Generating structured music for bagana using quality metrics based on Markov models

D. Herremans[a,*], S. Weisser[b], K. Sörensen[a], D. Conklin[c]

[a]*ANT/OR, University of Antwerp Operations Research Group, Antwerp, Belgium*
[b]*Université Libre de Bruxelles, Brussels, Belgium*
[c]*Department of Computer Science and Artificial Intelligence,*
University of the Basque Country UPV/EHU, San Sebastián, Spain
and IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

Abstract

In this research, a system is built that generates bagana music, a traditional lyre from Ethiopia, based on a first order Markov model. Due to the size of many datasets it is often only possible to get rich and reliable statistics for low order models, yet these do not handle structure very well and their output is often very repetitive. A first contribution of this paper is to propose a method that allows the enforcement of structure and repetition within music, thus handling long term coherence with a first order model. The second goal of this research is to explain and propose different ways in which low order Markov models can be used to build quality assessment metrics for an optimization algorithm. These are then implemented in a variable neighbourhood search algorithm that generates bagana music. The results are examined and thoroughly evaluated.

Keywords: Markov Models, Markov processes, Metaheuristics, Music, Bagana, Computer Aided Composition (CAC), Variable Neighborhood Search (VNS), Combinatorial Optimization
2000 MSC: 60J20, 60J22, 62P99, 90C27, 68T20, 90C59

[*]Corresponding author
Email addresses: `dorien.herremans@uantwerpen.be` (D. Herremans),
`stephanie.Weisser@ulb.ac.be` (S. Weisser),
`kenneth.sorensen@uantwerpen.be` (K. Sörensen), `darrell.conklin@ehu.es`
(D. Conklin)

1. Introduction

Music generation systems can be categorised into two main groups. On the one hand are the *probabilistic methods* (Xenakis, 1992; Conklin and Witten, 1995; Allan and Williams, 2005), and on the other hand are *optimization methods* such as constraint satisfaction (Truchet and Codognet, 2004) and metaheuristics such as evolutionary algorithms (Horner and Goldberg, 1991; Towsey et al., 2001), ant colony optimization (Geis and Middendorf, 2007) and variable neighbourhood search (VNS) (Herremans and Sorensen, 2013). The first group considers the solution space as a probability distribution, while the latter optimizes an objective function on a solution space. In this paper, we aim to bridge the gap between those approaches that consider music generation as an optimization system and those that generate based on a statistical model.

The advantage of composing music with optimization techniques is that they offer a way to impose structural constraints. The problem of automatically detecting structure and patterns in music has gained some attention, but remains a difficult task to solve (Conklin and Anagnostopoulou, 2001; Meredith et al., 2002; Collins, 2011). In this research we start from a given template structure and develop an efficient way of enforcing this structure with a variable neighbourhood search algorithm (see Section 2).

The main challenge when using an optimization system to compose music is how to determine the quality of the generated music. Some systems let a *human listener* specify how "good" the solution is on each iteration (Horowitz, 1994). GenJam, a system that composes monophonic jazz fragments given a chord progression, uses this approach (Biles, 2003). This type of objective function considerably slows down the algorithms (Tokui and Iba, 2000) and is known in the literature as the human fitness bottleneck.

Most automatic composition systems avoid this bottleneck by implementing an automatically calculated objective function based on either existing *rules from music theory* or by *learning from a corpus* of existing music. The first strategy has been used in compositional systems such as those of Geis and Middendorf (2007); Assayag et al. (1999) and Herremans and Sörensen (2013). Although every musical genre has its own rules, these are usually not explicitly available, which imposes huge limits on the applicability of this approach (Moore, 2001). This problem is overcome when style rules can be learned automatically from existing music, as is done in this research. This approach is more robust and expandable to other styles.

Markov models have been applied in a musical context, for learning from a

corpus, for a long time. The string quartet called the Illiac Suite was composed by Hiller and Isaacson in 1957 by using a rule based system that included probability distributions and Markov processes (see Sandred et al. 2009, for a recent overview of this work). Pinkerton (1956) learned first order Markov models based on pitches from a corpus of 39 simple nursery rhyme melodies, and used them to generate new melodies using a random walk method. Fred and Carolyn Attneave generated two perfectly convincing cowboy songs by performing a backward random walk on a first order transition matrix, as reported by Cohen (1962). Brooks et al. (1957) learned models up to order 8 from a corpus of 37 hymn tunes. A random process was used to synthesise new melodies from these models.

An interesting conclusion from this early work is that high order models tend to repeat a large part of the original corpus and that low order models seem very random. This conclusion was later supported by other researchers such as Moorer (1972), who states: "When higher order methods are used, we get back fragments of the pieces that were put in, even entire exact repetitions. When lower orders are used, we get little meaningful information out". These conclusions are based on a heuristic method whereby the pitch is still chosen based on its probability, but only accepted or not based on several heuristics which filter out, for instance, long sequences of non-tonic chords that might otherwise sound dull. Music compositions systems based on Markov chains need to find a balance in the order to use.

Other music generation research with Markov includes the work of Tipei (1975), who integrates Markov models in a larger compositional model. Xenakis (1992) uses Markov models to control the order of musical sections in his composition "Analogique A". Conklin and Witten (1995) present the multiple viewpoint method and apply it to probabilistic generation of chorale melodies. Markov models also form the basis for some real-time improvisation systems (Dubnov et al., 2003; Pachet, 2003; Assayag and Dubnov, 2004). Some more recent work involves the use of constraints for music generation using Markov models (Pachet and Roy, 2011). Allan and Williams (2005) trained hidden Markov models for harmonising Bach chorales. Whorley et al. (2013); Whorley and Conklin (2015) applied a Markov model based on the multiple viewpoint method to generate four-part harmonisations with random walk. A more complete overview of Markov models for music composition is given by Fernández and Vico (2013).

In an early survey of statistical models for music generation, Conklin (2003) highlighted the need for approaches to conserve structural patterns during generation, in order to effectively ensure intra-opus repetition. Collins et al. (2015) implemented this idea. They did not consider optimization to generate a solu-

tion, but used only a single random walk. The contributions of this research are twofold. First, we propose a method to generate music while conserving structural patterns during generation. Secondly, we propose and evaluate different ways in which machine learned models can be used to build quality evaluation metrics. To this end, a first order Markov model is built that quantifies note transition probabilities from a corpus of bagana music, a traditional lyre from Ethiopia. This model is then used to evaluate music with a certain repetition structure, generated by an optimization procedure previously developed by the authors (Herremans and Sörensen, 2012). Due to the size of many available corpora of music, including the bagana corpus used in this research, rich and reliable statistics are often only available for low order Markov models. Since these models do not handle structure and can produce very repetitive output, a method is proposed for handling long term coherence with a first order model. This method will also allow us to efficiently calculate the objective function, by using the minimal number of necessary note intervals as possible while still containing all information about the piece. Secondly, this paper will critically evaluate how Markov models can be used to construct evaluation metrics in an optimization context. In the next section more information is given about bagana music, followed by an explanation of the technique employed to generate repeated and cyclic patterns. An overview of the different methods by which a Markov model can be converted into an objective function are discussed in Section 3. Variable neighbourhood search, the optimization method used to generate bagana music, is then explained. An experiment is set up and the different evaluation metrics are compared in Section 5.

2. Structure and repetition in bagana music

The bagana is a ten-stringed box-lyre played by the Amhara, inhabitants of the Central and Northern part of Ethiopia. It is an intimate instrument, only accompanied by a singing voice, which is used to perform spiritual music. It is the only melodic instrument played exclusively for religious purposes (Weisser, 2012). The bagana melody and singing voice are quasi homophonic, meaning that the voice and bagana usually follow each other in unison (Weisser, 2005). In this research the focus is on analysing and generating the instrumental part.

The bagana is made of wooden pillars and soundbox, equipped with ten cattle gut strings. The strings are plucked with the left hand and four strings are used as finger rests. It is tuned to an Amhara traditional pentatonic scale. Each finger of the left hand is assigned to one string (see Figure 1), except in the case of the index finger (referred to as finger 2 and $2'$ in the figure), which plays two equally

tuned strings. This allows us to make abstraction from the actual pitch and work with the corpus made by Conklin and Weisser (2014) based on finger numbers (see Section 5).

Figure 1: Assignment of fingers to strings on the bagana
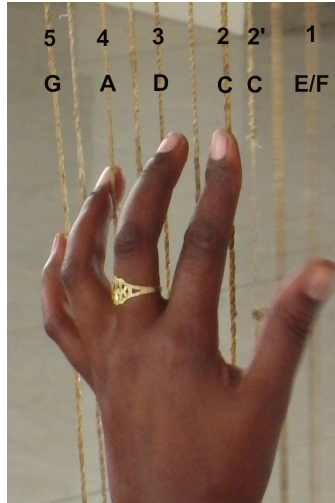
Bagana songs are typically very repetitive with a recognisable overall structure (Weisser, 2006). This repetition is intentional since repetitive music has a strong influence on the state of consciousness among musical traditions. Even Western-trained listeners describe the sounds as "becoming meditative objects, relaxing the mind" (Dennis, 1974).

Figure 2: Tew Semagn Hagere by Alemu Aga, as transcribed by Weisser (2005)

Figure 3: Yibelahala by Alemu Aga, as transcribed by Weisser (2005)

5

Two example bagana pieces, including finger numberings, are displayed in Figure 2 and 3. Both pieces consist of two sections, and only a few segments (A^1, A^2 and A^3) are used, and repeated many times throughout the duration of the piece. Additionally, both pieces contain a segment (A^2) that is repeated within different sections of the piece. In what follows, an approach is described for respecting this structure and repetition within new sequences generated from Markov models.

Since repetition is so important for bagana music, cycles and repetitions must be represented and evaluated in an efficient way. Markov models alone are incapable of representing such structures, which can involve arbitrarily long-range dependencies, and therefore the approach used here is to preserve the structure and repetition provided by an existing template piece. The next subsections will describe a method for representing and efficiently evaluating this structure and repetition while still employing a Markov model to generate the basic musical material.

2.1. Cycles and patterns

Following the theoretical approach of Angluin (1980), the structure of a bagana piece may be represented using a *pattern*, which is a sequence of variables drawn from a set V (we use A^1, A^2, ... as variables). Given a set ξ of *event* symbols (in the case of bagana, finger numbers), a *realization* of a pattern is a substitution from V to $\xi^\star$ (the set of all sequences formed from event symbols), mapping variables to sequences of finger numbers. Each variable is also associated with a *length*, that is, a constraint on the length of the sequence that can replace the variable. The event sequence replacing a variable A^i, associated with a length e, will be notated in this paper as $a_1^i a_2^i \ldots a_e^i$.

To represent repetition of entire sections, the notion of *cycles* and *cyclic patterns* is introduced. A *cycle* is a sequence of events that is repeated any number of times. For example, in the bagana song of Figure 2, the two cycles are the event sequences labelled by $A^1 A^2$ and $A^3 A^2$. Cycles can be abstracted and represented as *cyclic patterns*, which are patterns as described above but now enclosed in the symbols $\|{:}$ and $:\|$. For example, in the bagana song of Figure 2, the two cyclic patterns are $\|{:}\ A^1 A^2\ {:}\|$ and $\|{:}\ A^3 A^2\ {:}\|$.

Patterns can also be concatenated, forming compound patterns. Taking the bagana song of Figure 2 as an example, the pattern describing this piece is finally represented as the compound pattern:

$$\|{:}\ A^1 A^2\ {:}\|\|{:}\ A^3 A^2\ {:}\| \tag{1}$$

6

with the lengths of A^1, A^2, A^3 being specified as 6, 6, and 13, respectively. The corpus used in this research (see Section 5.1) has been annotated with these repetition structures by a bagana expert.

2.2. Realizing and evaluating cyclic patterns

A *realization* of a pattern is a mapping from variables of the pattern onto actual event sequences (i.e., sequences of finger numbers). The event sequences represented by any one variable are generated using a Markov model and the entire generation is given by replicating the instances of the same variable. In order to properly generate music that contains cyclic patterns, traditional statistical sampling methods like random walk are not suited because long-range dependencies cannot be captured by a Markov process. Therefore, we use a local search optimization technique to generate the actual event sequences, which allows us to implement the cyclic structure as a hard constraint. These realizations of the patterns are given to the objective function in order to assess the quality of a generated fragment.

In order to reduce the number of transition matrix lookups, without losing any information about the sequence, an expansion technique was developed to generate the minimal extended subsequence that can be used to calculate the objective function. For example, consider a cycle $A = a_1 a_2 \ldots a_k$ that is repeated n times in the template piece. When calculating the objective function, we should take care not to omit the sequence $a_k a_1$, which is the transition that is heard whenever the cycle is repeated. Since calculating the objective function on A alone is not sufficient, we could simply calculate it on the full sequence as it is played, but this would require roughly n times more transition matrix lookups than required. The expanded sequence A' will simply contain an additional element, which represents the transition from end to beginning: $A' = a_1 a_2 \ldots a_k a_1$. The expansion method used in this research reduces the number of lookups while retaining all the information of individual transitions.

2.3. Compound cyclic patterns

Bagana music is characterised by a large number of repetitions combined together. The expansion method discussed in the previous subsection is applied to reduce the number of transition matrix lookups. This method keeps the minimum number of transitions without forgetting the connections between the end and beginning of a cycle, as discussed in the subsection above. For a compound

pattern which contains cycles, some care needs to be taken to exclude certain transitions. For example, for the cyclic pattern described by Equation 1, the sequence on which the objective function is calculated thus becomes:

$$A^1 A^2 a_1^1 \bigtriangledown a_e^2 A^3 A^2 a_1^3 \qquad (2)$$

whereby A^i consists of the note sequence $a_1^i a_2^i \ldots a_e^i$ and the $\bigtriangledown$ represents a break point between the sequences.

The structure of the second piece, Yibelahala, can be described with the same method:

$$A^1 A^2 a_1^1 \bigtriangledown a_e^2 A^1 A^3 A^2 \qquad (3)$$

This method as described above is valid for first order evaluation. When an evaluation metric is based on note sequences of more than two subsequent notes (e.g., unwords, see Section 3.5), *higher order* expansion is necessary. In the case of a metric that evaluates sequences of length 3, second order expansion is necessary, and the cyclic pattern described by Equation 1 becomes:

$$A^1 A^2 a_1^1 a_2^1 \bigtriangledown a_{e-1}^2 a_e^2 A^3 A^2 a_1^3 a_2^3 \qquad (4)$$

where as before the $\bigtriangledown$ represents a break point. In the next section, different methods of using of Markov models to construct quality metrics for an optimization algorithm are explained.

3. Using Markov models within evaluation metrics

Markov models describe the note transition probabilities of a musical piece or style. These transition probabilities can be used to evaluate the quality of a musical piece. Farbood and Schoner (2001) use dynamic programming to find the highest probability sequence of notes in a counterpoint line given a cantus firmus. They used both manually created Markov models (based on music theory rules) and models learned from a corpus of 44 examples. A high probability or maximum likelihood approach is also explored by Lo and Lucas (2006) as a fitness function for a genetic algorithm when generating melodies, based on a corpus of 282 pieces. They conclude that high probability sequences sound uninteresting due to the large amount of oscillation between just two notes. Davismoon and Eccles (2010) use a different quality measure. They do not try to maximize the likelihood, but rather minimize the distance between the transition matrices (both of the original model and the newly generated piece) with simulated annealing.

8

In the next subsections, different methods that might be used as quality assessment from a Markov model are described. The first three quality assesment metrics can be used alone. The latter two are constraining metrics that are implemented in combination with one of the first three. These techniques will be implemented and thoroughly evaluated in Section 5.

3.1. High probability sequences (XE)

Farbood and Schoner (2001) and Lo and Lucas (2006) generate the maximum probability sequence from a statistical model. It makes intuitive sense that this type of sequence is preferred, yet there might be more to a good musical piece than just maximizing the probability (e.g., variety). This will be evaluated in Section 5.

Cross-entropy is used as a measure for high probability sequences, whereby minimal cross-entropy corresponds to a maximum likelihood sequence according to the model. The probability $P(s)$ of a fragment s consisting of a sequence of notes $e_1, e_2, \ldots, e_\ell$ is transformed into cross-entropy (Manning and Schutze, 1999). The cross-entropy is the mean of the *information content h_i* of each event e_i in the sequence:

$$h_i = -\log_2 P(e_i \mid e_{i-1}) \tag{5}$$

$$f(s) = \frac{1}{\ell - 1} \sum_{i=2}^{\ell} h_i \tag{6}$$

The quality of a musical fragment is thus evaluated according to the cross-entropy (average negative log probability) of the fragment computed using the dyad transitions of the transition matrix. This forms the objective function $f(s)$ that should be minimized.

3.2. Minimal distance between TM of model and solution (DI)

Davismoon and Eccles (2010) use an evaluation metric that tries to match the transition matrices of both the original model and the newly generated piece by minimizing the Euclidean distance between them. This will ensure that they have an equal distribution of probabilities after each possible note. The metric used in this paper is based on Davismoon and Eccles (2010) and can be formulated as follows for an $N \times N$ transition matrix:

9

$$f(s) = \frac{1}{N} \sqrt{\sum_{a \in \xi} \sum_{b \in \xi} \left(P(b \mid a) - \bar{P}(b \mid a) \right)^2} \tag{7}$$

where ξ is the set of event symbols, for example in the bagana the finger numbers, $P(b \mid a)$ is the model transition probability from a to b, and $\bar{P}(b \mid a)$ is the transition probability calculated from the new piece.

It is expected that this measure enforces more variety in the generated music, as the overall probability transition distribution is optimized to resemble the one of the corpus. The musical output of the VNS that minimizes this metric as its objective function will be evaluated in the experiment in Section 5.

3.3. Delta cross-entropy (DE)

In Subsection 3.1 cross-entropy was minimized to find the maximum likelihood sequence. It cannot be guaranteed that this is a sequence a listener would enjoy. If we look at the corpus, there are proportionally fewer pieces with cross-entropy below the average value. Figure 4 shows a histogram of the cross-entropy data calculated with leave-one-out cross-validation from the corpus used in the experiment of Section 5. That is, every piece was left out of the corpus, the model retrained, and the cross-entropy of that piece was computed according to the model. It is clear from this figure that most pieces are not close to the lowest entropy value that occurs in the corpus. As the results in Section 5 will indicate, the single minimal cross-entropy sequence can be very repetitive. Optimizing to the average cross-entropy value E might offer a solution for this.

When optimizing towards the average cross-entropy value, the function being minimized thus becomes:

$$f(s) = \left| E - \frac{1}{\ell - 1} \sum_{i=2}^{\ell} h_i \right| \tag{8}$$

where E is the average cross-entropy of the corpus.

3.4. Information contour (i)

One of the problems mentioned by Lo and Lucas (2006) with high probability sequences is that they often sound uninteresting and repetitive. More diversity might be achieved by defining the "information contour" within a piece. Information contour is a measure that describes the movement of information between two successive events (up indicating less expected than the previous event, down

10

Figure 4: Histogram of cross-entropy values of the corpus
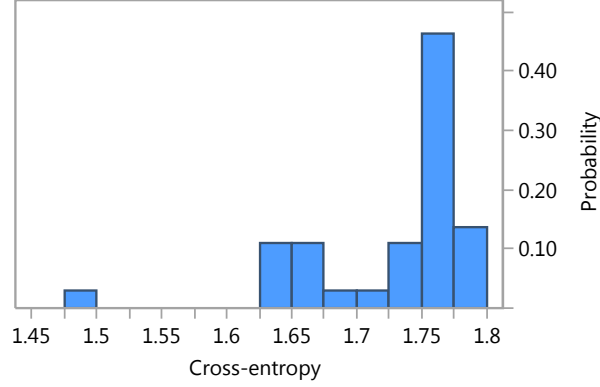
indicating more expected than the previous event). It can be seen as the contour of the information flow, which has been used by Witten et al. (1994) and Potter et al. (2007) to measure information dynamics in a musical analysis. In order to measure this a viewpoint is created that expresses if the information content, with respect to a model of the corpus that does not include the template piece for each event, is higher, lower, or equal to that of the previous event. The *information contour* $C(e_i)$ of event e_i is defined as:

$$
C(e_i) = \begin{cases} \text{up} & \text{when } h_i > h_{i-1} \\ \text{same} & \text{when } h_i = h_{i-1} \\ \text{down} & \text{when } h_i < h_{i-1} \end{cases}
$$

In the experiment performed in Section 5, the information contour was calculated for each note transition of a selected template song (Tew Semagn Hagere and Yibelahala). When evaluating a new solution, a similar information contour may be desirable. Therefore, the objective function to be minimized can be specified as follows for a piece of ℓ notes:

$$
f(s) = M_c \times \sum_{i=2}^{\ell} x_i \tag{9}
$$

(4, 2, 1)	(2, 1, 2)	(4, 1, 1)	(1, 4, 1)	(3, 4, 1)	(1, 4, 3)	(2, 4, 1)
(1, 4, 4)	(3, 2, 1)	(2, 1, 1)	(4, 1, 4)	(4, 1, 2)	(1, 2, 5)	(4, 5, 2)

Table 1: The set of unwords that were found in the bagana corpus

$$\text{whereby} \begin{cases} x_i = 1 & \text{when } C(e_i) \text{ is not the same as in the template} \\ x_i = 0 & \text{when } C(e_i) \text{ is the same as in the template} \end{cases}$$

and M_c is an arbitrarily high number.

This metric will be tested in conjunction with the first three metrics by summing the objective functions. By using the arbitrarily high number M_c in the equation above, optimizing the information part will have priority over the other term of the objective function (low entropy, minimize TM distance, delta cross-entropy).

3.5. Unwords (u)

While music contains patterns that are repeated, it equally contains rare patterns. Conklin (2013a) identified *antipatterns*, i.e., significantly rare patterns, from a corpus of Basque folk music and from the corpus of bagana music used in this research (Conklin and Weisser, 2014). A related category of rare patterns are those of *unwords*. Herold et al. (2008), in their paper on genome research, first suggested this term for the shortest words from the underlying alphabet that do not show up in a given sequence. Unwords are thus defined as the shortest sequence of notes (i.e., not contained within a longer unword) that never occur in the corpus. Among these words, we filter for those that are statistically significant. This results in a list of words whose absence from the corpus is surprising given their letter statistics (Conklin and Weisser, 2014). These patterns may represent structural constraints of a style.

A related approach to improve the music generated by simple Markov models is by adding constraints on the subsequences that can be generated. For example, Papadopoulos et al. (2014) efficiently avoid all subsequences greater than a specified maximum order k, for the purpose of avoiding simple regeneration of long fragments identical to the corpus. A contrasting approach to this problem is to constrain the types of short words that can be generated based on the analysis of a corpus, i.e., unwords, rather than uniformly forbidding all words of a specified length or greater.

To find unwords, the algorithm of Conklin and Weisser (2014) was used to efficiently search the space of bagana finger patterns for significant unwords. Table 1 lists the resulting set of 14 unwords. These unwords, all trigrams, are all formed from one or more bigrams that were identified as antipatterns by Conklin and Weisser (2014). To use these for evaluating music, their occurrence is given a penalty according to the following formula:

$$f(s) = M_w \times u \tag{10}$$

whereby M_w is an arbitrarily high number and u is the total number of unwords counted in the piece.

This quality measure can be seen more as a hard constraint since unwords never occur in the original corpus. Therefore it is combined with the first three techniques from this section in the experiment. This is done by summing the objective functions for both techniques. The use of an arbitrarily high number M_w will again give priority to the removal of the unwords over the other metric with which this is combined (low entropy, minimize TM distance, delta cross-entropy).

4. Variable neighbourhood search

This paper uses an optimization technique whereby the best possible combination of notes needs to be found to fit a certain style, whilst constraining long term coherence. A bridge between sampling from statistical models and optimizing according to an objective function is made by comparing different quality measures. The resulting problem is a combinatorial optimization problem which is computationally complex due to the exponential number of possible solutions. A variable neighbourhood search algorithm (VNS) is implemented as it is an efficient optimization method that is used in many more traditional optimization areas including (capacitated) vehicle routing (Kytöjoki et al., 2007), graph colouring (Avanthay et al., 2003) and project scheduling (Fleszar and Hindi, 2004). Hansen et al. (2001) find that VNS can outperform existing heuristics in terms of both computing time and solution quality for several problems.

A VNS for generating counterpoint based on formal rules from music theory is developed and implemented by the authors (Herremans and Sörensen, 2012). In later work, this algorithm has been modified to generate high probability sequences (Herremans et al., 2014). In this paper, different evaluation metrics are implemented and the obtained results are discussed.

Variable neighbourhood search, or VNS, is a local search based metaheuristic. The structure of the implemented VNS is represented in Figure 5. The VNS starts

Figure 5: Overview of the VNS.
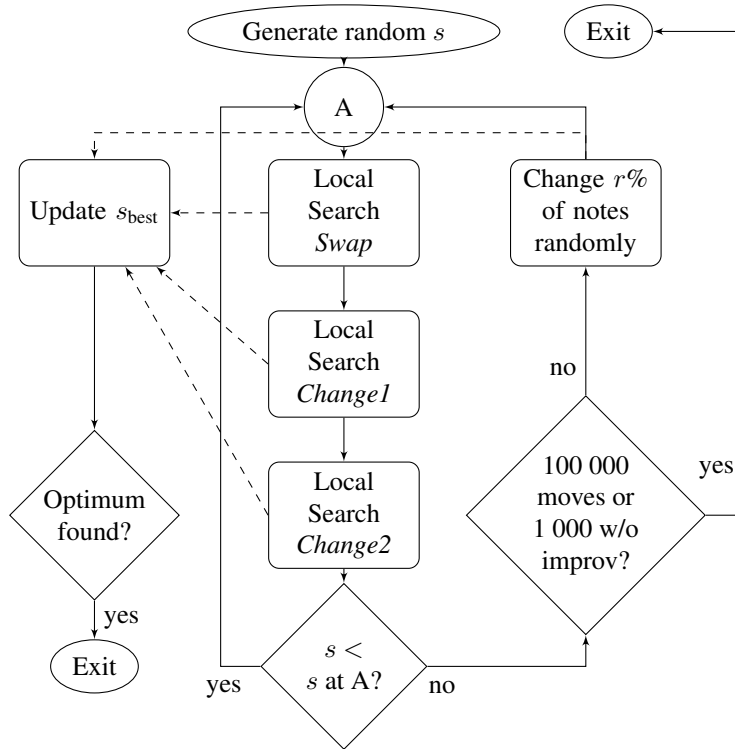
from an initial fragment that has random pitches. From this starting fragment the algorithm iteratively makes small improvements (called *moves*) in order to find a better one, i.e., a fragment with a lower value for the objective function. Three different move types are defined to form the different neighbourhoods that the algorithm uses. The first move type swaps the top notes of a pair of dyads (*swap*). The *change1* move changes any one pitch to any other allowed pitch. The last move, *change2*, is an extension of the previous one whereby two sequential pitch are changed simultaneously to all possible allowed pitches.

The *neighbourhood* is the set of all possible fragments s' that can be reached from the *current* fragment by a move type. Infeasible solutions are excluded from the neighbourhood. The first note is fixed to an A and the last note is fixed to a C. Solutions who do not comply with this hard constraint are considered infeasible. The local search uses a steepest descent strategy, whereby the best fragment is selected from the entire neighbourhood. This strategy will quickly steer the algo-

14

Figure 6: Comparison of VNS and Random Search for minimizing cross-entropy on the Tew Semagn Hagere template structure
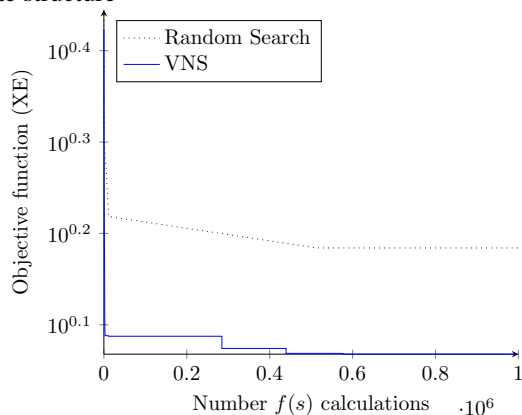
rithm away from choosing fragments with zero probability dyads, but it does not strictly forbid them (transitions with zero probability are set to an arbitrarily high cross-entropy). A tabu list is also kept, to prevent the local search from getting trapped in cycles.

When no improving fragment can be found by any of the move types, the search has reached a local optimum. A *perturbation* strategy is implemented to allow the search to continue and escape the local optimum (Hansen and Mladenović, 2003). This perturbation move changes the pitch of a fixed percentage of notes randomly. The size of the random perturbation as well as the size of the tabu lists and other parameters were set to the optimum values resulting from a full factorial experiment on first species counterpoint (Herremans and Sörensen, 2012). The VNS algorithm was implemented in C++ and the source code is available online[2].

The developed VNS was compared to a Random Search, whereby fingers are assigned to each note randomly, in order to confirm its efficiency. Figure 6 shows the evolution of the VNS versus the Random Search algorithm when minimizing cross-entropy on the Tew Semagn Hagere structure. We let both algorithms run until the objective function was calculated 1,000,000 times. The VNS quickly outperforms the Random Search. The best solution found by the Random Search is a lot worse that the best solution found by the VNS.

---

[2]http://antor.ua.ac.be/musicvns

5. Results

An experiment was set up in order to compare the outcome of the different evaluation techniques discussed in section 3. They were all implemented in the objective function of the VNS described in the previous section. The algorithm stopped after performing 100 000 moves or when no improving solution was found after 1 000 moves.

5.1. Training data and Markov model

The corpus used in this experiment is described in more detail by Conklin and Weisser (2014). It consists of 37 pieces of bagana music that have been recorded by Weisser (2005) between 2002 and 2005 in Ethiopia (except for two of them recorded in Washington DC). The songs consist of a relatively short melody, repeated several times with different lyrics, except for the refrain. The entire corpus has been annotated with the repetition structures described in Section 2 by a bagana expert.

Two pieces were selected from the corpus as structural template pieces. The first one is a piece called Tew Semagn Hagere by Alemu Aga (see Figure2). This piece is usually learned by bagana students at a relatively early stage and was selected because of its regular rhythm and simplicity (no ornamentations). Furthermore, this piece has a very repetitive structure, yet the repeated patterns do not dominate the other patterns and are fairly equally weighted. A different piece called Yibelahala by Alemu Aga (see Figure 3) was chosen as a second template structure. This piece has similar properties and is usually taught to bagana students just before the Tew Semagn Hagere piece. For both templates, the rhythm within the patterns was kept fixed. The evaluation method based on information contour described in Section 3 needs a template to calculate the target information contour. The same templates were also used to illustrate the global structure discussed in Section 2.3.

The output of the algorithm was rendered in the tezeta scale (Conklin and Weisser, 2014) using F for finger 1 (see Figure 1) with a bagana soundfont and presented to one of the authors, a bagana expert, who evaluated the fragments discussed in Section 5.2. Her comments on a preliminary experiment resulted in some improvements of the algorithm, including the fixation of the first note to an A (finger 4) for both template songs and the last note to a C (finger 2) for Tew Semagn Hagere and an F (finger 1) for Yibelahala. The results were then presented again for evaluation.

	2 (C)	3 (D)	1 (F)	5 (G)	4 (A)
2 (C)	0.291	0.263	0.015	0.040	0.390
3 (D)	0.238	0.039	0.694	0.018	0.011
1 (F)	0.029	0.330	0.237	0.357	0.047
5 (G)	0.049	0.032	0.401	0.153	0.366
4 (A)	0.502	0.005	0.005	0.281	0.206

Table 2: Transition matrix based on the bagana corpus; finger numbers as indices, and corresponding pitch class names (Tezeta scale) in brackets

A first order Markov model was learned from the corpus of bagana music. First order models can be weak models, as also stated by Lo and Lucas (2006). Yet in some cases there is not enough data to generate a higher order model, as in the case of the bagana corpus. Working with a first order model allows training on a small corpus, and also gives us a very clear overview of the effects of the different metrics, without having to look at more complicated second order patterns. The resulting transition matrix is represented in Table 2.

5.2. Musical results

The VNS algorithm was run with the different metrics from Section 3 as its objective function on both template pieces. The first three metrics were run independently. Then each of these metrics was combined with unwords and information contour. For each metric, the evaluation of cross-entropy and the distance of the transition matrices is shown over time for the Tew Semagn Hagere template in Figure 7. The average cross-entropy value E (see Section 3.3) of the corpus is also displayed on the plots in this figure as a reference value. The musical output corresponding to each of the runs visualised in Figure 7 is displayed in AppendixA and AppendixB. These music sheets were presented to the bagana expert for evaluation together with the rendered audio files. Table 3 shows that the generated music is different from the template piece, where similarity is measured as the percentage of notes that are the same in both the generated piece and the template piece. When measuring the similarity notes at the same position within the given structure were compared.

High probability sequences (XE)

Fragment 1 in AppendixA and Fragment 10 AppendixB show the output of minimizing the cross-entropy with the VNS. As also found by Lo and Lucas
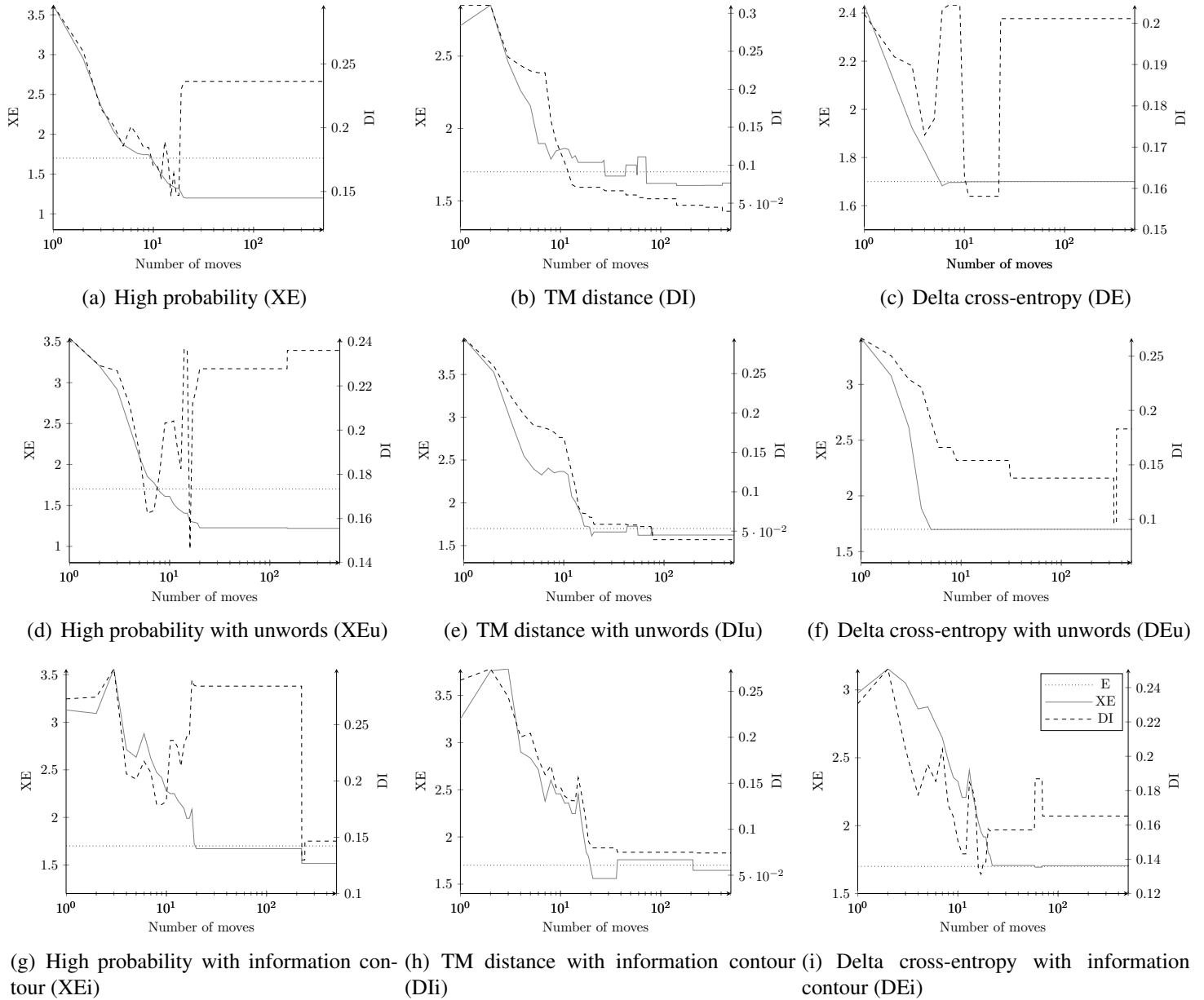
(a) High probability (XE)

(b) TM distance (DI)

(c) Delta cross-entropy (DE)

(d) High probability with unwords (XEu)

(e) TM distance with unwords (DIu)

(f) Delta cross-entropy with unwords (DEu)

(g) High probability with information contour (XEi)

(h) TM distance with information contour (DIi)

(i) Delta cross-entropy with information contour (DEi)

Figure 7: Evolution of cross-entropy and distance of transition matrices over time for the Tew Semagn Hagere template structure

18

	tmpl.	XE	DI	DE	XEu	DIu	DEu	XEi	DIi	DEi
Similarity (%)	T	29	36	26	23	29	52	48	36	48
	Y	17	40	30	20	27	30	20	26	26
Cover of range (%)	T	100	100	100	100	100	100	100	100	100
	Y	100	100	100	60	100	100	100	100	100
Number of unwords	T	0	0	0	0	0	0	0	1	0
	Y	2	0	0	0	0	0	0	0	1

Table 3: General characteristics of the generated music displayed in AppendixA (T) and AppendixB (Y)

(2006), the minimal cross-entropy sequence can be very repetitive. According to the transition matrix, the finger transitions corresponding to the note sequences A–C, C–A, F–D and D–F are indeed high probability transitions, still the global result is not the one a listener would enjoy as there is a lot of oscillation. The model generates two high probability transition loops (A–C and D–F). Figure 7(a) confirms that minimizing the cross-entropy using VNS causes a rapid decrease in cross-entropy. This is similar to the experiment done by the authors with first species counterpoint (Herremans et al., 2014), where it was shown that VNS is an efficient method for generating high probability sequences, and that VNS rapidly converges to the minimum cross-entropy sequence. It is also noticeable from Figure 7(a) that optimizing with the XE metric does not cause a decrease in the DI metric, but rather undirected movement.

Minimal distance between TM of model and solution (DI)

When minimizing the distance between the transition matrices of the model and a generated solution with VNS, we again see a rapid decrease in this metric in Figure 7(b). The cross entropy measure converts to the average cross-entropy value. This means that by minimizing the DI metric, the cross-entropy moves toward the average value. The music generated music is not too repetitive and the expert listener considered the fragment (Fragment 2 in AppendixA) to be very good. Fragment 11 (AppendixB) was considered as "interesting", which a good number of common patterns and not too many big jumps, however, the expert identified a transition (A–A) that sounds unusual given the rhythm. Since the model does not currently take rhythm int account, this has brought up an interesting topic for future improvements to the statistical model.

Delta cross-entropy (DE)

The average of the 37 cross-entropy values, calculated with leave-one-out cross-validation as described in Section 3.3, in the bagana corpus is $E = 1.7$. The algorithm is able to reach the average cross-entropy value quickly (Figure 7(c)). The DI metric is not constrained during DE minimization, and changes randomly throughout the generation process. This is an interesting observation, as minimizing the DI metric in the previous section did constrain both the DI metric to the minimum and the cross-entropy to the average value. This means that optimizing with the DI metric is stronger, more constrained, than solely with the DE metric as it seems to constrain two metrics. One of the resulting fragments (Fragment 3 in AppendixA) was described by the expert as "not easy to sing with". The other generated fragment (Fragment 12 in AppendixB) was considered as very interesting, with an interesting singing melody. This opposing result could be due to the fact that, as described above, this measure is not very constraining, and thus allows a wide range of solutions.

Unwords (u)

When minimizing the number of unwords together with the three previously discussed metrics, the evolution of the algorithm is very similar (Figures 7(d), 7(e) and 7(f)). This is probably due to the fact that unwords sometimes occur when using the other techniques (see Table 3), yet they do not dominate. The high probability sequence still has a lot of repetitions, though slightly decreased.

The expert found the sequence generated with the DEu metric (Fragment 6 in AppendixA) very good, with the remark that a player would rather play A–G–F–D in segment A^3 instead of A–F–D. This comment is supported by the higher transition probability A–G and G–F versus A–F. The DEu metric optimizes towards the average cross-entropy of the corpus, thus not always preferring the highest probability transitions. Similarly, Fragment 15 (AppendixB) was labeled as a good results, with the same remark that musicians would prefer an alternanace of A–C instead of C–C in A^3. This proposed sequence again has a higher transition probability. The expert also found the segment A^3 generated by the DIu metric (Fragment 5 in AppendixA and Fragment 14 in AppendixB) respectively very good and good. The results with the XEu metric (Fragment 4 in AppendixA and Fragment 13 in AppendixB) are less good as they are both very repetitive.

Information contour (i)

Constraining the information contour together with the first three metrics discussed seems to have a positive influence on the quality of the generated music.

When minimizing the cross-entropy, it forces the music out of the high probability loops and thus prevents oscillation. This results in a much more varied music (Fragment 7 and 16 in Appendix). The plots in Figures 7(g), 7(h) and 7(i) have a similar evolution as before.

The expert found the piece generated with the XEi metric (Fragment 7 in AppendixA) very good. The second piece generated with this method was labeled as "ok", due to some octave-range issues for singing. Both pieces have enough variation and not too much repetition, which is an improvement from generating with the XE or XEu metric.

The result generated with the DIi metric (Fragment 8 in AppendixA) was considered very good, with exception of segment A^3 which has some issue with the combination of rhythm and pitch. This is an interesting issue that the authors hope to address in future research by building a statistical model with takes both duration and pitch into account. The result based on the second template generated with this method (Fragment 17 in AppendixB) was considered as an interesting result by the expert, with the same remark about rhythm/pitch combination.

The piece generated with the DEi metric was considered as good music, with the remark that a player would rather play C–D–F–D instead of C–F–D. Similarly as in the above section, C–D and D–F have much higher transition probabilities than C–F. This can again be explained because the algorithm that was run with the DEi metric (Fragment 9 in AppendixA) optimizes towards the average cross-entropy of the corpus instead of the lowest cross-entropy. The piece generated with the Yibelahala template and the DEi metric has a good amount of repetitions and enough variation, yet the expert found them difficult to sing with (requires a wide range).

6. Conclusions

The results of the experiments conducted in this paper show that there is no single best metric to use in the objective function. Minimizing cross-entropy can lead to oscillating music, a problem which was corrected by combining this metric with information contour. Minimizing the distance between the transition matrix of the model and the generated music also outputs more varied music and seems to constrain the entropy to the average entropy of the corpus. This relationship is not valid in the opposite direction. By constraining the cross-entropy to the average value, the DI metric is not minimized. Optimizing with the DI metric is thus more constraining then optimizing solely with the DE metric. The bagana expert found that generating with the DI metric produces good musical results.

21

The cross-entropy, TM distance minimization and delta cross-entropy metric all produce good outcomes when combined with information contour. Forbidding the occurrence of unwords in the solution when combined with XE is not enough to avoid oscillations, because in fact even in the corpus extended oscillations do occur, hence they are not significant unwords.

While the comments of the bagana expert are very positive, one possible improvement would be to model and generate into a more complex template with more cyclic patterns. This can equally be handled by the approach used in this paper simply by specifying an alternative pattern structure for the template piece. It would also be interesting to build a statistical model with takes both note duration and pitch into account. This would address some of the comments of the bagana expert concerning the combination of certain notes with durations.

There are other techniques besides those mentioned above that could be used to improve and measure musical quality of music generated based on a Markov model. One option would be to look at a multiple viewpoint system (Conklin and Witten, 1995; Conklin, 2013b) that includes a viewpoint which models the coherence within finger number sequences. This is already partly implemented on a high level by generating into a certain fixed structure. Another possible idea would be to relax the unwords metric to include antipatterns, i.e., patterns that do occur, but only rarely.

All of the metrics above are based on models created from an entire corpus. Conklin and Witten (1995) additionally consider short term models for which the transition matrix is recalculated based on the newly generated music. This is done for each event, based on the notes before it. This metric might enforce even more diversity as it stimulates repetition and the creation of patterns. This interesting approach is left for future research.

Contrary to methods usually used to sample from Markov models, such as random walk and Gibbs sampling, the VNS algorithm allows us to specify a wide variety of constraints. VNS is a powerful method that is proven to be efficient in many fields (Yazdani et al., 2010; Adibi et al., 2010; Kuo and Wang, 2012). Whenever a neighbourhood is generated, the solutions that do not satisfy these constraints are excluded. This simple mechanism allows the user to implement many types of constraints, ranging from fixing the pitch of certain notes, to forbidding repetition and only allowing certain pitches. Another interesting expansion of this work would be to apply pattern discovery to automatically detect structural patterns, which serve as constraints for the VNS algorithm.

In this research, a method is developed that allows the enforcement of a structure and repetition within the music using the VNS algorithm previously devel-

oped by Herremans and Sörensen (2013), thus ensuring long term coherence. The second contribution of the paper is obtained by proposing different ways to construct evaluation metrics based on a Markov model while enforcing a structure. These metrics are used to evaluate generated bagana music in an optimization procedure. The advantage of using learned models for the evaluation of models compared to more traditional rule based techniques (Geis and Middendorf, 2007; Delgado et al., 2009; López-Ortega, 2013) is that there is no restriction to a particular style. The evaluation is robust as models can be learned from any database of music to compose pieces in the style of a corpus. Experiments show that integrating techniques such as information flow, optimizing delta cross-entropy, TM distance minimization and others improve the quality of the generated music based on low order Markov models. The previously developed VNS algorithm for music generation (Herremans and Sörensen, 2013) did not generate into structure and did not use an automatically learned objective function. Both problems have been tackled in this research. The methods developed and applied in this paper were applied to music for the Ethiopian bagana and should be applicable to a wide range of musical styles.

7. Acknowledgments

8. References

Adibi, M., Zandieh, M., Amiri, M., 2010. Multi-objective scheduling of dynamic job shop using variable neighborhood search. Expert Systems with Applications 37 (1), 282–287.

Allan, M., Williams, C. K. I., 2005. Harmonising chorales by probabilistic inference. Advances in neural information processing systems 17, 25–32.

Angluin, D., 1980. Finding patterns common to a set of strings. Journal of Computer and System Sciences 21, 46–62.

Assayag, G., Dubnov, S., 2004. Using factor oracles for machine improvisation. Soft Computing 8 (9), 604–610.

Assayag, G., Rueda, C., Laurson, M., Agon, C., Delerue, O., 1999. Computer-assisted composition at IRCAM: from PatchWork to OpenMusic. Computer Music Journal 23 (3), 59–72.

Avanthay, C., Hertz, A., Zufferey, N., 2003. A variable neighborhood search for graph coloring. European Journal of Operational Research 151 (2), 379–388.

Biles, J. A., 2003. Genjam in perspective: A tentative taxonomy for GA music and art systems. Leonardo 36 (1), 43–45.

Brooks, F. P., Hopkins, A. L., Neumann, P. G., Wright, W. V., 1957. An experiment in musical composition. IRE Transactions on Electronic Computers (3), 175–182.

Cohen, J. E., 1962. Information theory and music. Behavioral Science 7 (2), 137–163.

Collins, T., 2011. Improved methods for pattern discovery in music, with applications in automated stylistic composition. Ph.D. thesis, The Open University.

Collins, T., Laney, R., Willis, A., Garthwaite, P. H., 2015. Developing and evaluating computational models of musical style. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 1–28.

Conklin, D., 2003. Music generation from statistical models. In: Proceedings of the AISB Symposium on Artificial Intelligence and Creativity in the Arts and Sciences. Aberystwyth, Wales, pp. 30–35.

Conklin, D., 2013a. Antipattern discovery in folk tunes. Journal of New Music Research 42 (2), 161–169.

Conklin, D., 2013b. Multiple viewpoint systems for music classification. Journal of New Music Research 42 (1), 19–26.

Conklin, D., Anagnostopoulou, C., 2001. Representation and discovery of multiple viewpoint patterns. In: Proceedings of the International Computer Music Conference. Havana, pp. 479–485.

Conklin, D., Weisser, S., 2014. Antipattern discovery in Ethiopian bagana songs. In: Proceedings of 17th International Conference on Discovery Science, October 8-10. Bled, Slovenia, pp. 62–72.

Conklin, D., Witten, I., 1995. Multiple viewpoint systems for music prediction. Journal of New Music Research 24 (1), 51–73.

Davismoon, S., Eccles, J., 2010. Combining musical constraints with Markov transition probabilities to improve the generation of creative musical structures. In: Applications of Evolutionary Computation. Springer, pp. 361–370.

Delgado, M., Fajardo, W., Molina-Solana, M., 2009. Inmamusys: Intelligent multiagent music system. Expert Systems with Applications 36 (3), 4574–4580.

Dennis, B., 1974. Repetitive and systemic music. The Musical Times, 1036–1038.

Dubnov, S., Assayag, G., Lartillot, O., Bejerano, G., 2003. Using machine-learning methods for musical style modeling. IEEE Computer 36 (10), 73–80.

Farbood, M., Schoner, B., 2001. Analysis and synthesis of Palestrina-style counterpoint using Markov chains. In: Proceedings of the International Computer Music Conference ICMA. San Francisco, pp. 111–117.

Fernández, J. D., Vico, F., 2013. AI methods in algorithmic composition: A comprehensive survey. Journal of Artificial Intelligence Research 48, 513–582.

Fleszar, K., Hindi, K. S., 2004. Solving the resource-constrained project scheduling problem by a variable neighbourhood search. European Journal of Operational Research 155 (2), 402–413.

Geis, M., Middendorf, M., 2007. An ant colony optimizer for melody creation with Baroque harmony. In: IEEE Congress on Evolutionary Computation. pp. 461–468.

Hansen, P., Mladenović, N., 2003. Variable neighborhood search. In: Handbook of Metaheuristics. Vol. 57 of International Series in Operations Research & Management Science. Springer, US, pp. 145–184.

Hansen, P., Mladenović, N., Perez-Britos, D., 2001. Variable neighborhood decomposition search. Journal of Heuristics 7 (4), 335–350.

Herold, J., Kurtz, S., Giegerich, R., 2008. Efficient computation of absent words in genomic sequences. BMC Bioinformatics 9 (1), 167.

Herremans, D., Sörensen, K., 2012. Composing first species counterpoint with a variable neighbourhood search algorithm. Journal of Mathematics and the Arts 6 (4), 169–189.

Herremans, D., Sörensen, K., 2013. Composing fifth species counterpoint music with a variable neighborhood search algorithm. Expert Systems with Applications 40 (16), 6427–6437.

Herremans, D., Sorensen, K., April 2013. FuX, an android app that generates counterpoint. In: 2013 IEEE Symposium on Computational Intelligence for Creativity and Affective Computing (CICAC). pp. 48–55.

Herremans, D., Sörensen, K., Conklin, D., 2014. Sampling the extrema from statistical models of music with variable neighbourhood search. In: Proceedings of ICMC–SMC, Athens, Greece. pp. 1096–1103.

Horner, A., Goldberg, D. E., 1991. Genetic algorithms and computer-assisted music composition. Urbana 51 (61801), 437–441.

Horowitz, D., 1994. Generating rhythms with genetic algorithms. In: Proceedings of the International Computer Music Conference. San Francisco: International Computer Music Association, pp. 142–143.

Kuo, Y., Wang, C.-C., 2012. A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. Expert Systems with Applications 39 (8), 6949–6954.

Kytöjoki, J., Nuortio, T., Bräysy, O., Gendreau, M., 2007. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. Computers & Operations Research 34 (9), 2743–2757.

Lo, M. Y., Lucas, S. M., 2006. Evolving musical sequences with n-gram based trainable fitness functions. In: IEEE Congress on Evolutionary Computation, 2006 (CEC 2006). IEEE, pp. 601–608.

López-Ortega, O., 2013. Computer-assisted creativity: Emulation of cognitive processes on a multi-agent system. Expert Systems with Applications 40 (9), 3459–3470.

Manning, C., Schutze, H., 1999. Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA.

Meredith, D., Lemström, K., Wiggins, G., 2002. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. Journal of New Music Research 31 (4), 321–345.

Moore, A. F., 2001. Categorical conventions in music discourse: Style and genre. Music & Letters 82 (3), 432–442.

Moorer, J. A., 1972. Music and computer composition. Communications of the ACM 15 (2), 104–113.

Pachet, F., 2003. The Continuator: musical interaction with style. Journal of New Music Research 32 (3), 333–341.

Pachet, F., Roy, P., 2011. Markov constraints: steerable generation of Markov sequences. Constraints 16 (2), 148–172.

Papadopoulos, A., Roy, P., Pachet, F., 2014. Avoiding plagiarism in Markov sequence generation. In: Proceedings of AAAI 2014. Quebec, pp. 2731–2737.

Pinkerton, R. C., 1956. Information theory and melody. Scientific American 194 (2), 77–86.

Potter, K., Wiggins, G. A., Pearce, M. T., 2007. Towards greater objectivity in music theory: Information-dynamic analysis of minimalist music. Musicae Scientiae 11 (2), 295–324.

Sandred, O., Laurson, M., Kuuskankare, M., 2009. Revisiting the Illiac Suite – a rule-based approach to stochastic processes. Sonic Ideas/Ideas Sonicas 2, 42–46.

Tipei, S., 1975. MP1: a computer program for music composition. In: Proceedings of the Annual Music Computation Conference, Univ. of Illinois, Urbana, Illinois. pp. 68–82.

Tokui, N., Iba, H., 2000. Music composition with interactive evolutionary computation. In: Proceedings of the Third International Conference on Generative Art. Milan, pp. 215–226.

Towsey, M. W., Brown, A. R., Wright, S. K., Diederich, J., 2001. Towards melodic extension using genetic algorithms. Educational Technology & Society 4 (2), 54–65.

Truchet, C., Codognet, P., 2004. Musical constraint satisfaction problems solved with adaptive search. Soft Computing-A Fusion of Foundations, Methodologies and Applications 8 (9), 633–640.

Weisser, S., 2005. Etude ethnomusicologique du bagana, lyre d'Ethiopie / ethno-musicological study of the bagana lyre from ethiopia. Ph.D. thesis, Universite Libre de Bruxelles.

Weisser, S., 2006. The Ethiopian lyre bagana: An instrument for emotion. In: Proceedings of the 9th International Conference on Music Perception and Cognition. pp. 376–382.

Weisser, S., 2012. Emotion and music: The ethiopian lyre bagana. Musicae Scientiae 16 (1), 3–18.

Whorley, R. P., Conklin, D., 2015. Improved iterative random walk for four-part harmonisation. In: MCM 2015: International Conference on Mathematics and Computation in Music. London.

Whorley, R. P., Wiggins, G. A., Rhodes, C., Pearce, M. T., 2013. Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonization problem. Journal of New Music Research 42 (3), 237–266.

Witten, I. H., Manzara, L. C., Conklin, D., 1994. Comparing human and computational models of music prediction. Computer Music Journal 18 (1), 70–80.

Xenakis, I., 1992. Formalized Music: Thought and mathematics in composition. No. 6. Pendragon Pr.

Yazdani, M., Amiri, M., Zandieh, M., 2010. Flexible job-shop scheduling with parallel variable neighborhood search algorithm. Expert Systems with Applications 37 (1), 678–687.

AppendixA. Musical output based on Tew Semagn Hagere template structure

Fragment 1: High probability (XE)

Fragment 2: TM distance (DI)

Fragment 3: Delta cross-entropy (DE)

Fragment 4: High probability and unwords (XEu)

Fragment 5: TM distance and unwords (DIu)

Fragment 6: Delta cross-entropy and unwords (DEu)

Fragment 7: High probability and information contour (XEi)

Fragment 8: TM distance and information contour (DIi)

29

Fragment 9: Delta cross-entropy and information contour (DEi)

AppendixB. Musical output based on Yibelahala template structure

Fragment 10: High probability (XE)

Fragment 11: TM distance (DI)

Fragment 12: Delta cross-entropy (DE)

Fragment 13: High probability and unwords (XEu)

Fragment 14: TM distance and unwords (DIu)

Fragment 15: Delta cross-entropy and unwords (DEu)

Fragment 16: High probability and information contour (XEi)

Fragment 17: TM distance and information contour (DIi)

Fragment 18: Delta cross-entropy and information contour (DEi)