THE UNREAL BOOK. ALGORITHMIC COMPOSITION FOR JAZZ LEAD SHEETS

Andrea VALLE (andrea.valle@unito.it) [1]

[1] *CIRMA/StudiUm*, **Università di Torino**, Italy

ABSTRACT

This paper presents the Unreal Book project which aims at exploring algorithmic generation of jazz lead sheets. Lead sheets are the standard notation format in jazz composition and are collected in many publications, the most relevant being the Real Book. Lead sheet format provides simple constraints (melody and chords) that allow for the application of algorithmic composition techniques based on the formalization of various jazz concepts. A computer-aided solution for the generation of a Real Book-like collection of lead sheets is presented, that takes also into account notation, including visual features that are considered defining of the Real Book. Seven examples of composition applications are shown, ranging from the implementation of jazz-inspired techniques to corpus-driven procedures.

1. BETWEEN COMPOSITION AND IMPROVISATION: LEAD SHEETS

Jazz has often been, and still is, largely described as an oral/aural practice, in which direct and mediated listening plays a pivotal role [1, 2]. This evidently holds true if one considers, respectively, the importance of musicians jamming together and of learning by imitation, and the relevance of recordings in the worldwide diffusion of jazz. Nevertheless, it has been observed that such an emphasis on oral/aural tradition has overshadowed the fundamental role that written sources in music notation have played since the early age of jazz, both in terms of organization of music structures (and thus performances) and diffusion of jazz repertoires [3]. The main written sources in jazz practice are lead sheets [4, 5]. Lead sheets are based on a notation format originating from classic American song. As they come from singing, they include three elements. The first two are the vocal melody, typically notated in the treble key, and the lyrics to be sung; the third is the harmonic background. Harmonies, i.e. chords, have been notated by means of a specific notation format, partly inherited from baroque abbreviated notation for basso continuo, but modified to take into account post-impressionistic harmonies [6], as shown in Figure 1. Chord sequences are known as "changes" in jazz practice. As jazz is mostly instrumental, most of the times lyrics have been dropped.

In short, the lead sheet notation format is a bare-bone one, including a single melody notated in common practice notion, with alphanumeric symbols on top representing chords. Apart from title and author, other typical features are an approximate indication tempo, and form abbreviations (da capo, section labels). An example is shown in Figure 2. Lead sheets are at the core of jazz practice as they mediate between composition and improvisation. On one side, they are written sources that provide input information to be taken into account by performers, like in classical Western composition. At the same time, many lead sheets are transcribed from recordings, so they do not share the same status of classical composed pieces, as they are rather scored a posteriori, even if the title and the author are referred to. On the other side, they are taken into account by performers as a starting material that is reorganized in various ways. Melodies can be modified by changing key, pitches and rhythm. Not only harmonies can be transposed to a new key, but they can also be radically altered, a standard practice known as "chord substitution" (see e.g. [8–10]). Form is only hinted at by lead sheets, typically showing the 32-bar form of the classic American song or the 12-bar form of blues: but these structures can be seen as starting points to be extended by intros, outros, soloing blocks, variable repetitions and reorganizations. Finally, lead sheets do not include arrangement features (orchestration), that are to be decided by the performer/arranger. To sum up, lead sheets, while still residing on the composition side, thanks to the openness of the format, propel a whole set of activities, leading to the final performance and steering to the improvisation side. This set of activities, placed in a crucial grey zone between composition and performance/improvisation, has been called "precomposition" [11]. Differently from written scores in Western classical music, lead sheets provide features and constraints that prompt, but not entirely determine, the final performance.

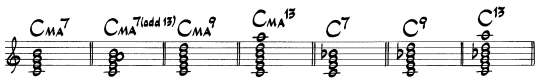

Figure 1. Some chords abbreviations and pitch content [7].

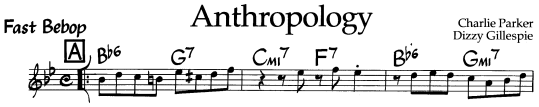

Figure 2. Beginning of *Anthropology*, from [7].

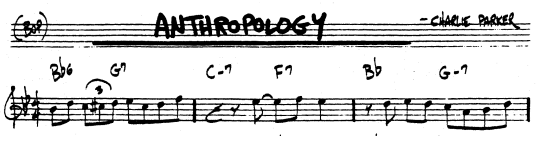

Figure 3. Beginning of *Anthropology*, from the *Real Book*.

2. FAKE, REAL, NEW REAL BOOKS

Lead sheets from famous jazz compositions have been shared in the jazz community, and soon they have been grouped into printed collections, reflecting the relevance in jazz practice of the repertoire of the so-called "standards" [12]. Printed collections of standards are typically named "fake books", a fake book being "a collection of charts or lead sheets used by jazz musicians (so-called because jazz musicians improvise, or "fake," their way through a performance)" [2, Appendix, p. 15]. The most famous fake book is the (pun intended) *Real Book*. The history of the latter is mostly unknown and goes unaddressed in all the recent histories of jazz. Apart some older pieces, it contains a selection of bebop and post-bop tunes, biased towards the late '60s-early '70s. The Real Book is a pirate assemblage of almost 5-hundreds, ad hoc handwritten transcriptions (including some well-known mistakes). Despite its obscure origins (it does not report any editorial data), the Real Book has gained a worldwide diffusion in the jazz community, first by means of photocopies, then in PDF format shared through p2p networks, thus becoming a de facto standard. The model is now at the core of jazz practice, as witnessed by the flowering of legal collections inspired by it (e.g. [7], Figure 2), including the authorized version by Hal Leonard of the very Real Book, providing the same cover, song list and typeface [13]. The Real Book handwritten typeface, while rooted into popular music arrangements, has become so iconic that it has been associated to the specific jazz flavour of music fonts as currently available in music notation software packages. Figure 2 is evidently inspired by Figure 3 from the original Real Book. The main features of the Real book are:

- dimension in order of 5-hundred of pieces;
- alphabetical indexing of the pieces;
- heterogeneous, even if not representative [3], sampling of jazz repertoire in terms of history, style, composition techniques;
- homogeneous notation format, based on lead sheet notation;
- hanwdritten "jazz" typeface.

3. THE UNREAL BOOK PROJECT

The Unreal Book Project is an algorithmic music composition project inspired by the Real Book. It aims at generating jazz-inspired compositions, notated in the lead sheet format, collected into a coordinated volume. The project focuses on three main objectives:

1. **Music composition**: as far as the author knows, no other project has focused on exploring the features and constraints of Real Book format in the context of algorithmic symbolic composition (i.e. resulting into music notation generation, see [14] rather than [15]). Algorithmic approaches to symbolic composition, while having a long and flourishing history, have never been applied to lead sheet generation [16, 17]. Of course, jazz has been extensively considered from an algorithmic approach, but mostly in terms of generative improvisation strategies (e.g. [18, 19]) and in relation to computational musicological analysis, including generative music theory [20] and corpus-based analysis [21, 22].
2. **Algorithmic generation of music notation**. Music notation generation is a complex issue, both from a theoretical perspective [23] and in terms of available, viable solutions related to specific projects. The Unreal Book project aims at investigating this issue in a broader sense, including all visual elements of the score;
3. **Formalization of jazz techniques**. Jazz theory has moved over the decades (in particular from the '60s) from a state of total absence (as it was implied into practices) to an abundant literature, related to the ongoing institutionalisation of jazz, discussing a variety of topics: mostly improvisation and harmony but also arrangement and practice routine. On a negative side, it has been observed that the quantitative increase in jazz pedagogy has led to a sort of homogenization. On a positive one, many resources are available, some detailing technical aspects, other discussing the latter in relation to historical developments, other proposing innovative approaches (see in general [1–3, 11]).

The Unreal Book project tries to match the Real Book's main features:

- dimension: the Unreal Book actually includes 102 pieces, notated in 1-page lead sheets (the most typical piece size in the Real Book). It is open to further expansion;
- heterogeneity: at the moment, seven different techniques have been used to compose lead sheets (see later), but the projects is meant as an open directory that allows to include other options and cross-hybridization among technical aspects from previously used techniques;
- visual consistency: not only the Unreal Book adopts the Real Book format, but it aims at mimicking its main visual features. Apart from cover design and TOC style, these include music notation typeface, page organization, title style. This aspect is not only related to visual aesthetics: rather, it helps musicians to enter a "Real Book mood" while approaching it. [1]

4. NOTATION ISSUES

In a symbolic algorithmic composition system, the score might be considered as the output of a terminal module that

[1] This is not at all irrelevant for music performance. For the very same reason, the legal Hal Leonard reissue of the Real Book [13] is marketed with the claim "You won't even notice the difference: [. . .] the covers and typeface look the same".

maps generated music data structures onto music notation symbols. From this perspective, music notation is meant as a form of data visualization. This encapsulation is not always possible. Historically, there is a feedback loop from notation to composition, as the former strongly constraints the latter [23]. In algorithmic composition practices there is a continuum going from integrated approaches, in which the whole pipeline from abstract music data processing to music notation is completely automated, to assisted or aided composition, in which the composer and the computer are loosely coupled – the computation agent providing elements (data structures, notation sketches) that are finally integrated into the score by the composer [24]. As in the Unreal Book project the notation format is explicitly given in advance, the most relevant issue is a technical one: for an output in the hundreds, it is mandatory to define at least partially automated strategies to include notation generation in the composition system. Many computer-aided composition packages (like OM [25], PWGL [26], Common Music [27], more recently Bach [28]) provide notation facilities that allow for drafting notation to be then finalized in a specialized environment. A viable option for totally automated notation generation in the style of the Real Book is the Lilypond notation software, a TEX-based language that compiles textual source files into PDF files [29]. Being text based, source files can be generated algorithmically. Lilypond source files can include not only melody notation but also chord symbols placed on top of it. Visual style can be adapted ingeniously so to reproduce all the main Real book visual items [2] . While elegant, this integrated solution does not allow an explorative approach to the results. As the output is a graphic file, there is no audio feedback associated to notation. MIDI commands can be included into Lilypond source but the resulting MIDI file must be open in a DAW environment with no interaction with the score. Moreover, MIDI does not support chord notation. Interactive features are instead typically provided by standard WYSIWYG notation environments such as Finale, Sibelius, Dorico, MuseScore. On the other side, these softwares use proprietary file formats. While not a notation format, MIDI has thus been used as a good compromise interchange solution (like in [22]). On one side, the target notation for standard lead sheets is metrically simple (resolution is limited to eight notes, rarely to 16ths, with only triplets as irregular groups) and can be accurately imported. On the other side, it is not possible to include chord symbols into MIDI, that must be consequently added by hand to the score, but they are comparatively sparse with respect to other notation symbols. The use of a WYSIWYG software allows for an immediate aural evaluation of results, and an eventual fine tuning of notation. MusicXML [30] is gaining momentum as an interchange format. A promising solution to be explored is to directly generate MusicXML code, integrating notes and chords, e.g. as allowed by the music21 toolkit [3] .

[2] See A. Lee, "Mimicking the Real Book", `http://leighverlag.blogspot.com/2015/12/mimicking-real-book-look.html`

[3] `https://web.mit.edu/music21/`

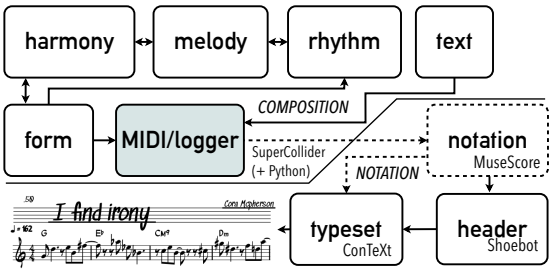

Figure 4. Overall system organization.

5. OVERVIEW OF THE SYSTEM

Figure 4 shows the overall software organization of the system. It can be divided into two subsystems: COMPOSITION and NOTATION. The COMPOSITION subsystem is developed in SuperCollider [31], and partially in Python (see later). It features four modules, meant as open libraries of functions for data processing and generation. Contents of the modules will be discussed in the next section in relation to examples. The rhythm module contains functions to generate metrically based events. The final format is always `[n, att, dur]`, where `n` is a placeholder to be filled with pitch information provided by the melody modules, and `att` and `dur` represent attack and duration, expressed in quarter beat units. The melody module includes various functions to generate pitch models: it outputs the same format of rhythm while replacing `n` proxies with actual values in MIDI notation. The harmony module is responsible for harmonic generation, that is, it outputs sequences of chords in standard alphanumerical notation. The form module coordinates the previous three modules. For each piece, it generates a melody and a harmony background. It also defines a general form in terms of sections (e.g. the classic American song form AABA) and takes care of handling section durations. The form module defines "composition configurations", that is, selections and parameterizations of functions from the various modules, including a generated bpm tempo. Logical ordering of the modules depends on the composition configuration (hence the double arrows between modules in Figure 4). The MIDI/logger module writes melodies into MIDI files, including tempo and key signature. When key signature is not decided in advance, it is inferred by comparing altered pitches in melody with various key signatures, and taking the signature that requires less momentary alterations. Chords cannot be included into a MIDI file, and are written by MIDI/logger into an easily human readable text file. The form module is typically used iteratively, feeding the MIDI/logger with 100 pieces in one shot, in relation to a certain composition configuration. Composition parameters are setup in in the SuperCollider/Python code. A crucial aspect of the Real Book is to be a collection of works written by many composers. The text module is devoted to generate names of composers and title of pieces. For each composition configuration, a batch of 5 composer names is generated: these groups of "composer" labels thus represent a specific

"style". Each composer is then associated with 20 titles. Composer names are generated from lists of most popular names (male and female) and surnames in the USA [4]. This is both a homage to the birth of jazz in New Orleans, and a good strategy to introduce variety, as USA have a largely differentiated linguistic community. Each title is generated by selecting one of the 50k movie reviews from the Large Movie Review Dataset [5], and then extracting and processing a textual fragment. The database has been chosen as it provides a colloquial tone typically associated with jazz titles, while paying a homage to cinema, an art form which has developed in 20th century, like jazz.

The NOTATION subsystem has the notation module at its core. The notation module is operated manually (hence the dashed contour in Figure 4), by interacting with the MuseScore notation editor. The latter is used to import MIDI files, and to edit them. MuseScore's importing options may be used to initially tune the notation. If occurring, editing involves notation aspects (e.g. F♯ might be replaced by G♭, da capo symbols might be inserted) but not music content. Chords are added by hand by taking as a reference the harmony log file relative to the MIDI file. The notation module is where the evaluation step takes place. Each piece, once imported is evaluated by the composer on the base of a set of loosely defined criteria, such as rhythmic complexity (pieces with too simple/complex rhythms are discarded), melodic contour (too static/variable melodies are discarded), melody/harmony clashing, etc. This manual filtering operation has resulted in an acceptance rate ranging between 10% and 20% of the generated pieces for the different composition configurations. While centered around acceptance and final notation of the pieces, the evaluation step provides also feedback on composition configurations, so that it can lead to various modification in the strategies on which these have been based. After acceptance and editing, pieces are exported in PDF files named with the format `composer_title`. The remaining part of the process is again fully automated. By using the Shoebot vector graphics package [6], a Python script is used by the header module to create automatically for each piece a graphic file containing the title, the composer and a progressive page number based on alphabetical ordering of the pieces, in the style of the Real Book. The header module also generates a source file for ConTEXt, a TEX-based typesetting document system [7]. The source file, including references to all files, is compiled into the final Unreal Book PDF. In terms of Computer-aided composition systems, the architecture can be thought as "fluid", i.e. made up of various modules "glued" together by two high level programming languages (SuperCollider and Python) [24].

6. SEVEN EXPERIMENTS

In the following, seven composition configurations of the previous system are described. These configurations, each of which associated with a set of composer labels, are meant as formalizations based on various features typically associated with jazz composition. Techniques are inspired by harmony/melody relationships as discussed in literature, mostly on improvisation [8]. In fact, there is a substantial permeability between composed melodies and improvised ones, as many times the latter are turned by jazz players into composed motives [11].

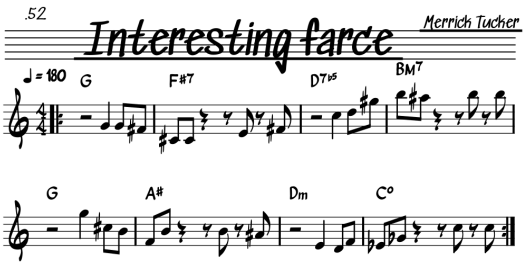

Figure 5. Atonal boppers example.

6.1 Atonal boppers

As the name suggests, Atonal boppers refers to both atonality and bebop. From the rhythmic point of view, bebop style results in less legato, with a distinctive presence of fast eight notes and less syncopations across bars [1]. In particular, Thelonious Monk's music is characterized by shifting riffs and accents, and isolated notes [6]. Atonal boppers are thus loosely inspired by these features. In this case, rhythm is handled by a drum machine-like pattern generator (contained in the rhythm module) that can be tuned so to generate blocks of a given metric duration with a certain density (i.e. average number of events for time unit). There are no slurs across bars. Figure 5 shows the section A of a piece. Here the rhythm pattern has a 2-bar duration, and is then repeated for the whole section. This repetition is crucial to ensure a certain degree of redundancy as the melody is freely atonal, a feature occurring in more experimental bebop pieces. In order to fill the rhythm pattern, the melody module exploits a Brownian generator: starting from a pitch and given a certain range in terms of semitones (e.g. ± 3), it generates a new pitch, then the process is reapplied. The Brownian model is interesting as it creates pitch contours. If a new pitch is outside a given overall boundary (substantially the treble staff, as it is customary), then it is flipped by 1 octave so to stay inside. The harmony module then provides automatic harmonization of the melody, in three steps. First, as the chords change at every bar, for each bar all the pitch classes in the melody are collected in a set, and stacked by thirds. Then, the resulting set is matched against a collection of given chords in a normalized form (e.g. major triad = [0,4,7]). Each chord is ranked in relation to how many pitches it is able to match in the pitch set, and the best is taken. Finally, in case more chords are available with the

[4] https://namecensus.com/

[5] https://ai.stanford.edu/~amaas/data/sentiment/

[6] http://shoebot.github.io/shoebot/

[7] https://wiki.contextgarden.net/Main_Page

[8] Code for the composition subsystems is available here https://github.com/vanderaalle/unrealBookComp

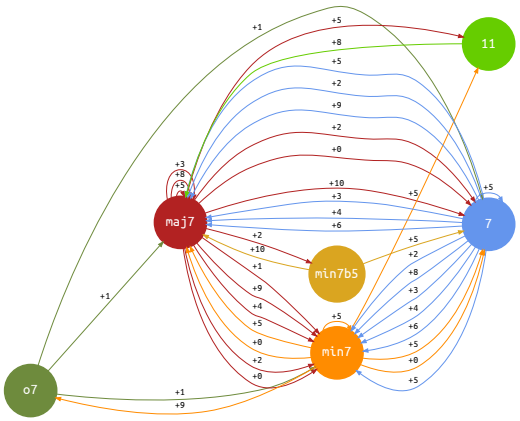

Figure 6. Harmonic graph.

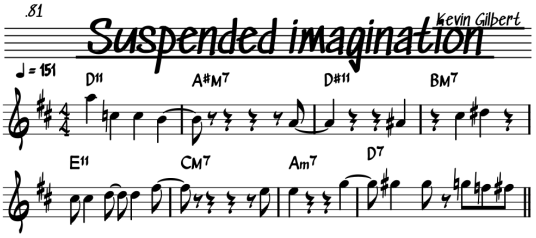

Figure 7. Forgetful harmonizers example.

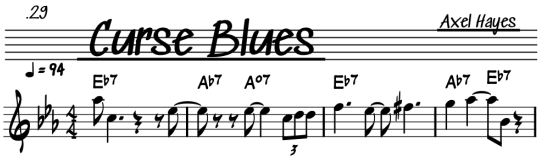

Figure 8. Bluesers example.

same score, one is selected on the base of a given priority ranking, that is: chord types have been ordered so that e.g. major chords have a higher rank than diminished ones. In this case, harmonies are limited to 7th chords (as can be seen in Figure 5). The form module randomly chooses among various section structures: AA, AB, ABA, ABAB, ABAC, and each section can have a length of 4, 6, 8, 12 bars. Figure 5 shows the A section of the piece, with a standard 8 bar length.

6.2 Forgetful harmonizers

Forgetful harmonizers are obtained by reversing some assumptions of Atonal boppers. Here, the rhythm module exploits a "time tape" model: time units in quarter notes are subdivided in 16th notes, and the latter are randomly grouped into longer events (triplets included). Then, some of the events are deleted according to a density parameter, thus creating rests. This process yields to irregular rhythmic groups. The relationship between melody and harmony is reversed, as the latter comes first in the generation process. Chosen chord types (mostly 7th chords) are organized in a graph in which each chord type (e.g. `min7b5`) is associated with a list of successors, based on [5] (Figure 6). As chords are taken into account, pitch classes, represented as integer in the range $[0...11]$, are at stake. Each edge is labeled with the number of positive semitones to reach the new pitch class root (e.g. +5 is the relative subdominant) from the chosen starting one. As chords are taken into account, the resulting pitch is to be taken $\mod 12$ (pitch class). In short, a cyclic graph results, that can be traversed randomly (and starting from a random vertex), each path representing a chord sequence once a starting pitch class is given. Actual pitches are inserted into the rhythmic pattern by looking at the relative harmony (again, one for each bar). A set of modes is available, and each chord type (e.g. `min7`) is associated with a subset of relative modes (e.g. minor, phrygian, dorian). A Brown process picks up a random pitch and selects a mode relative to the actual chord: if the pitch is in the mode, it is taken as is, else it is matched against the nearest one in the mode (e.g. in the context of Cmaj an E♭ becomes an E given a lydian mode). The next pitch is chosen randomly in a settable range around the previous pitch (e.g. ± 3 semitones), and again matched against one of the available modes for the actual chord. If a pitch duration extends across the bar, then the pitch is matched against the intersection of two modes, one for each chord. Form is organized as in the Atonal boppers case. This configuration is based on two assumptions. First, in jazz many times harmony comes first, not only in improvisation but also in composition, as in the cases when a given chord sequence is reused for a new piece (see the Rhythm Changers subsection). Second, since bebop there is a strong relation between harmonies and modes/scales (see [4] for an extreme application). In Forgetful harmonizers, harmonic sequences and harmony/mode relationships are based on [5]. Figure 7 shows an example, section A.

6.3 Bluesers

Bluesers are inspired by the classic 12-bar blues form. In this case, form is fixed a priori, and harmony is generated by reading an annotation file including typical 12-bar blues chord progessions [9]. Thus, in this case form and harmony are coupled and come first. Rhythm is again generated via the time tape model. As in the case of Forgetful harmonizers, melody matches harmonies, this time exploiting typical blues modes (e.g. including blues minor and bebop dominant scales [5]). Figure 8 shows an example with (matched) harmonies changing every half bar. As a side note, the typical "Blues" suffix has been appended to most titles.

6.4 Parkerians

Parkerians follow a very different path. Historically, the Charlie Parker Omnibook [32], a collection of 50 tran-

[9] E.g. `https://en.wikipedia.org/wiki/Twelve-bar_blues`

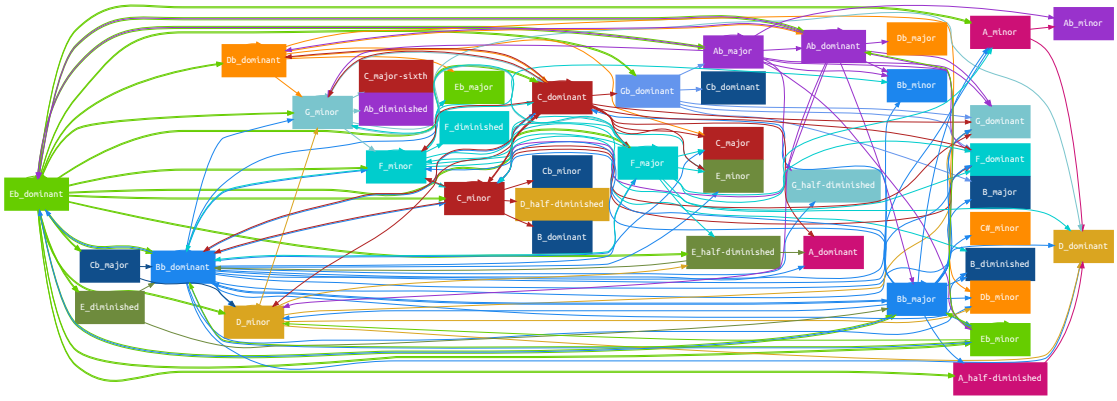

Figure 9. E♭ dominant harmonic subgraph from Charlie Parker's Omnibook.

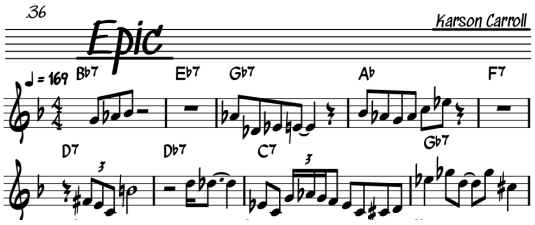

Figure 10. Parkerians example.

scriptions of Parker's solos, has been a crucial source of inspiration for generations of jazz players. The Omnibook has been annotated digitally [18] and is freely available in various formats (MIDI, MusicXML, MuseScore) [10]: the repository includes a Python script that allows to extract all the MusicXML data from each tune, that is, time-stamped melodies and chords. In Parkerians, following the model of Forgetful harmonizers, first a data structure has been created, in which each chord (in this case retaining the pitch class, e.g. B♭) is associated to all its chord successors in the whole corpus. The graph in Figure 9 is a subgraph of the overall, cyclic harmonic graph, showing for sake of readability only the 3-chord sequences starting with E♭ dominant. Enharmonic pitch classes are shown with the same color, regardless of their chord quality, and arrow colors are related to the starting pitch class. A second data structure is then generated that associates for each chord all the melodic fragments that are relative to it in the corpus. These two data structures bind together rhythm, melody and harmony. The form module selects a form among various options (AA, AB, ABA, ABAB, ABAC), and for each section a duration in bars (like in Atonal boppers). Then, for each section a chord sequence is obtained by traversing the harmonic graph. Finally, for each chord in the generated sequence, a melodic fragment is randomly choosen from those relative to the chord in the corpus. The whole set of compositions is thus a Markovian, harmony-driven recombination of Parker's melodic fragments, poured into a new form. An example is shown in Figure 10.

6.5 Modalists

Modalism in jazz typically refers to a variety of solutions based primarily on one or more modes at the basis of a piece. This usually results in a slower harmonic rhythm and in the use of one or more modes specified in advance as a pitch reservoir. A chord acts as a wider background for a mode-base melodic block. In Modalists, a set of modes is taken into account. Initially, a chord is generated from a degree in the chosen mode by stacking thirds. Then, the following chord is generated from a mode having at least one pitch in common with the starting one. Chords last 4 bars. As in Parkerians, rhythm and melody are extracted from existing sources. The Jazzomat project [11] makes freely available a large corpus of solo transcriptions, covering a variety of performers, styles, epochs [22]. The database can be explored by means of the MelospyGUI interactive tool, and selected items can be exported in MIDI. For Modalists, compositions from Jazzomat database have been chosen, converted into MIDI, re-edited so to simplify complex rubato rhythmic grouping, finally re-exported in MIDI. As solos are typically very dense, durations have been doubled. Then, all resulting melodies have been sliced into 1-measure-duration fragments. The recombining process is based on an enlarging reservoir. The reservoir is filled at initialization with a random fragment, then, every time a fragment is picked up, a new fragment is added to the reservoir. In this way, the reservoir keeps memory of the available fragments, leading to repeating patterns in the generated melody. Once selected, fragments are adapted to the mode occurring in their context, like in Forgetful harmonizers. In the example given in Figure 11, the same initial 2-bar melodic/rhythmic fragment is repeated thrice, then a new one is presented. Once the harmony changes at bar 5 from Fo7 to EM7, the recurring fragment is adapted to the different mode. The form module simply defines a variable number of 4-bar chord units.

[10] https://nubo.ircam.fr/index.php/s/BPtsmcqPQCNedgc

[11] https://jazzomat.hfm-weimar.de

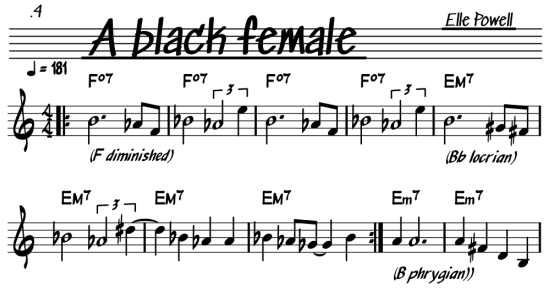

Figure 11. Modalists example.

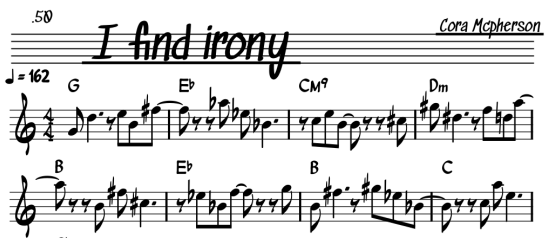

Figure 12. Minmaximalists example.

In terms of notation, modes are logged with harmony so to be reported –as customary with modal pieces– into the final score, as can be seen in Figure 11.

6.6 Minmaximalists

While rooted in blues and in post-impressionistic harmony, jazz has soon incorporated a variety of techniques developed in different contexts. Contemporary music techniques, above all serial and twelve-tone procedures, have been widely explored, in particular by the so-called Third Stream movement [2,3]. Inspired by this perspective, Minmaximalists are based on a serial, but not twelve-tone, technique. Variable length series between 3 and 6 elements are used for the pieces. Each integer item from the generated series is mapped onto a duration while some eight-note rest is added, thus defining the rhythmic pattern to be repeated. Analogously, a mapping strategy converts the same values from the series into pitches. At each repetition of the rhythmic pattern, the pitch series is then transposed following a random interval pattern. This organization results in a combination of rhythmic redundancy and melodic variety. In Figure 12 the rhythmic/melodic pattern has a duration of 6/4, so that two patterns fill 3 bars. Harmony advances at a regular 4/4 pace, thus providing a second rhythmic layer. Chords are obtained by adopting the same strategy used for Atonal boppers, but in this case harmonies are more complex, up to 9ths rather then 7ths. Coherently with the experimental assumptions, the form, while maintaining a standard overall duration of 32 bars, is not the classic AABA. Section A can be 8-bar, to be repeated (then AA) or a single 16 bar. Section A' is the same of A if the latter has a duration of 8 bars, or half A if the duration of A is 16 bars. The resulting pieces are atonal but the melody is anchored to jazz harmony, while rhythmically they are based on a straight 8-note rhythm, with a certain geometrical flavor, thanks to the eight-note based patterns shifting on the 4/4 bar grid. Hence the reference both to Minimalism and to its serial opposite.

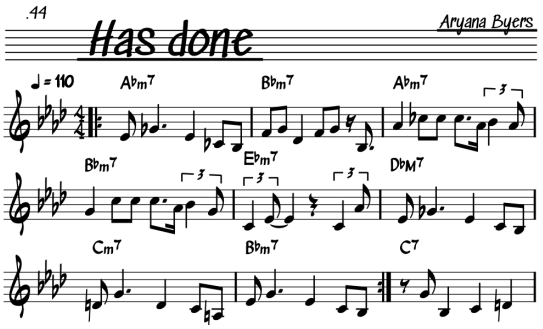

Figure 13. Rhythm changers example.

6.7 Rhythm changers

Some chord progressions are widely used in jazz. The most famous is derived from Gershwin' song *I Got Rhythm*. Known as Rhythm changes [5], it is at the base of many famous jazz pieces. Rhythm changes are thus taken into account as the starting point of the composition configuration. Thus, the form is the standard 32-bar AABA' from *I Got Rhythm*. In Rhythm changes, the harmonic rhythm proceeds at 2 chords per bar in the section A, while slowing down at 1 chord every 2 bars in section B. A common practice in jazz is to perform chord substitutions, that is, to replace original chords with new ones. This is seen as instrumental to offer new melodic possibilities. As discussed by Liebman [33], chord substitution can also be seen as a way to redefine harmony on the fly while improvising: a certain given chord is thus mentally replaced by the improviser with a different one, the latter acting as a reference for expanded melodic construction. In Rhythm changers, first, a second set of chords is defined for section A, which is an altered/complexified version of the original sequence: as an example, the starting maj7 chord is replaced by a min9. Section B is subject to an extensive chord substitution, following the so-called Coltrane changes, while, as a second step, it is altered/complexified as in A. Also, the harmonic rhythm is converted into 2 chords per bars, like in A. These richer A and B harmonies are used to create modes. For each bar, the union of the two chords' pitch sets defines a reference mode. The raw material from melody is created from Jazzomat, as described for Modalists. The fragments are matched onto the reference mode for their relative bar. While chord substitutions are used to create modes, they are not displayed in the score. Rather, the original harmony is displayed for section A, but only in the first chord of each bar. Analogously, section B displays the Coltrane changes but before the complexification step, and only the first chord of each bar. In short, harmonic complexification results in a surface harmonic slower and homogeneous rhythm, while feeding the melodic construction. An example (in A♭) is shown in Figure 13.

7. CONCLUSIONS AND FUTURE WORK

The Unreal Book project has proven to be an interesting test bench to develop an algorithmic-based, computer-aided composition system capable to integrate final typesetting by means of a "fluid" architecture. Performances of pieces are planned, so that results can be tested and properly evaluated in the context of jazz playing. The project can be expanded by implementing new composition configurations that may be triggered both by further investigations in jazz theory and analysis, and by various algorithmic composition processes. In particular, larger harmonic contexts could be taken into account to ensure harmonic structure, and more automated data extraction procedures could be implemented.

8. REFERENCES

[1] T. Gioia, *The History of Jazz*. Oxford: Oxford UP, 2011.

[2] S. Deveaux and G. Giddins, *Jazz*. New York: Norton, 2009.

[3] S. Zenni, *Storia del Jazz. Una prospettiva globale*. Viterbo: Stampa alternativa, 2012.

[4] M. Levine, *The Jazz Theory Book*. Petaluma: Sher Music Co., 1995.

[5] R. Rawlins and N. E. Bahha, *Jazzology*. Milwaukee: Hal Leonard, 2005.

[6] S. Zenni, *I segreti del Jazz*. Viterbo: Stampa alternativa, 2007.

[7] B. Bauer, Ed., *The New Real Book*. Petaluma: Sher Music Co., 1988.

[8] D. Haerle, *The Jazz Language*. Miami: Studio 224, 1980.

[9] A. Jaffe, *Jazz Harmony*. Tübingen: Advance Music, 1996.

[10] J. Mulholland and T. Hojnacki, *The Berklee of Jazz Harmony*. Boston: Berklee Press, 2013.

[11] P. F. Berliner, *Thinking in Jazz. The Infinite Art of Improvisation*. Chicago: Chicago UP, 1994.

[12] T. Gioia, *The Jazz Standards. A Guide to the Repertoire*. Oxford: Oxford UP, 2012.

[13] *The Real Book*. Hal Leonard, 2004.

[14] G. Nierhaus, Ed., *Patterns of Intuition*. Dordrecht: Springer, 2015.

[15] A. McLean and R. T. Dean, *The Oxford Handbook of Algorithmic Music*. New York: Oxford UP, 2018.

[16] C. Roads, *The Computer Music Tutorial*. Cambridge, MA, USA: MIT Press, 1996.

[17] G. Nierhaus, *Algorithmic Composition. Paradigms of Automated Music Generation*. Wien: Springer, 2009.

[18] K. Déguernel, E. Vincent, and G. Assayag, "Using Multidimensional Sequences For Improvisation In The OMax Paradigm," in *13th Sound and Music Computing Conference*, Hamburg, Germany, 2016.

[19] D. Vassilakis, A. Georgaki, and C. Anagnostopoulou, "'Jazz Mapping' an Analytical and Computational Approach to Jazz Improvisation," in *Proceedings of the 16th Sound and Music Computing Conference (SMC2019)*, 2019.

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

[21] Y. Broze and D. Shanahan, "Diachronic changes in jazz harmony: A cognitive perspective," *Music Perception*, vol. 31, no. 1, pp. 32–45, 2013.

[22] M. Pfleiderer, K. Frieler, J. Abeßer, W.-G. Zaddach, and B. Burkhart, Eds., *Inside the Jazzomat*. Mainz: Schott Campus, 2017.

[23] A. Valle, *Contemporary Music Notation. Semiotic and aesthetic aspects*. Berlin: Logos, 2018.

[24] ——, "Integrated Algorithmic Composition. Fluid Systems for including notation in music composition cycle," in *NIME 2008: Proceedings*, 2008, pp. 253–256.

[25] G. Assayag, C. Rueda, M. Laurson, C. Agon, and O. Delerue, "Computer-assisted composition at IRCAM: From PatchWork to OpenMusic," *Computer Music Journal*, vol. 23, no. 3, pp. 59–72, 1999.

[26] M. Kuuskankare and M. Laurson, "Expressive Notation Package," *Computer Music Journal*, vol. 30, no. 4, pp. 67–79, 2006.

[27] H. Taube, "An introduction to Common Music," *Computer Music Journal*, vol. 21, no. 1, pp. 29–34, 1997.

[28] A. Agostini and D. Ghisi, "A Max Library for Musical Notation and Computer-Aided Composition," *Computer Music Journal*, vol. 39, no. 2, pp. 11–27, 2015.

[29] H.-W. Nienhuys and J. Nieuwenhuizen, "LilyPond, a system for music engraving," in *Proceeding of the XIV CIM 2003*, Firenze, 2003, pp. 167–172.

[30] M. Good, "Lessons from the adoption of MusicXML as an interchange standard," in *XML 2006 Conference Proceedins*, D. Megginson, Ed., 2006.

[31] S. Wilson, D. Cottle, and N. Collins, Eds., *The SuperCollider Book*. Cambridge, Mass.: The MIT Press, 2011.

[32] J. Aebersold, Ed., *Charlie Parker Omnibook*. Atlantic, 1946.

[33] D. Liebman, *A Chromatic Approach to Jazz Harmony and Melody*. Advance Music, 2015.