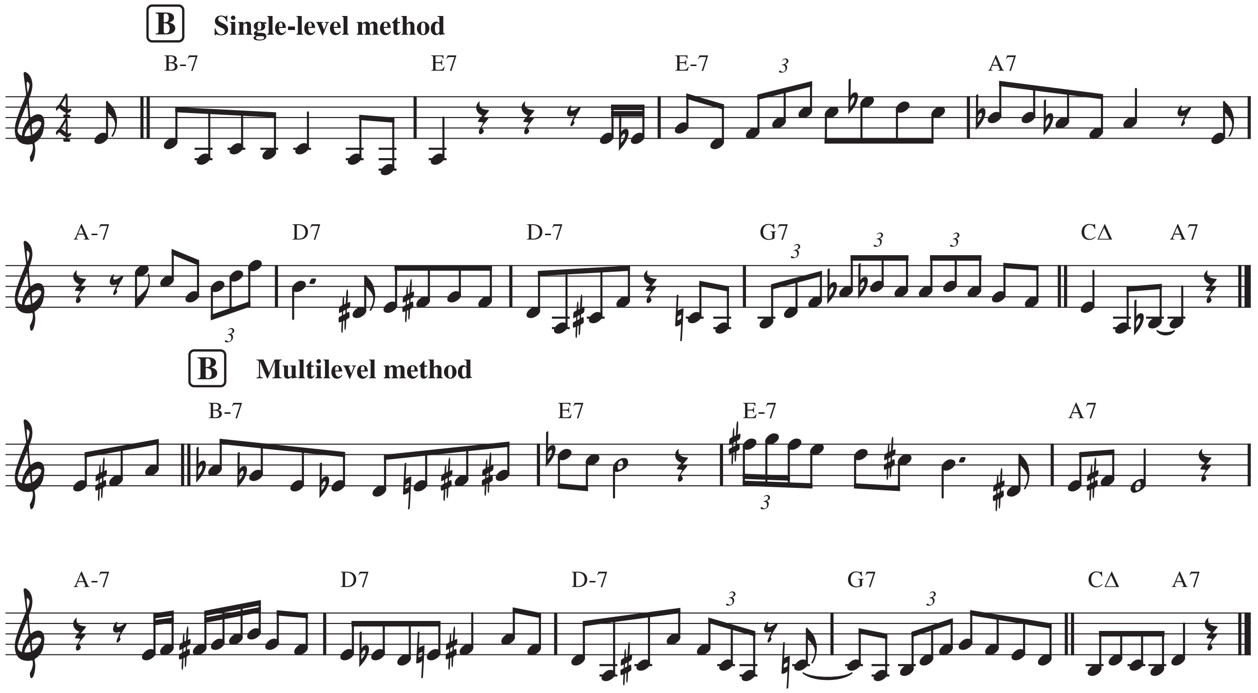
> It's true that there is this thing, at the very beginning we are afraid that the computer will take several fragments and put them side by side without really linking them. And now, compared to the other version, I think this is a huge improvement regarding the authenticity, the realism of Charlie Parker's language.

When the scenario involved chords that did not appear in memory, the freedom given by the multilevel approach to play on other chords sharing the same tonal function and section also helped to create improvisations that were less fragmented, giving a better sense of consistency and smoothness. For instance, this can be heard in the improvisations generated with the memory from "An Oscar for Treadwell," where the chords in the bridge are $\texttt{III}^7$ $\texttt{III}^7$ $\texttt{VI}^7$ $\texttt{VI}^7$ $\texttt{II}^7$ $\texttt{II}^7$ $\texttt{V}^7$ $\texttt{V}^7$, while the scenario consisted of the subdominant and dominant chords $\texttt{VII-}^7$ $\texttt{III}^7$ $\texttt{III-}^7$ $\texttt{VI}^7$ $\texttt{VI-}^7$ $\texttt{II}^7$ $\texttt{II-}^7$ $\texttt{V}^7$. According to Bourhis:

> It's just that, with the first version, it doesn't know how to deal with the subdominant. With the [multilevel] method, it goes less in the details but it makes it more consistent on a larger scale.

Figure 10 shows examples of improvisations on the *B* section using either the single-level approach or the multilevel approach, with both using the same memory from "An Oscar for Treadwell." A similar

Figure 10. Comparison of improvisations on the B section from a version of Rhythm Changes, using the single-level and multilevel scenarios from the memory of "An Oscar for Treadwell."



remark was made by Mabit: In some *A* sections, the subdominant function had a diminished chord in it, which is a chord class that does not exist in the memory of "Anthropology." Under this condition the base method could not generate an improvisation, but we could do it with the multilevel method and obtained convincing results. Figure 11 shows an example of this situation. Moreover, being able to play chords from the scenario that are different from the ones in memory brings a kind of creativity, whereby the guide notes and the extensions give new colors to the improvisation.

More surprisingly, another improvement noticed by the musicians is that the improvisations generated with the multilevel approach have better rhythmic consistency. The development of the improvisation from the points of view of density and of rhythmic flow was deemed more realistic. Gauvrit compared both methods to his jazz students:

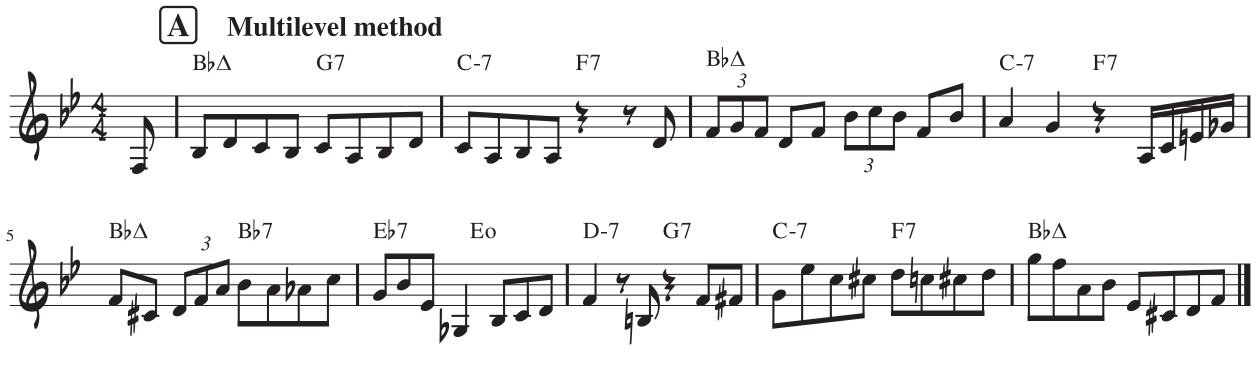
> The construction makes sense. You feel like there is a person thinking behind it. Absolutely, it's impressive. . . . It's really interesting from a teaching point of view. I feel like I'm seeing different stages of my students' understanding, you see, of what's written, analytically. . . . And that spontaneous insight is funny because, if you think about it, that consciousness of the harmonic structures brings rhythmical things that are much more natural.

Overall, the multilevel method received little criticism from the musicians. Bourhis still noticed some unsettling jumps in memory (for instance, octave leaps that were not realistic), but they were much less frequent than with the base method:

> Then you still have some weird things, but everything is exaggerated with the first method when it's disjointed, octave leaps mainly.

All in all, the musicians' feedback on this experiment is gratifying. The understanding of the multilevel structure of a harmonic progression enables the generation of improvisations that are more realistic. Over and above being able to improvise more freely and in a fashion that is more varied (even on previously unseen chords), this improves melodic and rhythmic consistency.

Figure 11. Improvisation on the A *section of version of the Rhythm Changes using a diminished chord (E°) in the subdominant function with the memory of "Anthropology."*



The generated improvisations do not "focus on the details to serve a better organization at a larger scale," making it closer to the improvisation process used by human musicians, and enabling a better understanding of harmonic spaces and a better phrasing of melodies.

Following these interviews, another listening session was conducted with Mabit and Bourhis, this time using the grammar generated automatically with sequence selection. Both musicians agreed that the improvisations generated with both grammars were pretty much indistinguishable from each other and shared the same qualities when compared to the base method.

Discussion

We presented a method to analyze and use the multilevel structure of a jazz composition to generate music in the context of guided improvisation with a temporal scenario. First, we introduced the use of a phrase structure grammar to describe the hierarchical structure of a scenario over several levels of organization. We then put forward a method for grammar induction based on probabilistic sequence selection to perform automatic analysis of a scenario to obtain a multilevel scenario. We obtained satisfactory results close to the ground truth we defined in collaboration with a professional musician. Next, we designed new heuristics for guided improvisation based on a multilevel scenario. These heuristics enable the system to take the global form of the tune into account and also enrich the creative potential of the system, making it possible to improvise on previously unforeseen events. This system was evaluated with professional improvisers during listening sessions and in interviews. These musicians concluded that taking into account the hierarchical structure of the scenario led to improvisations that were more realistic, with better melodic and rhythmic consistently on both local and global levels.

A possible extension of this work would be to exploit the generative grammar to create a system in which the terminal symbols of the grammar (e.g., the chords in a chord progression) adapt to what is being improvised with reactive listening. This could be done with the idea of scenario inference (Nika et al. 2017b), for which using multilevel structures could make it possible to perform inference on a larger level. It would also be interesting to test whether considering higher levels of hierarchical analysis (i.e., higher than the scenario itself) could enable the system to generate improvisations with consistency upon several iterations of the same scenario. Finally, it would also be interesting to extend this work to other styles of music and to extend the work on metacomposition of scenarios started by Nika, Chemillier, and Assayag (2017a) to the metacomposition of multilevel scenarios.

Acknowledgments

This work is a joint collaboration between the Université de Lorraine, the French National Centre for Scientific Research (CNRS), Inria, Loria, and the laboratory Sciences et technologies de la musique et du son (STMS Lab), IRCAM, CNRS, Sorbonne Universités. This research was made possible by support from the French National Research Agency, in the framework of the project DYCI2 "Creative Dynamics of Improvised Interaction," ANR-14-CE24-0002-01 (http://repmus.ircam.fr/dyci2/project), and with the support of the Région Lorraine.

References

Allauzen, C., M. Crochemore, and M. Raffinot. 1999. "Factor Oracle: A New Structure for Pattern Matching." In J. Pavelka, G. Tel, and M. Bartosek, eds., *SOFSEM'99: Theory and Practice of Informatics*. Berlin: Springer, pp. 291–306.

Assayag, G., et al. 2006. "OMax Brothers: A Dynamic Topology of Agents for Improvisation Learning." In *Proceedings of the ACM Workshop on Audio and Music Computing for Multimedia*, pp. 125–132.

Bickerman, G., et al. 2010. "Learning to Create Jazz Melodies Using Deep Belief Nets." In *Proceedings of the International Conference on Computational Creativity*, pp. 228–236.

Bigo, L., et al. 2017. "Sketching Sonata Form Structure in Selected Classical String Quartets." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 752–759.

Bimbot, F., et al. 2016. "System and Contrast: A Polymorphous Model of the Inner Organization of Structural Segments within Music Pieces." *Music Perception* 33(5):631–661.

Chomsky, N. 1957. *Syntactic Structures*. Berlin: De Gruyter.

Déguernel, K., E. Vincent, and G. Assayag. 2016. "Using Multidimensional Sequences for Improvisation in the OMax Paradigm." In *Proceedings of the Sound and Music Computing Conference*, pp. 117–122.

Déguernel, K., E. Vincent, and G. Assayag. 2018. "Probabilistic Factor Oracles for Multidimensional Machine Improvisation." *Computer Music Journal* 42(2):52–66.

Déguernel, K., et al. 2017. "Generating Equivalent Chord Progressions to Enrich Guided Improvisation: Application to 'Rhythm Changes'." In *Proceedings of the the Sound and Music Computing Conference*, pp. 399–406.

de Haas, W. B., et al. 2009. "Modeling Harmonic Similarity Using a Generative Grammar of Tonal Harmony." In *Proceedings of the International Conference on Music Information Retrieval*, pp. 549–554.

Donzé, A., et al. 2014. "Machine Improvisation with Formal Specifications." In *Proceedings of the International Computer Music Conference*, pp. 1277–1284.

Dubnov, S., G. Assayag, and R. El-Yaniv. 1998. "Universal Classification Applied to Musical Sequences." In *Proceedings of the International Computer Music Conference*, pp. 332–340.

Gallé, M. 2011. "Searching for Compact Hierarchical Structures in DNA by Means of the Smallest Grammar Problem." PhD dissertation, Université Rennes 1.

Gillick, J., K. Tang, and R. M. Keller. 2010. "Machine Learning of Jazz Grammars." *Computer Music Journal* 34(3):56–66.

Giraud, M., et al. 2015. "Computational Fugue Analysis." *Computer Music Journal* 39(2):77–96.

Guichaoua, C. 2017. "Modèles de compression et critères de complexité pour la description et l'inférence de structure musicale." PhD dissertation, Université Rennes 1.

Hopcroft, J. E., and J. D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Boston, Massachusetts: Addison-Wesley.

Keller, R., et al. 2013. "Automating the Explanation of Jazz Chord Progressions Using Idiomatic Analysis." *Computer Music Journal* 37(4):54–69.

Keller, R. M., et al. 2012. "A Creative Improvisation Companion Based on Idiomatic Harmonic Bricks." In *Proceedings of the International Conference on Computational Creativity*, pp. 155–159.

Nika, J., M. Chemillier, and G. Assayag. 2017a. "ImproteK: Introducing Scenarios into Human–Computer Music Improvisation." *ACM Computers in Entertainment* 4(2):Art. 4.

Nika, J., et al. 2017b. "DYCI2 Agents: Merging the 'Free,' 'Reactive,' and 'Scenario-based' Music Generation Paradigms." In *Proceedings of the International Computer Music Conference*, pp. 227–232.

Pachet, F. 2002. "The Continuator: Musical Interaction with Style." In *Proceedings of the International Computer Music Conference*, pp. 211–218.

Pachet, F., and P. Roy. 2011. "Markov Constraints: Steerable Generation of Markov Sequences." *Constraints* 16(2):148–172.

Papadopoulos, A., P. Roy, and F. Pachet. 2016. "Assisted Lead Sheet Composition Using FlowComposer." In *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, pp. 769–785.

Parker, C., and J. Aebersold. 1978. *Charlie Parker Omnibook*. Los Angeles, California: Alfred Music.

Roy, P., and F. Pachet. 2013. "Enforcing Meter in Finite-Length Markov Sequences." In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 854–861.

Steedman, M. 1996. "The Blues and the Abstract Truth: Music and Mental Models." In J. Oakhill and A. Garnham, eds., *Mental Models in Cognitive Science: Essays in Honour of Phil Johnson-Laird*. Hove, UK: Psychology Press, pp. 305–318.

Zitouni, I., K. Smaili, and J.-P. Haton. 2000. "Beyond the Conventional Statistical Language Models: The Variable-Length Sequences Approach." In *Proceedings of the International Conference on Spoken Language Processing*, pp. 562–565.