GENERATING EQUIVALENT CHORD PROGRESSIONS TO ENRICH GUIDED IMPROVISATION : APPLICATION TO RHYTHM CHANGES

Ken Déguernel[1,2] Jérôme Nika[2] Emmanuel Vincent[1] Gérard Assayag[2]
[1]Inria, F-54600 Villers-lès-Nancy, France
[2]IRCAM STMS Lab (CNRS, UPMC, Sorbonne Universités)
ken.deguernel@inria.fr jerome.nika@ircam.fr
emmanuel.vincent@inria.fr gerard.assayag@ircam.fr

ABSTRACT

This paper presents a method taking into account the form of a tune upon several levels of organisation to guide music generation processes to match this structure. We first show how a phrase structure grammar can represent a hierarchical analysis of chord progressions and be used to create *multi-level progressions*. We then explain how to exploit this multi-level structure of a tune for music generation and how it enriches the possibilities of guided machine improvisation. We illustrate our method on a prominent jazz chord progression called 'rhythm changes'. After creating a phrase structure grammar for 'rhythm changes' with a professional musician, the terminals of this grammar are automatically learnt on a corpus. Then, we generate melodic improvisations guided by multi-level progressions created by the grammar. The results show the potential of our method to ensure the consistency of the improvisation regarding the global form of the tune, and how the knowledge of a corpus of chord progressions sharing the same hierarchical organisation can extend the possibilities of music generation.

1. INTRODUCTION

In jazz music, and more generally in improvised music, the inherent form of a chord progression is often composed upon several levels of organisation. For instance, a subsequence of chords can create a tonal or modal function and a sequence of functions can be organised as a section. This multi-scale information is used by musicians to create variations of famous chord progressions whilst keeping their original feel. In this article, we define as *equivalent* two chord progressions sharing the same hierarchical organisation. We call a *multi-level progression* an entity describing an analysis of a music progression upon several hierarchical levels (for instance chord progression, functional progression, sectional progression, etc.) that can be used to generate equivalent chord progressions. We want to design a musical grammar from which we can infer this hierarchical information and then use it to extend the possibilities of music generation.

Machine improvisation systems use style modelling to learn the musical style of a human improviser. Several methods of style modelling have been used for music generation including statistical sequence modelling [1–3], extended Markov models [4], methods using deep and recurrent neural networks [5, 6], and methods based on the factor oracle, a structure from formal language theory [7, 8] adapted to a musical context in [9]. Studies around style modelling have made this structure a proficient tool for improvised performance and interaction. These studies involve the creation of specific navigation heuristics [10], its adaptation to audio features [11, 12] or its integration in a system mixing background knowledge with the local context of an improvisation [13]. However, none of these systems takes the long-term structure of the improvisation into account.

Improvising on idiomatic music such as jazz [14], where the improvisation is guided by a chord progression, has been a major interest in machine improvisation. Donze et al. introduced the use of a control automaton to guide the improvisation [15]. Nika et al., with ImproteK [16], designed a system of *guided improvisation*, that is to say a system where the music generation model uses prior knowledge of a temporal scenario (e.g. a chord progression). The music generation model anticipates the future of a scenario while ensuring consistency with the past of the improvisation. It received positive feedback from professional improvisers and evolved according to musicians expertise [17]. However, these scenarios are purely sequences of symbols. The chord progression is only considered on a local level and it does not take into account the fact that the same harmonic progression can have different roles when played in different parts of a tune. Therefore, the improvisation does not adapt to the global structure. We would like the improvisation to take this global structure into account in a similar way as humans apply syntactic cognitive processes capable of handling the hierarchical structure of a song [18].

Form analysis, i.e. the problem of defining and analysing the global structure of a piece of music, is a major topic in Music Information Retrieval and Computational Musicology and it has been studied with methods from different fields. Giraud et al. use methods from bioinformatics and

dynamic programming, based on the Mongeau-Sankoff algorithm [19], to perform fugue analysis [20] or theme and variations recognition [21]. Geometric approaches have also been used [22]; for instance, algorithms using the spiral array can be used to determine key boundaries [23] and therefore reveal tonal functions. Another approach using grammars was introduced in [24] for structure, rhythm and harmony analysis in tonal music. Some generative grammars can infer a hierarchical multi-level phrase structure [25]. This type of structure has been used in [26] to compute harmonic similarity of chord sequences. However, the above form analysis methods were used for music analysis only. Focusing on music generation instead, Steedman proposed a generative grammar based on rewrite rules for jazz music [27]. This grammar was used in ImproteK by Chemillier et al. [28] to create new instances of a given chord progression. However, we would like to use generative grammars to create a hierarchical analysis of a chord progression, thus creating a multi-level progression, and then directly involve it in the generative aspect of the improvisation.

In this article, we present two contributions. First, we propose a method formalising multi-level progression with phrase structure grammars. The structural aspect of equivalent chord progressions is analysed with a musician but the actual contents of the multi-level progression are then automatically learnt on a corpus. The grammar enables the system to generate multiple instances of a given progression rather than a single one and to provide its hierarchical structure. Second, we introduce a method using the information from a multi-level progression to expand the current methods of music generation. This method is based on the algorithms from [16] for music generation that we adapt to take into account the information from a multi-level progression; thus, creating a system able to improvise on a progression with changing voicing and considering its position in the structure of the tune. We apply this method to 'rhythm changes', the most played chord progression of jazz music, after the *blues* [29]. We first create a phrase structure grammar for 'rhythm changes', and then generate melodic improvisations over multi-level progressions provided by the grammar.

In section 2, we explain how a phrase structure grammar can model a multi-level progression by creating a hierarchical structure and we create a grammar for 'rhythm changes'. Then, in section 3, we introduce a heuristic to generate a music sequence following a multi-level progression. In section 4, we describe our experiments and the results obtained with this new generation model. Finally, we conclude and propose some perspectives for this generation model in section 5.

2. GENERATIVE GRAMMAR AND SYNTACTIC STRUCTURE

In this section, we present how we can use a phrase structure grammar to create a multi-level progression from equivalent chord progressions. First, we present the definition of a phrase structure grammar, and then show an application to a jazz chord progression: the 'rhythm changes'.

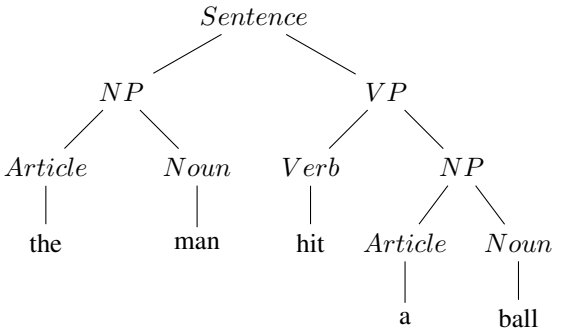

Figure 1. Diagram of the derivation of the sentence "the man hit a ball".

2.1 Phrase structure grammar

A grammar $G = (N, \Sigma, R, s)$ is defined by:

- two disjoint finite sets of symbols: N the set of non-terminal symbols, and Σ the set of terminal symbols,
- a singular element $s \in N$ called the axiom,
- a finite set R of rewrite rules included in $(N \cup \Sigma)^* N (N \cup \Sigma)^* \rightarrow (N \cup \Sigma)^*$

where * is the Kleene star, i.e., X^* is the set of finite sequences of elements of X.

Phrase structure grammars are a type of grammar presented by Noam Chomsky, based on constituent analysis, that is to say on a breakdown of linguistic functions within a hierarchical structure. In [25], Chomsky presented this example of phrase structure grammar:

(i)	$Sentence \rightarrow NP + VP$
(ii)	$NP \rightarrow Article + Noun$
(iii)	$VP \rightarrow Verb + NP$
(iv)	$Article \rightarrow$ a, the...
(v)	$Noun \rightarrow$ man, ball...
(vi)	$Verb \rightarrow$ hit, took...

Rule (i) should be read as "rewrite $Sentence$ as $NP + VP$", i.e., a *sentence* consists of a *noun phrase* followed by a *verb phrase*. Rule (ii) should be read as "rewrite NP as $Article + Noun$", i.e., a *noun phrase* consists of an *article* followed by a *noun*, etc.

A *derivation* is the sequence of rules applied to create a specific sentence. Figure 1 shows a diagram representing the derivation of the sentence "the man hit a ball". This diagram does not convey all the information about the derivation since we cannot see the order in which the rules were applied. Nevertheless, it clearly shows the hierarchical syntactic structure of this sentence, thus creating a visualisation of the constituent analysis.

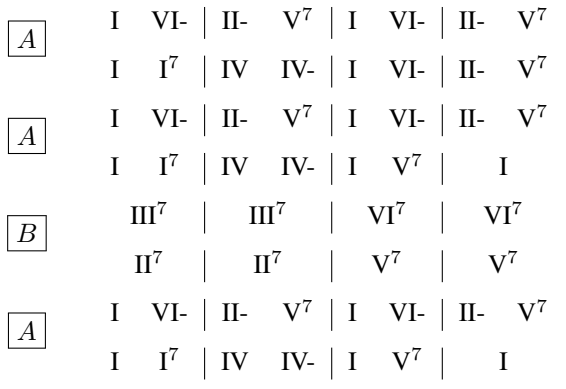

Figure 2. Original chord progression of 'I Got Rhythm'.

2.2 Application to 'rhythm changes'

In order to test the formalisation of multi-level progressions with phrase structure grammars, we create such a grammar for a specific jazz chord progression: the 'rhythm changes'. We show how we can obtain a hierarchical analysis of the chord progression and that it can be used to generate equivalent chord progressions.

The 'rhythm changes' is a 32-bar chord progression used in George Gershwin's tune 'I Got Rhythm' ('rhythm changes' is short for "chord changes of 'I Got Rhythm'"). Figure 2 shows the original version of this chord progression. The main feature of this chord progression is its $AABA$ structure with a B section (the bridge) that contrasts sharply with the A section.

- The A section is an 8-bar structure with fast changing chords based on:
  - a series of *turnarounds*: a 2-bar function (for instance, I VI- II- V^7) affirming the tonic of the tune on bars 1&2, 3&4 and 7&8. We note τ a *turnaround* on the tonic. We also note τ_I as a restriction of *turnarounds* starting with a Ist degree chord ($\tau_1 \subset \tau$).
  - a short modulation to the IVth degree on bars 5&6. We note σ this modulation to the IVth degree.
- The B section is an 8-bar structure consisting of dominant seventh chords following the circle of fifths (III^7 VI^7 II^7 V^7). Each chord is played on a two bar span, giving a sense of key shifting. The improvisers usually emphasize this contrast, insisting on the guide notes (the 3rd and the 7th) of these dominant seventh chords. We note δ_i these 2-bar dominant seventh chords on the i^{th} degree functions.

The 'rhythm changes' is an interesting case study for our method because of the amount of variations existing around this chord progression. This is actually one of the main interests for musicians; the changes can be modified on the fly without discussion as long as the functions are present. They mix various versions of 'rhythm changes' as they improvise [29], the chord progression can be different at every iteration. Using a phrase structure grammar to generate 'rhythm changes' seems therefore appropriate. Considering chords, functions and sections as constituents, we can create a phrase structure grammar embedding the hierarchical structure of the 'rhythm changes' where chords are the terminal symbols.

In order to create this phrase structure grammar for 'rhythm changes', we analysed with a professional jazz musician all the 'rhythm changes' from the Omnibook corpus [13, 30]. This sub-corpus consists of 26 derivations of 'rhythm changes' and contains the melodies and annotated jazz chord symbols [31]. We present here the grammar that we created. This grammar has then been validated with jazz musicians (cf. section 4.1).

$$
\begin{array}{cl}
(i) & \mathit{Rhythm\ Changes} \rightarrow A_1 + A_2 + B + A \\
(ii) & A_1 \rightarrow \tau_I + \tau + \sigma + \tau \\
(iii) & A_2 \rightarrow \tau_I + \tau + \sigma + \omega \\
(iv) & A \rightarrow A_1, A_2 \\
(v) & B \rightarrow \delta_{III} + \delta_{VI} + \delta_{II} + \delta_V
\end{array}
$$

$\tau_I, \tau, \sigma, \omega, \delta_{III}, \delta_{VI}, \delta_{II}, \delta_V$ are learnt on the corpus

- Rule (i) shows the $AABA$ structure of the 'rhythm changes'. These 8-bar sections are the biggest constituents after the whole chord progression itself.
- Rules (ii) and (iii) show the composition of an A section in four 2-bar functions. An A section starts with a τ_I in order to highlight the beginning of the section with a Ist degree chord. Then, another τ is played, followed by the modulation to the IVth degree σ. The difference between the first and second A section is on the last two bars. On the one hand, the last two bars of A_1 is another *turnaround* τ, and on the other hand, the last two bars of A_2 is an end slate on the tonic ω in anticipation of the bridge.
- Rule (iv) indicates that the final A section can be either A_1 or A_2.
- Rule (v) shows the composition of a B section in four 2-bar functions. Each δ represents respectively one key shift on the circle of fifth on degrees III VI II and V.
- The possible contents for each function are then learnt on the corpus. Each function is composed of a sequence of four chords with a duration of two beats each. Training them on a corpus enables the system to take many different possibilities into account for each function, and to constitute some form of style modelling for the generated chord progressions. The likelihood of each contents could be taken into account with probabilistic models [13] for a better emulation of the musical style. We trained the functions on the 'rhythm changes' sub-corpus from the Omnibook.

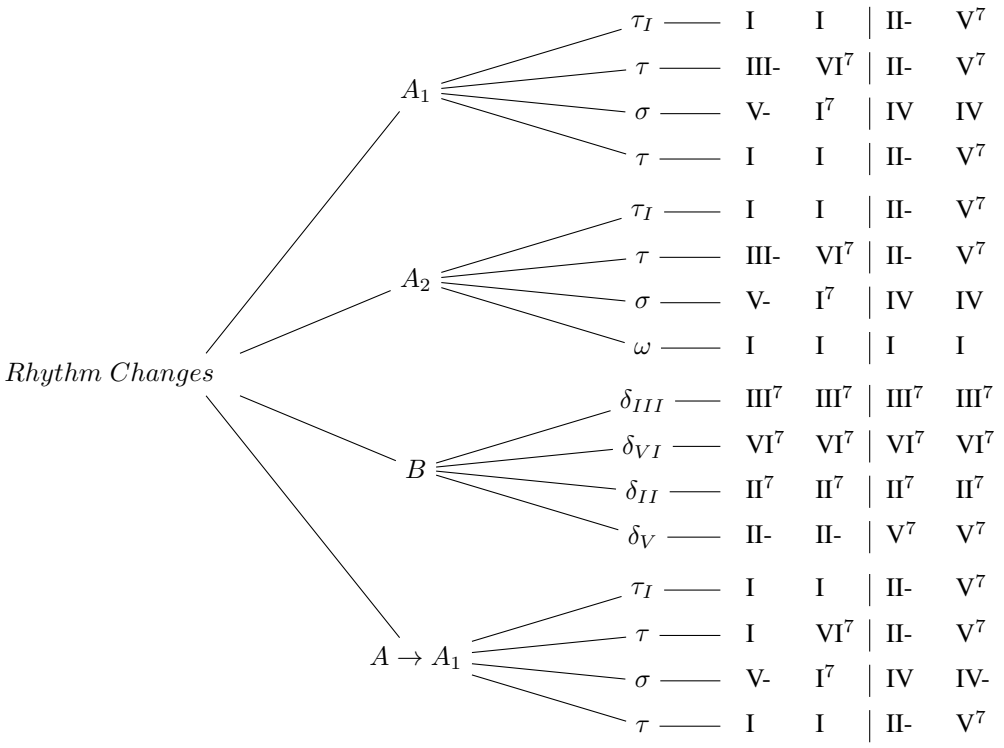

Figure 3. Diagram of a 'rhythm changes' derivation on Charlie Parker's theme *Celerity* (each chord lasts two beats). $A \to A_1$ denotes the application of rule (iv).

Figure 3 shows the diagram of one derivation of this grammar for one of the chord progressions on Charlie Parker's theme *Celerity*.

3. GENERATING MUSIC ON A MULTI-LEVEL PROGRESSION

In this section, we present a method using the information from a multi-level progression to enrich current music generation methods using the knowledge of the hierarchical structure and the equivalence between chord progressions. We first present the concepts we used as a basis for our work, and then introduce how to use the equivalent chord progressions of a multi-level progression in a music generation model and how it extends its possibilities.

3.1 Generating music on a chord progression

We propose to base our generation model on existing methods of machine improvisation guided by a scenario, to which we add the notion of multi-level progression.
The basis of our generation model is a simplified version of ImproteK [16] which is inspired by the work from OMax [32]. In OMax, improvising consists in navigating a memory on which we have information on the places sharing the same context, i.e. the same musical past. The fundamental premise is that it is possible to jump from one place in the memory to another with the same context. This non-linear path on the memory results in the creation of a new musical sentence that differs from the original material, but that retains its musical style [10]. ImproteK uses a similar approach, but with the addition of a temporal scenario used to guide the improvisation. The scenario is a prior known sequence of symbols (called *labels*) that must be followed during the improvisation. Contrary to Omax, the memory is not based on a sequence of pitches, but on a sequence of musical contents tagged by a label.

The goal of the associated generation model is to combine at every time in the generation an anticipation step ensuring continuity with the future of the scenario, and a navigation step keeping consistency with the past of the memory. Therefore, a phase of generation of a music sequence follows two succesive steps:

- The anticipation step consists in looking for an event in the memory sharing a common future with the current scenario. This is done by indexing the prefixes of the suffix of the current scenario in the memory using the regularites in the pattern [16]. Therefore, by finding such a prefix in the memory, we insure continuity with the future of the scenario.
- The navigation step consists in looking for events in the memory sharing a common context with what was last played in order to follow a non-linear path in the memory, thus creating new musical sentences.

As in OMax, the search for places in the memory sharing a common past is done thanks to the factor oracle: a structure from the field of formal language theory introduced by Crochemore et al. [7, 8]. Initially designed as an automaton recognising a language including at least all the factors

of a given word, the factor oracle has been widely used in machine improvisation systems such as OMax, ImproteK, PyOracle [11] or CatOracle [12].

3.2 Considering a multi-level progression

We now present a method to take into account a multi-level progression, that is to say a scenario that is not just a sequence of symbols. If we want to keep the previous formalism, each label is now a list of symbols corresponding to each scale. We therefore need to propose a way to adapt the previous method to take this additional information into account in both the anticipation step and the navigation step.

- For the anticipation step, we first look for a prefix of the suffix of the current multi-level progression in the memory with exact labels; the contents of the memory must match the contents of the scenario on every scale. By doing so, we ensure the consistency of the musical contents in the global context of the scenario. If such a prefix cannot be found, we proceed to a search with equivalent labels. For instance, we can accept places in the memory without the right chord, but with the right function and the right section. We can favour places in the memory sharing a similiar multi-level label as the one in the multi-level progression according to a score (see below). The system can thus react and generate an improvisation on previously unmet chords, as long as they share a similar role in the scenario as previously known chords.

- The navigation step is done in a similar way. Using the information given by the factor oracle, we first look for places in the memory sharing the same context with exact labels, but we also open the possibilities to equivalent labels. This way, we extend the realm of possibilities for music generation. Once again, we can favour places in the memory sharing similar multi-level label as the one in the multi-level progression (with a weight); thus, taking better consideration of the progression's hierarchical structure.

In order to compute a score of similarity between multi-level labels, a weight W_i can be attributed to each level in order to evaluate the similarity of the considered places in the memory with the scenario such as

$$\sum_{i \in \text{level}} W_i = 1.$$

In this way, we can prioritise the choice of places in the memory with high scores, or limit the choice to highest scores according to a threshold. For instance, a user can consider that matching the functions is more important than matching the chords.

Finally, Figure 4 sums up the whole process for generating music upon multi-level progressions.

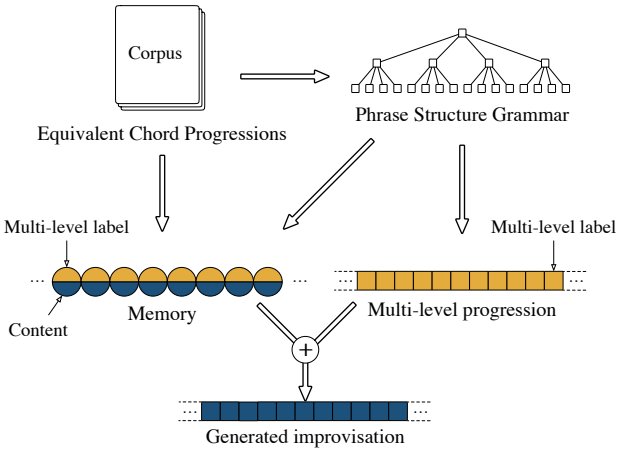

Figure 4. Process for generating an improvisation on a multi-level progression. First the phrase structure grammar is constructed with a musician using a corpus of equivalent chord progressions. Then the grammar is used to generate a multi-level progression, and used to provide information when creating the memory with multi-level labels. Finally, the improvisation is generated by navigating the memory guided by the multi-level progression.

I	II- V^7	I	II- V^7
I^7	IV^7	III- VI^7	II- V^7
I	II- V^7	III- VI^7	II- V^7
V- I^7	IV #IVo	I	I
III^7	III^7	VI^7	VI^7
VI-	II^7	II-	V^7
I	II- V^7	I VI^7	II- V^7
I I^7	IV #IVo	I	II- V^7

Figure 5. Example of derivation of 'rhythm changes' generated by the grammar.

4. EXPERIMENTATION ON RHYTHM CHANGES

4.1 Evaluation of the grammar

In order to evaluate our phrase structure grammar, we generated 30 derivations of 'rhythm changes' with it. The generated multi-level progressions have been assessed by the musician, who helped creating the grammar and by another professional jazzman, who was not involved in the initial process and therefore analysed the generated multi-level progressions strictly from a musicology point of view. Figure 5 shows an example of generated 'rhythm changes'. Other derivations of 'rhythm changes' can be found online at `repmus.ircam.fr/dyci2/ressources`.

Every generated 'rhythm changes' has been validated by both musicians. However, these generated chord progressions have to be seen only as realisations of 'rhythm changes' for improvisation accompaniment. For automatic composition of a chord progression for a theme, some par-

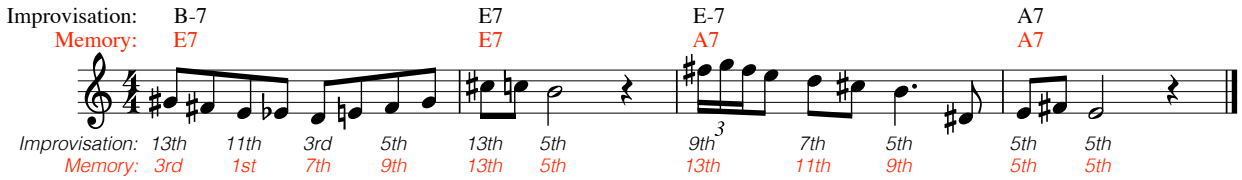

Figure 6. First four bars from the bridge of an improvisation on *An Oscar for Treadwell*. The 'Improvisation' lines shows the chord and degrees played in the generated improvisation; the 'Memory' lines shows the chord and degrees on which the different parts of the melody used to generate the improvisation were actually learnt. We see that when generating improvisations, we can reach places in the memory with a different chord label. The melody still makes sense in its continuity and harmonically because the multi-level labels from the memory and from the multi-level progressions are equivalent. The generated improvisations are therefore enriched with a new form of creativity.

allelism constraints assuring some symmetries between the different A section would be welcome. Moreover, this grammar covers the whole span of traditional bebop style 'rhythm changes' (on which it was trained on with the Omnibook corpus). No important possibilities were reported missing by any of the musicians. However, as expected, it does not create more modern versions of 'rhythm changes' such as the one of *The Eternal Triangle* by Sonny Stitt, or *Straight Ahead* by Lee Morgan, for instance. However, this is only due to the training corpus. No structural changes to the grammar would be needed for these. It would be interesting in the future to extend our 'rhythm changes' corpus to more varied chord progressions.

4.2 Improvisation on 'rhythm changes'

In order to test the benefits of using a multi-level progression, we generated some improvisations on Charlie Parker's music. First, using the phrase structure grammar trained on the 'rhythm changes' from the Omnibook, we generated multi-level progressions. On these progressions, we generated improvisations using two methods:

- the base generation model introduced in section 3.1. In this case, only the chord progression level of the multi-level progression was taken into account for generation.
- the extended generation model introduced in section 3.2. In this case, all the information from the multi-level progression was taken into account, i.e., the chord progression, the functional progression, and the sectional progression. A score was attributed to each level. We considered that for 'rhythm changes' the most important aspect was the functional aspect. Therefore, we attributed a weight of 0.5 to the functional level. We then attributed a weight of 0.3 to the chord level, and a weight of 0.2 to the sectional level.

In both cases, the content of the memory is a tune from Charlie Parker's Omnibook. Examples of generated improvisations (with both models) can be listened online at `repmus.ircam.fr/dyci2/ressources`.

We conducted a listening session with two professional jazz musicians to analyse the generated improvisations. First of all, the most significant difference seems to be that when using the multi-level progression, the improvisations are indeed better at following the structure. This can be especially noticed during the B section of the improvisations, where the guide notes and 5^{th} of each chord are more pronounced (cf. example on *An Oscar for Treadwell*). Figure 6 shows an excerpt from the bridge of an improvisation on *An Oscar for Treadwell*.
Second, when encountering chords that do not appear in the memory, the freedom provided by the multi-level progression (e.g., playing on a different chord as long as it shares the same functional role) enables the improvisation system to generate musical sentences with less fragmentation, creating a better sense of consistence and fluidity (cf. example on *Thriving from a Riff*). Moreover, this generates a form of creativity exemplified by playing musical sentences on a different chord than the original one, thus playing new guide notes or extensions adding colour to the improvisations, whilst keeping consistency (cf. example on *Anthropology*).

These results are promising and show how using a multi-level progression can result in significant improvement on how the improvisations are guided through the memory of the system.

5. CONCLUSIONS

We have shown how we can model the multi-level progression of a chord progression with a phrase structure grammar, and how to use this information for machine improvisation. First, this abstraction of a chord progression enables the system to generate several instances (or voicings) of this chord progression injecting some creativity in the chord progression generation whilst keeping its multi-level structure under control. Second, this additional musical information is used during the generation of a melodic improvisation; the generation process is able to take into account the global form of the scenario with the multi-level aspect to adapt to an ever-changing scenario and expand its possibilities. We applied this method on 'rhythm changes'

and constructed a phrase structure grammar for this type of chord progressions. This grammar was built and validated with musicians expertise. We then generated improvisations on multi-level progressions generated with the grammar (including chord progression, functionnal progression and sectional progression), using new navigation heuristics adapted to this multi-level information. The generated improvisations give promising results according to professional jazz musicians. The global form of the chord progression is respected, and the improvisations are still consistent, even on unmet chord progressions.
The 'rhythm changes' grammar was trained on a corpus of traditional bebop style 'rhythm changes'. It would be interesting to extend this corpus to more modern variations of 'rhythm changes' and to apply this method on other scenarios such as a blues structure or more abstract scenarios such as Steve Coleman's Rhythmic Cycles based on the lunation cycle [33]. It would also be interesting to see if this hierarchical structure could be extended to a higher level to take the organisation of an improvisation upon several successive occurences of the chord progression into account. Finally, it would also be interesting to extend this work into a system where not only the improvisation adapts to the generated progression, but the derivations generated by the grammar also take what is being improvised into account, or into a system with an automatic generation of the hierarchical grammar based on machine learning on a corpus.

Acknowledgments

This work is made with the support of the French National Research Agency, in the framework of the project DYCI2 "Creative Dynamics of Improvised Interaction" (ANR-14-CE24-0002-01), and with the support of Region Lorraine. We thank the professional jazzmen Pascal Mabit and Louis Bourhis for their musical inputs.

6. REFERENCES

[1] S. Dubnov, G. Assayag, and R. El-Yaniv, "Universal classification applied to musical sequences," in *Proceedings of the International Computer Music Conference*, 1998, pp. 332–340.

[2] V. Padilla and D. Conklin, "Statistical generation of two-voice florid counterpoint," in *Proceedings of the 13th Sound and Music Computing Conference*, 2016, pp. 380–387.

[3] R. P. Whorley and D. Conklin, "Music generation from statistical models of harmony," *Journal of New Music Research*, vol. 45, pp. 160–183, 2016.

[4] F. Pachet, "The Continuator : musical interaction with style," in *Proceedings of the International Computer Music Conference*, 2002, pp. 211–218.

[5] M. I. Bellgard and C. P. Tsang, "Harmonizing music the boltzmann way," in *Musical Networks*, N. Griffith and P. M. Todd, Eds. MIT Press, 1999, pp. 261–277.

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

[7] C. Allauzen, M. Crochemore, and M. Raffinot, "Factor oracle : a new structure for pattern matching," in *Proceedings of SOFSEM'99, Theory and Practice of Informatics*, 1999, pp. 291–306.

[8] A. Lefebvre and T. Lecroq, "Computing repeated factors with a factor oracle," in *Proceedings of the 11th Australasian Workshop On Combinatorial Algorithms*, 2000, pp. 145–158.

[9] G. Assayag and S. Dubnov, "Using factor oracles for machine improvisation," *Soft Computing*, vol. 8-9, pp. 604–610, 2004.

[10] G. Assayag and G. Bloch, "Navigating the oracle : a heuristic approach," in *Proceedings of the International Computer Music Conference*, 2007, pp. 405–412.

[11] G. Surges and S. Dubnov, "Feature selection and composition using PyOracle," in *Proceedings of the 2nd International Workshop on Musical Metacreation*, 2013, pp. 114–121.

[12] A. Einbond, D. Schwarz, R. Borghesi, and N. Schnell, "Introducing CatOracle: Corpus-based concatenative improvisation with the audio oracle algorithm," in *Proceedings of the 42nd International Computer Music Conference*, 2016, pp. 140–147.

[13] K. Déguernel, E. Vincent, and G. Assayag, "Using multidimensional sequences for improvisation in the OMax paradigm," in *Proceedings of the 13th Sound and Music Computing Conference*, 2016, pp. 117–122.

[14] D. Bailey, *Improvisation, its nature and practice in music*. Mootland Publishing, 1980.

[15] A. Donze, S. Libkind, S. A. Seshia, and D. Wessel, "Control improvisation with application to music," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-183, November 2013.

[16] J. Nika, M. Chemillier, and G. Assayag, "ImproteK : introducing scenarios into human-computer music improvisation," *ACM Computers in Entertainment*, vol. 4, no. 2, 2017.

[17] J. Nika, "Guiding human-computer music improvisation: introducing authoring and control with temporal scenarios," Ph.D. dissertation, UPMC - Université Paris 6 Pierre et Marie Curie, 2016.

[18] S. Koelsch, M. Rohrmeier, R. Torrecuso, and S. Jentschke, "Processing of hierarchical syntactic structure in music," *Proceedings of the National Academy of Sciences*, vol. 110, no. 38, pp. 15 443–15 448, 2013.

[19] M. Mongeau and D. Sankoff, "Comparison of musical sequences," *Computer and the Humanities*, vol. 24, pp. 161–175, 1990.

[20] M. Giraud, R. Groult, E. Leguy, and F. Levé, "Computational fugue analysis," *Computer Music Journal*, vol. 39, no. 2, 2015.

[21] M. Giraud, K. Déguernel, and E. Cambouroupoulos, "Fragmentations with pitch, rhythm and parallelism constraints for variation matching," in *Proceedings of the 10th International Symposium of Computer Music Multidisciplinary Research*, 2014, pp. 298–312.

[22] D. Tymoczko, *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford University Press, 2011.

[23] E. Chew, *Mathematical and Computational Modeling of Tonality*. Springer, 2014.

[24] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. MIT Press, 1983.

[25] N. Chomsky, *Studies on semantics in generative grammar*. Walter de Gruyter, 1996, vol. 107.

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

[27] M. Steedman, "A generative grammar for jazz chord sequences," *Music Perception*, vol. 2, no. 1, pp. 52–77, 1984.

[28] M. Chemillier, "Toward a formal study of jazz chord sequences generated by Steedman's grammar," *Soft Computing*, vol. 9, no. 8, pp. 617–622, 2004.

[29] M. Levine, *The jazz theory book*. Sher Music, 1995.

[30] C. Parker and J. Aebersold, *Charlie Parker Omnibook*. Alfred Music Publishing, 1978.

[31] M. Kaliakatsos-Papakostas, D. Makris, A. Zacharakis, C. Tsougras, and E. Cambouropoulos, "Learning and blending harmonies in the context of a melodic harmonisation assistant," *Journal of Creative Music Systems*, vol. 1, no. 1, 2016.

[32] G. Assayag, G. Bloch, M. Chemillier, A. Cont, and S. Dubnov, "OMax Brothers : a dynamic topology of agents for improvization learning," in *Workshop on Audio and Music Computing for Multimedia*, 2006.

[33] S. Coleman, "The lunation cycle as a point of departure for musical ideas," in *Arcana II : Musicians on Music*, J. Zorn, Ed., 2007, pp. 56–61.