Learning of Hierarchical Temporal Structures for Guided Improvisation

Ken Déguernel, Emmanuel Vincent, Jérôme Nika, Gerard Assayag, Kamel Smaïli

Learning of Hierarchical Temporal Structures for Guided Improvisation

Ken Déguernel[1,2] **Emmanuel Vincent**[1] **Jérôme Nika**[2]
ken.deguernel@ircam.fr emmanuel.vincent@inria.fr jerome.nika@ircam.fr
Gérard Assayag[2] **Kamel Smaïli**[1]
gerard.assayag@ircam.fr kamel.smaili@loria.fr

[1]Université de Lorraine, CNRS, Inria, Loria, F-54000 Nancy
[2]STMS Lab — IRCAM, CNRS, Sorbonne Universités

Authors' final version. This article appears in *Computer Music Journal* 43:2.

Abstract

This paper focuses on learning the hierarchical structure of a temporal *scenario* (for instance, a chord progression) to perform automatic improvisation consistently upon several time scales. We first present how to represent a hierarchical structure with a phrase structure grammar. Such a grammar enables us to analyse a scenario upon several levels of organisation creating a *multi-level scenario*. Then, we propose a method to automatically induce this grammar from a corpus based on sequence selection with mutual information. We applied this method on a corpus of *rhythm changes* and obtained multi-level scenarios similar to the analysis performed by a professional musician. We then propose new heuristics to exploit the multi-level structure of a scenario to guide the improvisation with anticipatory behaviour in the factor oracle driven improvisation paradigm. This method ensures consistency of the improvisation regarding the global form and opens up possibilities when playing on chords that do not exist in the memory. This system was evaluated by professional improvisers during listening sessions and received very good feedback.

Introduction

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 organisation. For instance, for 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 organised in several sections (long scale). In the following, we use the word "level" rather than "scale" to describe the temporal organisation in order to avoid confusion with the notion of tonality/modality. Our goal is first to perform an automatic analysis of the hierarchical structure of a chord progression and then use this structure to guide an automatic improvisation system.

Over the years, several generative models have been proposed for machine improvisation

such as statistical sequence models (Dubnov et al. 1998), Markov models (Pachet 2002), deep neural networks (Bickerman et al. 2010) or factor oracles (Assayag et al. 2006; Déguernel et al. 2018).
More recently, research has focused on *guided improvisation* where the improvisation system relies both on a generative model representing the musical style and on a prior knowledge of a sequential structure called *scenario* hereafter. Gillick et al. (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 or 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. *FlowComposer* (Papadopoulos et al. 2016) introduced a system where the user can choose a stylistic corpus and set structural constraints on melody and harmony; the system then generates a melody and a chord progression following the style and the constraints. Donzé et al. (2014) built harmonic and rhythmic constraint automata given a chord progression and a melodic and a rhythmic factor oracle to model the style of a musician. Improvising in this system consists in finding a path in the factor oracles that also satisfies the constraints enforced by the automata. However, while they describe a global structure, those systems only enforce this structure on a local level when generating an improvisation and they do not benefit from anticipatory behaviour.
Keller et al. (2012) proposed to use idiomatic harmonic bricks to guide the improvisation. Chord progressions are analysed and organised in harmonic functions creating a scenario with harmonic brick hierarchies upon which melodic improvisations are generated. While this methods takes higher-level structure into account, it still does not benefit from anticipatory behaviour.
ImproteK (Nika et al. 2017a) introduced anticipatory behaviours 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. However, the scenarios used in *ImproteK* are mere sequences of symbols; the global form is not considered in the generative process, which can lead to a feeling of inconsistency in the long term.

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, counter subject, etc.). A similar method was used for sonata form analysis by Bigo et al. (2017). Bimbot et al. (2016) introduced the *System & Contrast* model where 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. 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 on a local level. Haas et al. (2009) used a context free grammar of tonal harmony to create a hierarchical analysis of chord progressions upon four 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 CYK algorithm.

While these methods are partially based on empirically created grammars, Guichaoua (2017) proposed a method of grammar induction on leaves alone using the minimal grammar problem (Gallé 2011). Unfortunately, this method is purely symbolic and therefore is not robust when encountering variations.

In this article, we propose two contributions:

- First, we propose to automatically infer a phrase structure grammar (Chomsky 1957) to represent the hierarchical temporal structure of a scenario, creating what we call a *multi-level scenario*. We propose a probabilistic grammar induction process for musical sequences based on the sequence selection method of Zitouni et al. (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 modelling problems encountered with local variations of a given motif. Once the grammar has been trained, any sequence of symbols representing a scenario of that form can be reformulated as a multi-level scenario represented by *multi-level labels*. We applied this method on a corpus of *rhythm changes* (a famous jazz chord progression) and compared the generated grammar with a ground truth grammar created in collaboration with a professional musician.
- Second, we propose new heuristics that extend the work by Nika et al. (2017b) on *ImproteK* in order to exploit a multi-level scenario to guide improvisation in the factor oracle driven improvisation paradigm. These heuristics were first introduced in (Déguernel et al. 2017) and consist in performing anticipation of the scenario and navigation in the memory upon several temporal levels by considering *equivalent labels*, i.e. multi-level labels sharing common information with the target scenario on one or more levels. In this article, we extend this preliminary study with a more in-depth formalisation of these heuristics and we conduct an evaluation of our system by means of listening sessions with professional improvisers.

In the first section, we present how a grammar based on a constituent analysis can represent the hierarchical structure of a temporal scenario in order to create multi-level scenarios. We create such a grammar in collaboration with a professional musician for a particular chord progression called *rhythm changes*, considered here as a multi-level scenario. In the second section, we propose to use machine learning for grammar induction based on probabilistic methods of word sequence selection with mutual information. We apply this method on a corpus of *rhythm changes* and compare the generated grammar with the ground truth. In the third section, we present a new heuristic for generating improvisations on multi-level scenarios based on the notion of scenario anticipation. The generative model is evaluated via listening sessions and interviews with professional jazz musicians.

Using a grammar to model a multi-level structure

In this section, we present how to use a grammar to represent the hierarchical structure of a chord progression on several time levels. We first 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 multi-level aspect of a famous jazz chord progression called the *rhythm changes*.

Phrase structure grammar

Considering a set of symbols X, we denote X^* the set of finite sequences of elements from X. A grammar $G = (\Sigma, N, R, s)$ is defined by a set Σ of terminal symbols called alphabet, a set N of non-terminal symbols that is disjoint from N, a singular element $s \in N$ which 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" (Hopcroft and Ullman 1979).

A *phrase structure grammar* is a particular type of grammar based on a linguistic description of a language on a syntactic level using a constituent analysis, i.e. a breakdown of the linguistic functions following a hierarchical structure. Chomsky (1957) proposed an example phrase structure grammar for the construction of simple sentences in English, shown in Grammar 1. In this example, the non-terminal symbols are written in italics and the terminal symbols are written in regular font. 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.

Grammar 1 Example of phrase structure grammar.

1: $Sentence \to NP \;\; VP$
2: $NP \to Article \;\; Noun$
3: $VP \to Verb \;\; NP$
4: $Article \to$ a | the ...
5: $Noun \to$ man | ball ...
6: $Verb \to$ hit | took ...

A *derivation* is the sequence of rewrite rules used to obtain a particular sentence. It can be represented with a diagram. Figure 1 shows a diagram for the derivation of the sentence "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 only retains from the derivation the necessary information to determine the phrase structure. As such, it clearly shows the hierarchical syntactic structure of the sentence and its constituent analysis.

In this article, we are only considering a particular type of phrase structure grammar called

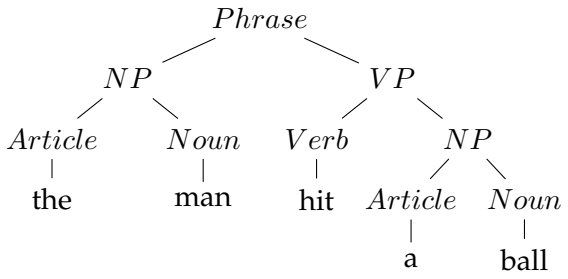

Figure 1. Diagram for the derivation of the sentence "the man hit a ball".

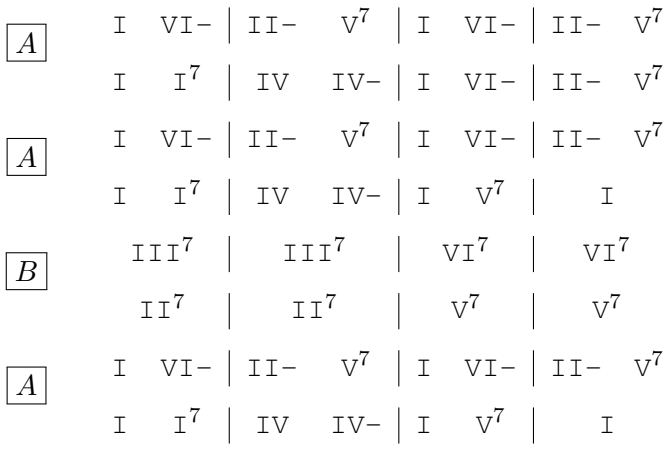

Figure 2. Chord progression of the tune I Got Rhythm by George Gershwin. The chords are shown as degrees with respect to the main tonality of the tune using jazz notation (hyphens indicate minor chords).

context-free grammars for which the rewrite rules R are a subset of $N \rightarrow (N \cup \Sigma)^*$.

A phrase structure grammar for *rhythm changes*

In order to show how a phrase structure grammar can be used to create multi-level scenarios, we decided to create such a grammar for a popular *bebop* chord progression: the *rhythm changes*. To do so, we analysed the set of *rhythm changes* from the *Omnibook* corpus (Déguernel et al. 2016) with Pascal Mabit, a professional jazz musician and teacher. This corpus consists of the chord progressions on the theme and the transcribed voicings played during Parker's solos of the 10 tunes with the rhythm changes form found in the Omnibook, leading to 26 variations of rhythm changes. The tunes from the *Omnibook* with the rhythm changes form are : *An Oscar for Treadwell*, *Anthropology*, *Celerity*, *Chasing the Bird*, *Kim* (2 versions), *Moose the Mooche*, *Passport*, *Red Cross*, *Steeplechase* and *Thriving from a Riff*. The chords are 2-beat long each and are annotated with their harmonic function relative to the tonic of the tune's main key.

Rhythm changes tunes follow a 32-bar chord progression, based on the tune *I Got Rhythm* by George Gershwin. The name *rhythm changes* is actually a shorthand for the "*chord changes of I Got*

Rhythm" that are shown in Figure 2. The global form of the *rhythm changes* is $AABA$, with a B section called *bridge* contrasting with the A sections:

- The A sections are 8-bar long sections with fast changing chords (usually two chords per bar) staying close to the initial tonality. These sections consist of:
  - a series of two *turnarounds* on bars 1&2 and 3&4. The *turnaround* is a 2-bar tonal function based on an arc of the circle of fifths. Its most characteristic form is `I VI- II- V` but it has many variations based on the usual substitutions (for instance, `I I II-` $\mathtt{V}^7$, `III-` $\mathtt{VI}^7$ $\mathtt{II-}$ $\mathtt{V}^7$, or `I` $\mathtt{VI}^7$ `II-` $\flat\mathtt{II}$ etc.). The first *turnaround* always starts with a `I`st degree to highlight the tonality at the beginning of the section. We denote by τ the set of all variations of *turnarounds* and denote by $\tau_{\mathtt{I}}$ the subset of *turnarounds* starting with a `I`st degree ($\tau_{\mathtt{I}} \subset \tau$).
  - a temporary tonicization of the sub-dominant (`IV`th degree) on bars 5&6. Its characteristic form is `I` $\mathtt{I}^7$ `IV IV-` but it also has many variations based of the usual substitutions (for instance, `V-` $\mathtt{I}^7$ `IV` $\sharp\mathtt{IV}^\circ$, or `I` $\mathtt{I}^7$ $\mathtt{IV}^7$ $\flat\mathtt{VII}^7$, etc.). We denote the set of all variations of this harmonic function by σ.
  - a last *turnaround* on bars 7&8. Except for the first A, this *turnaround* can be replaced with a *loop cadence* in order to either end on a tonic chord or anticipate the following chord function. For instance, a *loop cadence* can be `II-` $\mathtt{V}^7$ `I I`. We denote the set of all variations of *loop cadence* by ω.
- The B section is an 8-bar long section with a slower chord progression (usually, each chord is played over 2 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 the 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 one bar of `II-` followed by a bar of $\mathtt{V}^7$ (or of $\flat\mathtt{II}^7$). We denote by $\delta_{\mathtt{X}}$ the tonal function of dominant on degree `X`.

The *rhythm changes* is an interesting case study for our application because there exist many variations of this chord progression. This is actually a major 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 turn of a chorus. Using a phrase structure grammar in order to analyse 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 *rhythm changes* where the chords are the terminal symbols. Grammar 2 shows the grammar for *rhythm changes* created by Pascal Mabit that will be used as a ground truth in the next section. Figure 3 shows the diagram for the derivation of this grammar on one instance of *rhythm changes* from the *Omnibook* entitled *Celerity*.

Grammar 2 *Rhythm changes* phrase structure grammar

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

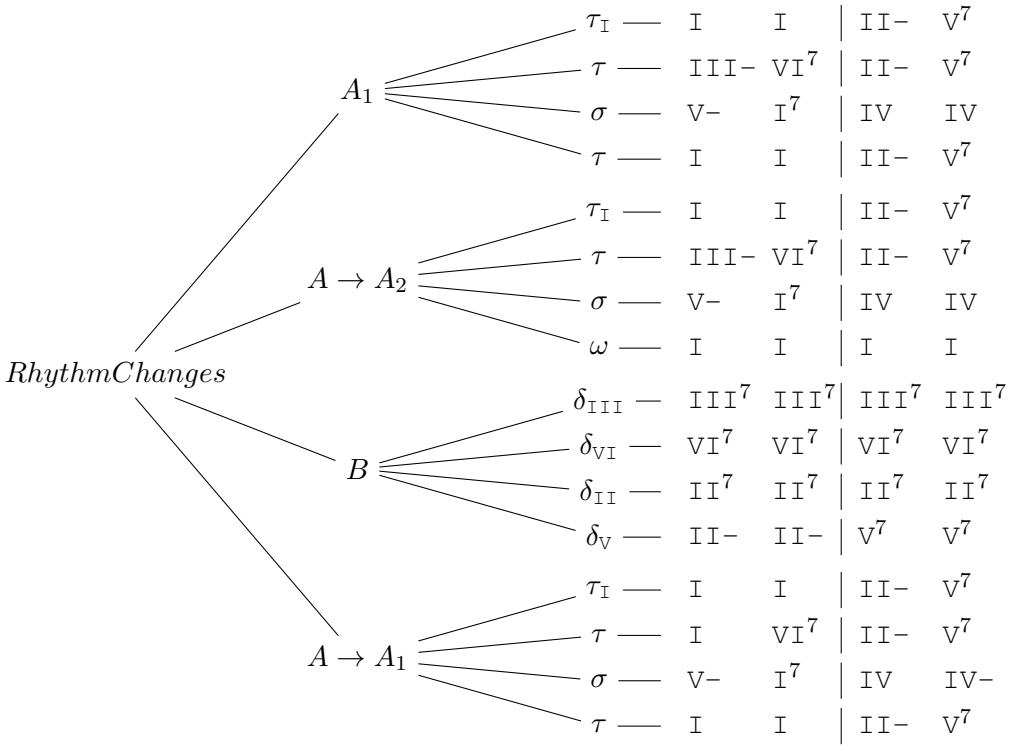

Figure 3. Diagram for the derivation of the rhythm changes on the tune Celerity from the Omnibook. Each chord lasts two beats. $A \rightarrow A_2$ and $A \rightarrow A_1$ denote the use of Rule 4 from Grammar 2.

Learning of multi-level structures

In this section, we propose a method of unsupervised grammar induction to automatically learn a multi-level structure from a training corpus. In our case, the training corpus is composed of several scenarios (here, chord sequences) which 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 theory input by adapting the sequence selection method by Zitouni et al. (2000), originally introduced for text, to the specific problems encountered in musical sequences.

Inducing a context-free grammar with sequence selection

The main idea of this method is to segment all sequences in the corpus simultaneously and iteratively 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 :

$$J(a,b) = \log \frac{\texttt{count}(a\ b)N}{\texttt{count}(a)\texttt{count}(b)}\ , \tag{1}$$

where `count` is the number of occurrences in the corpus ($\texttt{count}(a\ b)$ is the number of occurences where a appears right before b in the corpus) and N the size of the corpus. A high value of mutual information means that the symbols a and b occur as a sequence much more frequently than can be expected from pure chance. The iterative unification of symbols sharing the highest mutual information therefore leads to a hierarchical structure.
We applied Zitouni et al.'s original method on the *Omnibook rhythm changes* sub-corpus.

We obtained mixed results and identified two main limitations to this method:

1. Iterative unification leads to a clumping phenomenon around rare symbols appearing only in a single context. This is due to the limited amount of data. If a symbol appears in one context only, it will share a very high mutual information with its neighbours; it will therefore be unified with one of them, creating a new symbol appearing in one context only. This creates a vertical hierarchical structure centered on this symbol.

2. 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 while these sequences have the same tonal function). This results in bad high-level structures since, at higher levels, we obtain a too large set of symbols for too little data.

In order to deal with the first problem, we introduce 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 prioritise the unification of pairs of symbols of short length. To this end, we propose to normalise the mutual information by the length of the symbols:

$$\tilde{J}(a,b) = \frac{1}{l(a)+l(b)} \log \frac{\texttt{count}(a\ b)N}{\texttt{count}(a)\texttt{count}(b)}\ . \tag{2}$$

In order to deal with the second problem, we propose when creating a new symbol (by merging two other symbols) to check whether it is equivalent to another existing symbol of the same length using the sequential structure. We want to consider as equivalent two symbols of the same

length with similar neighbourhood in terms of mutual information. Two symbols a and b are considered equivalent if

$$\Psi(a,b) = \frac{1}{K}\sum_{u,v}(J(u,a) - J(u,b))^2 + (J(a,v) - J(b,v))^2 \leq \xi \ , \tag{3}$$

with K the size of the vocabulary, u the list of symbols on the left of a and b, v the list of symbols on the right of a and b, and ξ a threshold 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 neighbour of a or b (on the left and on the right). $\Psi(a,b)$ is the sum of all these distances, normalised by the size of the vocabulary. The lower $\Psi(a,b)$, the more a and b are used in similar context. If a and b are used in similar enough contexts, then the two symbols are unified. Though it has been set manually here, the value of the threshold ξ could also be set using machine learning on a validation corpus either in a supervised way, based on a ground-truth, or in an unsupervised way using characteristics of the inferred grammar and derivation (minimum depth, minimum number of symbols, etc.).

We propose Algorithm 1 for grammar induction from a corpus of scenarios.

Algorithm 1 Grammar induction from a corpus of scenarios

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

Evaluation of the induced grammar

We applied this algorithm on the *Omnibook rhythm changes* sub-corpus. Figure 4 shows the hierarchical analysis obtained on a *rhythm changes* example. The automatic analyses of the 26 variations of *rhythm changes* from the corpus have been studied and have all been validated by Pascal Mabit.

First, the different tonal functions (*turnaround*, *sub-dominant tonicization*, etc.) have been properly segmented and the structural organisation is correct on all desired levels (chords, tonal functions, and sections), and for all variations of rhythm changes. Second, the different variations of a given tonal function have been correctly identified as equivalent.

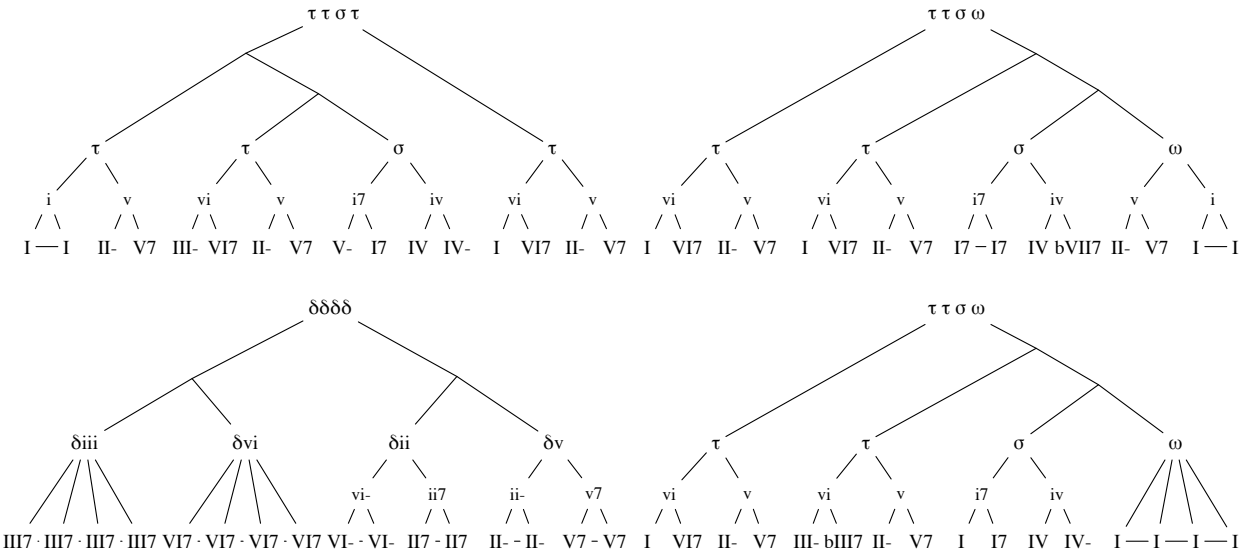

Figure 4. Hierarchical analysis of a rhythm changes obtained with Algorithm 4. The different sections are separated in 4 sub-trees, the last level of organisation isn't shown. Note that the lowercase roman numerals in the first level of analysis above chords represent here 2 chords long harmonic functions and not minor chords.

For instance, `I VI`^{7} `II- V`^{7} and `III- ♭III`^{7} `II- V`^{7} are considered equivalent, and also `V- I`^{7} `IV IV-` and `I`^{7} `I`^{7} `IV ♭VII`^{7}. This analysis also reveals another level of organisation between the chord level and the functional level that does not appear in the ground truth grammar but that was deemed semantically meaningful by the musician.

However, this analysis is less accurate than the ground truth on a couple of points:

- First, the different types of A section (A_1 and A_2 in Grammar 2) 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 with a `I`st degree (this specific variant of *turnaround* is called $\tau_{\texttt{I}}$ in Grammar 2).

Overall, the results of this method are promising: the obtained multi-level structures are very close to those found by a professional musician. It must be noted that we used a corpus of scenarios of the same form; this method could therefore be applied immediately to other corpora representative of another form, like the *blues* for instance. It would be interesting to compare this method with other grammar induction methods which are applicable to limited-size corpora. It would also be interesting to apply this method to larger and more varied corpora such as the *Realbook*, which involves different forms, and to other genres (classical music, pop music, Georgian music...). Early experiments on the *Realbook* make us expect this method to be able to still detect the

most common harmonic functions and their substitutions and to recognise harmonic similarities across different forms, similarly as Haas et al. (2009).

Using a multi-level structure for guided improvisation

In this section, we first present how to guide an automatic improvisation system using a scenario, following Nika et al. (2017a), and then we propose a new method using a multi-level scenario to enrich the improvisation in a more flexible way. The models are then compared during listening sessions and interviews with professional musicians.

Improvisation on a temporal scenario

Our method is based on the work by Nika et al. (2017a) who introduce a temporal scenario, modelled as a sequence of *labels*, in the guiding of improvisation. During training, the system's memory is constructed as a sequence of contents from the improvised dimension organised, with a factor oracle (Allauzen et al. 1999), from the scenario labels. In the case of jazz music, the labels can be the chords from 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 with anticipatory behaviours, and with the past of the memory using the properties of the factor oracle (Assayag et al. 2006). Musical contents from the chosen states are then played to generate the improvisation.

The generation process is divided in two successive steps: an *anticipation step* followed by a *navigation step* (see (Nika et al. 2017a) for a more detailed description of the algorithm). Let us denote by $S = S_1...S_s$ the scenario, t the current position in the scenario and let us consider a memory represented with a factor oracle with states $0...m$ and labels $\Lambda_0...\Lambda_m$.

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

$$\text{Future}(t) = \{j \in [0...m] \mid \exists c_{\text{future}} > 0, \Lambda_j...\Lambda_{j+c_{\text{future}}-1} \in \text{Pref}(S_t...S_s)\} \ . \tag{4}$$

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

$$\text{Past}(i) = \{j \in [0...m] \mid \exists c_{\text{past}} \in [1, j], \Lambda_{j-c_{\text{past}}+1}...\Lambda_j \in \text{Suff}(0...i)\} \ . \tag{5}$$

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

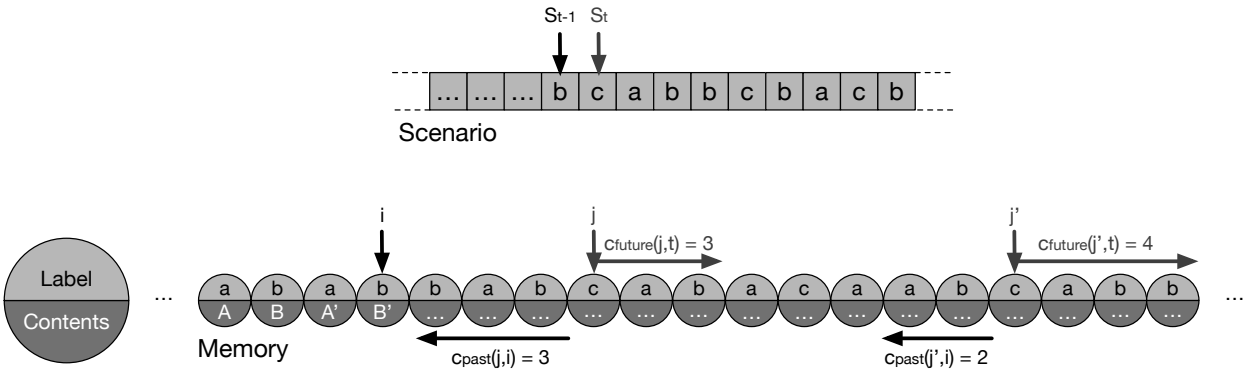

Figure 5. Creation of the set of states in the memory sharing a common future with the current position in the scenario t and a common past with the current event in the memory i_{t-1} for the anticipation step, following Nika et al. (2017a).

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

$$j \in \text{Future}(t) \quad \text{and} \quad j-1 \in \text{Past}(i) \ . \tag{6}$$

Figure 5 shows an anticipation step.

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

$$\Lambda_j = S_t \quad \text{and} \quad j-1 \in \text{Past}(i) \ . \tag{7}$$

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

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

Using multi-level information

Now, we want to take a multi-level scenario into account in the generation process. In this case, the scenario is not just a sequence of symbols anymore, but a sequence of lists of symbols corresponding to each level that we call *multi-level labels*. The memory is therefore built from multi-level labels corresponding to the multi-level scenario. Figure 6 shows an example of multi-level scenario with three levels of temporal organisation.

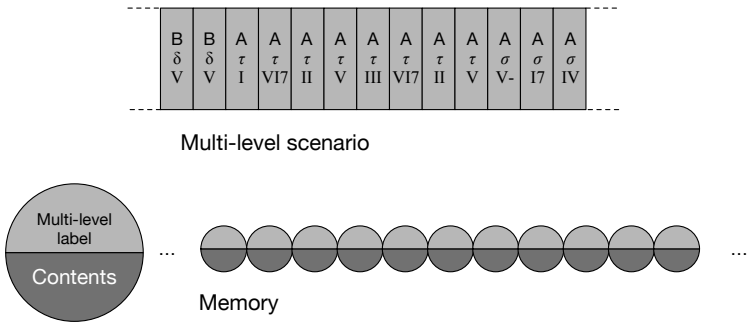

Figure 6. Example of a multi-level scenario and a multi-level memory.

We adapted the generation method presented in the previous section in order to account for the information brought by the multi-level aspect in both the anticipation step and the navigation step. For each step, we want to extend the possible positions in the memory to states that share equivalent multi-level labels to those of the scenario, that is to say labels sharing common information on one or more levels but possibly different information on the other levels. For instance, for jazz music, we could accept positions in the 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 the memory but share the same tonal function as a known chord. We would also like to prioritise positions in the memory sharing a strong common information with the future of the scenario and the past of the memory. 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 as $S = S_1...S_s$ the given multi-level scenario (in our case, $S_1, ..., S_s$ are chords), of length s with Q levels, with $\forall p \in [1...s], S_p = \{S_p^1...S_p^Q\}$. Let t be the current position in the scenario and let us consider a memory built with a factor oracle with states $0...m$ and multi-level labels $\Lambda_0...\Lambda_m$, with $\forall p \in [0...m], \Lambda_p = \{\Lambda_p^1...\Lambda_p^Q\}$. We consider that two multi-level 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 $u = \left\{ \begin{smallmatrix} A \\ \tau \\ \text{I} \end{smallmatrix} \right\}$, and $v = \left\{ \begin{smallmatrix} A \\ \tau \\ \text{III-} \end{smallmatrix} \right\}$, we have $u \simeq v$. By extension, we consider that a sequence of multi-level symbols $u = u_1...u_n$ with $\forall p \in [1...n], u_p = \{u_p^1...u_p^Q\}$ is an equivalent prefix of a sequence of multi-level symbols $v = v_1...v_m$ with $\forall p \in [1...m], v_p = \{v_p^1...v_p^Q\}$ if

$$u = u_1...u_n \in \text{Pref\_eq}(v = v_1...v_m) \iff \forall p \in [1...n], u_p \simeq v_p \ . \tag{8}$$

For instance, $u = \left\{ \begin{smallmatrix} A \\ \omega \\ \text{V7} \end{smallmatrix} \right\} \left\{ \begin{smallmatrix} A \\ \omega \\ \text{I} \end{smallmatrix} \right\} \in \text{Pref\_eq} \left(v = \left\{ \begin{smallmatrix} A \\ \omega \\ \text{I} \end{smallmatrix} \right\} \left\{ \begin{smallmatrix} A \\ \omega \\ \text{I} \end{smallmatrix} \right\} \left\{ \begin{smallmatrix} B \\ \delta_{III} \\ \text{III7} \end{smallmatrix} \right\} \left\{ \begin{smallmatrix} B \\ \delta_{III} \\ \text{III7} \end{smallmatrix} \right\} \right)$.
Similarly, we consider that a sequence of multi-level symbols $u = u_1...u_n$ with $\forall p \in [1...n], u_p = \{u_p^1...u_p^Q\}$ is an equivalent suffix of a sequence of multi-level symbols $v = v_1...v_m$ with $\forall p \in$

$[1...m], v_p = \{v_p^1...v_p^Q\}$ if

$$u = u_1...u_n \in \text{Suff\_eq}(v = v_1...v_m) \iff \forall p \in [1...n], u_p \simeq v_{p+m-n} \ . \tag{9}$$

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

1. For the anticipation step, we replace the sets Future(t) and Past(i) with the sets Future_eq(t) and Past_eq(i), respectively:

$$\text{Future\_eq}(t) = \{j \in [0...m] \ | \exists c_{\text{future}} > 0, \Lambda_j...\Lambda_{j+c_{\text{future}}-1} \in \text{Pref\_eq}(S_t...S_s)\} \ , \tag{10}$$

$$\text{Past\_eq}(i) = \{j \in [0...m] \ | \exists c_{\text{past}} \in [1, j], \Lambda_{j-c_{\text{past}}+1}...\Lambda_j \in \text{Suff\_eq}(0...i)\} \ . \tag{11}$$

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

$$j \in \text{Future\_eq}(t) \quad \text{and} \quad j-1 \in \text{Past\_eq}(i) \ . \tag{12}$$

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

$$\Lambda_j \simeq S_t \quad \text{and} \quad j-1 \in \text{Past\_eq}(i) \tag{13}$$

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

The similarity between two multi-level labels Λ_i and Λ_j is defined by

$$\varsigma(\Lambda_i, \Lambda_j) = \sum_q W_q \delta_{\Lambda_i^q, \Lambda_j^q} \ , \tag{14}$$

where δ is the Kronecker symbol.
For instance, if we consider the case with section level, tonal function level and chord level, and attribute the respective weights of 0.1, 0.6 and 0.3, we get

$$\varsigma\left(\left\{\begin{smallmatrix} A \\ \tau \\ \mathtt{I} \end{smallmatrix}\right\}, \left\{\begin{smallmatrix} A \\ \tau \\ \mathtt{III-} \end{smallmatrix}\right\}\right) = 0.1 + 0.6 = 0.7 \text{ and } \varsigma\left(\left\{\begin{smallmatrix} B \\ \delta_{VI} \\ \mathtt{III-} \end{smallmatrix}\right\}, \left\{\begin{smallmatrix} A \\ \tau \\ \mathtt{III-} \end{smallmatrix}\right\}\right) = 0.3.$$

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

$$\sum_{k=0}^{c_{\text{future}}(j,t)} \varsigma(\Lambda_{j+k}, S_{t+k}) + \sum_{k=0}^{c_{\text{past}}(j,i)} \varsigma(\Lambda_{j-k}, \Lambda_{i-k}) \tag{15}$$

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

$$\varsigma(\Lambda_j, S_t) + \sum_{k=0}^{c_{\text{past}}(j,i)} \varsigma(\Lambda_{j-k}, \Lambda_{i-k}) \ . \tag{16}$$

where $c_{\text{future}}(j,t)$ and $c_{\text{past}}(j,i)$ are respectively the maximum length of c_{future} associated to $j \in$ Future_eq(t) and of c_{past} associated to $j \in$ Past_eq(i).

With these scores, several strategies can be applied to prioritise elements sharing more common information with the future of the scenario and the past of the memory. 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 normalise the scores with the sum of the scores of all possible events and then pick an event in a random fashion following the obtained pseudo-probabilities.

Evaluation and musicians' feedback

In order 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 the case of *rhythm changes* from the *Omnibook* corpus by Charlie Parker. For these interviews, multi-level scenarios of *rhythm changes* were generated using the ground-truth grammar created by Pascal Mabit. Therefore, three levels of organisation were considered: chords, tonal functions, and sections. For each improvisation generated, the contents in the memory are based on a theme and an improvisation from one *rhythm changes* from the *Omnibook*. We used the improvisations on *Thriving from a Riff*, *An Oscar for Treadwell*, and *Anthropology*. The memory is divided per beat in order to match more easily the labels of the memory with the labels of the scenario. In these scenarios, we generated improvisations using:

1. the approach following Nika et al. (2017a) (hereafter called base method): the anticipation of the scenario and the navigation in the memory use only the chord level of the scenario (the other levels are not taken into account); when a chord in the scenario doesn't appear in the memory, key transposition is applied;

2. the proposed multi-level approach: the anticipation of the scenario and the navigation in the memory use all the multi-level information of the scenario.

Examples of generated improvisations with both methods are available at http://repmus.ircam.fr/dyci2/demos/multi-level. Each musician listened to a dozen improvisations for each method. Each time, for a given chord progression generated by the phrase structure grammar, an improvisation was generated with both methods.

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

> *"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 detailed this idea noticing that with the proposed method, the improvisation is driven around the functions and the sections, creating phrases (or sequences of phrases) with a longer term consistence:

> *"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 proposed method were also deemed less disjointed. This can be explained by the fact that this method opens up more possibilities in the navigation, enabling the system to make fewer jumps in the 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 the memory, the freedom given by the proposed method to play on other chords sharing the same tonal function and section also helped to create less fragmented improvisations, 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*

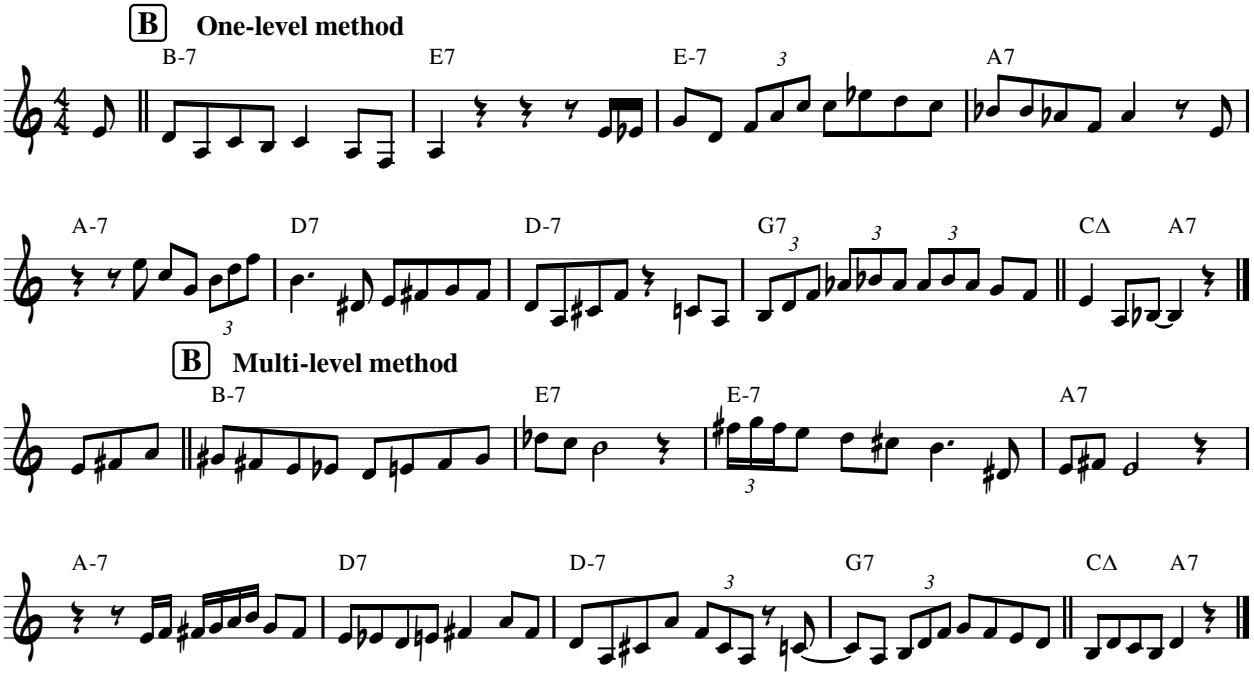

Figure 7. Comparison of improvisations on the B section of a given rhythm changes B using the one-level scenario and the multi-level scenario with the memory of An Oscar for Treadwell.

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 sub-dominant / dominant chords: $\texttt{VII}^{-7}$ $\texttt{III}^7$ $\texttt{III}^{-7}$ $\texttt{VI}^7$ $\texttt{VI}^{-7}$ $\texttt{II}^7$ $\texttt{II}^{-7}$ $\texttt{V}^7$. Bourhis said:

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

Figure 7 shows examples of improvisations on the B section using either the one-level scenario approach or the multi-level scenario approach and the same memory from *An Oscar for Treadwell*. A similar remark was made by Mabit: in some A sections, the sub-dominant function had a diminished chord in it, which is a chord class that does not exist in the memory of *Anthropology*; in this condition, the base method could not generate an improvisation, but we could do it with the proposed method and we obtained convincing results. Figure 8 shows an example of this situation. Moreover, being able to play chords from the scenario that are different from the ones in the memory brings some form of creativity, whereby the guide notes and the extensions give new colours to the improvisation.

More surprisingly, another improvement noticed by the musicians is that the improvisations generated with the proposed method have better rhythmic consistency. The development of the improvisation from the points of view of density and rhythmic flow was deemed more realistic.

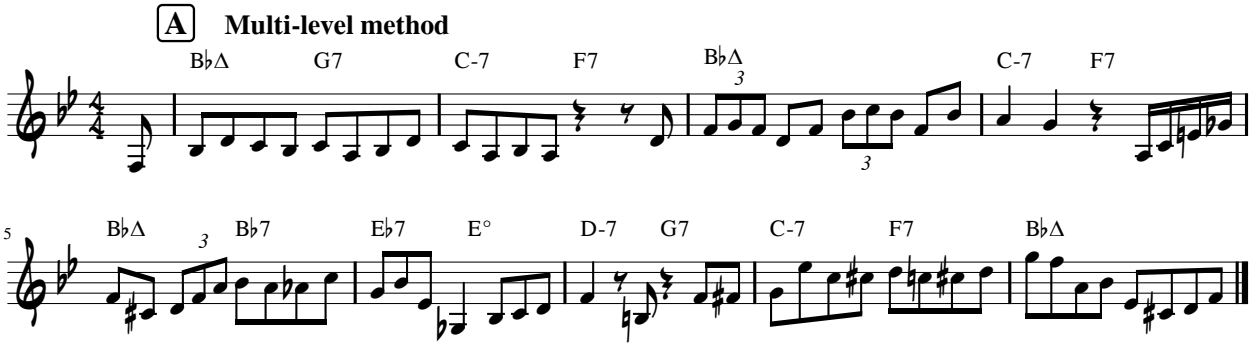

Figure 8. Improvisation on the A section of a rhythm changes with a diminished chord (E°) in the subdominant function with the memory of Anthropology.

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 rhythmically things that are much more natural."*

Overall, the proposed method received little criticism by the musicians. Bourhis still noticed some unsettling jumps in the memory (for instance, octave leaps that were not realistic), but their frequency was much lower 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."*

Overall, the musicians' feedback on this experiment is gratifying. The understanding of the multi-level structure of a harmonic progression enables the generation of more realistic improvisations. On top of being able to improvise more freely and in a more varied fashion (even on previously unseen chords), this improves melodic and rhythmic consistency. The generation "does not focus on the details to serve a better organisation at a larger scale", making it closer to the improvisation process employed 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 Pascal Mabit and Louis Bourhis, using this time 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.

Conclusion and discussion

We presented a method to analyse and use the multi-level structure of a tune for music generation in the case 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 upon several levels of organisation. Then, we proposed a grammar induction method based on probabilistic sequence selection to perform automatic analysis of a scenario to obtain a multi-level scenario. We obtained satisfactory results close to the ground truth defined in collaboration with a professional musician. Next, we designed new heuristics for guided improvisation based on a multi-level 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 unseen events. This system was evaluated with professional improvisers during listening sessions and interviews who concluded that taking into account the hierarchical structure of the scenario led to more realistic improvisations with better melodic and rhythmic consistently both on a local and global level.

A possible extension of this work would be to exploit the generative grammar to create a system where 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 multi-level 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 (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 meta-composition of scenarios started by Nika et al. (2017a) to the meta-composition of multi-level scenarios.

Acknowledgments

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 *Proceedings of SOFSEM'99, Theory and Practice of Informatics*. pp. 291–306.

Assayag, G., G. Bloch, M. Chemillier, A. Cont, and S. Dubnov. 2006. “OMax Brothers : a dynamic topology of agents for improvization learning.” In *Proceedings of the 1st ACM Workshop on Audio and Music Computing for Multimedia*. pp. 125–132.

Bickerman, G., S. Bosley, P. Swire, and R. M. Keller. 2010. "Learning to create jazz melodies using deep belief nets." In *Proceedings of the International Conference on Computational Creativity*. pp. 228–236.

Bigo, L., M. Giraud, R. Groult, N. Guiomard-Kagan, and F. Levé. 2017. "Sketching sonata form structure in selected classical string quartets." In *Proceedings of the International Society for Music Information Retrieval Conference*. pp. 752–759.

Bimbot, F., E. Deruty, G. Sargent, and E. Vincent. 2016. "System & 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: Walter de Gruyter.

Déguernel, K., J. Nika, E. Vincent, and G. Assayag. 2017. "Generating equivalent chord progressions to enrich guided improvisation : application to Rhythm Changes." In *Proceedings of the the 14th Sound and Music Computing Conference*. pp. 399–406.

Déguernel, K., E. Vincent, and G. Assayag. 2016. "Using multidimensional sequences for improvisation in the OMax paradigm." In *Proceedings of the 13th 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.

Donzé, A., R. Valle, I. Akkaya, S. Libkind, S. A. Seshia, and D. Wessel. 2014. "Machine improvisation with formal specifications." In *Proceedings of the 40th 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." Ph.D. thesis, 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., R. Groult, E. Leguy, and F. Levé. 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." Ph.D. thesis, Université Rennes 1.

Haas, W. B. D., M. Rohrmeier, R. C. Veltkamp, and F. Wiering. 2009. "Modeling harmonic similarity using a generative grammar of tonal harmony." In *Proceedings of the 10th International Society for Music Information Retrieval Conference*. pp. 549–554.

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

Keller, R., A. Schofield, A. Toman-Yih, Z. Merritt, and J. Elliott. 2013. "Automating the Explanation of Jazz Chord Progressions Using Idiomatic Analysis." *Computer Music Journal* 37(4):54–69.

Keller, R. M., A. Toman-Yih, A. Schofield, and Z. Merritt. 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):4:1–27.

Nika, J., K. Déguernel, A. Chemla-Romeu-Santos, E. Vincent, and G. Assayag. 2017b. "DYCI2 agents : merging the "free", "reactive" and "scenario-based" music generation paradigms." In *Proceedings of the 43rd 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 22nd International Conference on Principles and Practice of Constraint Programming*. pp. 769–785.

Roy, P., and F. Pachet. 2013. "Enforcing meter in finite-length Markov sequences." In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*. pp. 854–861.

Steedman, M. 1996. "The Blues and the Abstract Truth : music and mentals models." In J. Oakhill, and A. Garnham, (editors) *Mental Models in Cognitive Science*. Mahwah: Erlbaum, 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 Interspeech*. pp. 562–565.