Sampling the extrema from statistical models of music with variable neighbourhood search.

Herremans, D; Sörensen, K; Conklin, D

Sampling the extrema from statistical models of music with variable neighbourhood search

Dorien Herremans[1] **Kenneth Sörensen**[1] **Darrell Conklin**[2,3]
[1] ANT/OR - University of Antwerp Operations Research Group
University of Antwerp, Belgium
[2] Department of Computer Science and Artificial Intelligence
University of the Basque Country UPV/EHU, San Sebastián, Spain
[3] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
dorien.herremans@uantwerpen.be
kenneth.sorensen@uantwerpen.be
darrell.conklin@ehu.es

ABSTRACT

Statistical models of music can be used for classification and prediction tasks as well as for generating music. There are several different techniques to generate music from a statistical model, but not all are able to effectively explore the higher probability extrema of the distribution of sequences. In this paper, the vertical viewpoints method is used to learn a Markov Model of abstract features from an existing corpus of music. This model is incorporated in the objective function of a variable neighbourhood search method. The resulting system is extensively tested and compared to two popular sampling algorithms such as Gibbs sampling and random walk. The variable neighbourhood search algorithm previously worked with predefined style rules from music theory. In this work it has been made more versatile by using automatically learned rules, while maintaining its efficiency.

1. INTRODUCTION

Ever since the very first computer was created, the idea of using this device to generate music has existed. Even Ada Lovelace, the world's first conceptual programmer who worked together with Charles Babbage on the Difference Engine and Analytical Engine [18], hinted at using computers for automated composition around 1840:

> *"[The Engine's] operating mechanism might act upon other things besides numbers [. . .] Supposing, for instance, that the fundamental relations of pitched sounds in the signs of harmony and of musical composition were susceptible of such expressions and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent."* – [5]

Since then many researchers have worked on automatic composition systems, both for melody harmonization (i.e., finding the most musically suitable accompaniment to a given melody) [32], generating a melodic line to a given chord sequence or cantus firmus [22] and even generating a full musical piece from scratch [34].

An important difference between the various music generation systems is the actual method used to generate the music. This ranges from probabilistic methods and rule-based systems [1, 11, 39], to constraint satisfaction methods [37] and the use of metaheuristics such as evolutionary algorithms [25, 36], ant colony optimization [17] and variable neighbourhood search (VNS) [24]. For a more complete overview of existing automatic composition systems, the reader is referred to Herremans and Sörensen [23].

A second way to differentiate between music generation systems is the way in which they determine the quality of the generated music. One method is to have a *human listener* determine how "good" the solution is. GenJam, a genetic algorithm that composes monophonic jazz fragments given a fixed chord progression, uses this approach [4]. The solution quality is not coded in the algorithm, but feedback is given by a human mentor for each population member individually. This causes a delay known as the human fitness bottleneck and places a non-negligible psychological burden on the listener [35].

To circumvent this bottleneck, most systems automatically assess the quality of a musical fragment. This can be done based on existing *rules from music theory* or by *learning from a corpus* of existing musical pieces. The first strategy has been applied in automatic composition systems such as those by Geis and Middendorf [17], Assayag et al. [2] and Donnelly and Sheppard [13]. An obvious disadvantage is that the rules of the chosen musical style need to be formally written down. Although every musical genre has its own rules, these are generally not explicitly available [28]. Therefore, it is useful to automatically learn style rules from existing music. The second method can be considered as being more robust and expandable to other styles. David Cope's Experiments in Musical Intelligence (EMI) extract signatures of musical pieces using pattern matching with a grammar based system to understand a specific composer's style [31]. Xenakis [39] uses Markov Models to control the order of musical sections in his composition "Analogique A". Markov models have also been used to generate Palestrina counterpoint based on a given cantus firmus with dynamic programming [15]. Allan and Williams [1] trained hidden Markov models for harmonising Bach Chorales and Whorley et al. [38] applied a Markov model based on the multiple viewpoint method to generate four-part harmonisations. Markov models also form the basis for some real-time improvisation systems [14, 29] and more recent work on Markov constraints for generation [30].

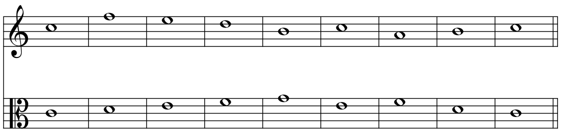
In this paper we adopt the view that music generation can be viewed as sampling high probability sequences from statistical models of a music style. With a simple (first-order) statistical model, such as the one explored in this paper, high probability sequences might not be the best in terms of musicality [27]. This issue will be examined in more detail in future research of the authors, but in this paper we focus on high probability sequences. Although many systems are available to learn styles from existing music, few have been combined with an efficient optimization algorithm such as VNS [27, 12]. This is important since generating high probability sequences from complex statistical models containing multiple conditional dependencies between variables can be a computationally hard problem. In this research we apply the vertical viewpoints method [7] to learn a model that quantifies how well music resembles first species counterpoint. This model is then used to replace the rule-based objective function in a VNS previously developed by the authors [22].

We chose to work with simple first species counterpoint in this paper in order to explore the theoretical concepts of sampling. It is not the goal of this research to develop a complete model, but evaluate the different methods to sample from a statistical model. Section 2 of this paper describes the statistical model used in this study and Section 3 describes the sampling methods used. The statistical model was chosen so that the optimal (Viterbi) solution could be computed, allowing us to evaluate the absolute in addition to the relative performance of various sampling methods. In Section 4 the resulting system is extensively tested and compared to the optimal solution and to the random walk and Gibbs sampling methods.

2. VERTICAL VIEWPOINTS

This section describes the model that provides the probabilities of each note in a first species counterpoint fragment. First species counterpoint can be viewed as a sequence of *dyads* i.e., two simultaneous notes (see Figure 1). In this research the number of possible pitches is constrained to the scale of C major and the range of the cantus firmus, i.e., the fixed voice against which the counterpoint line is composed, is constrained to 48 and 65 (in midi pitch values) and the counterpoint ranges from 59 to 74. These constraints are based on counterpoint examples from Salzer and Schachter [33]. This results in 110 possible dyads (11 × 10).

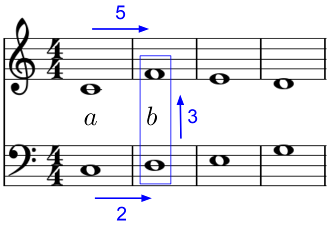
When generating counterpoint fragments, it is essential to consider both vertical (harmonic) and horizontal (melodic) aspects. These two dimensions should be linked instead of treated separately. Furthermore, in order to confront the data sparsity issue in any corpus, abstract representations should be used instead of surface representations. These representational issues are handled by defining a *viewpoint*, a function that transforms a concrete event into an abstract feature. In this paper the *vertical viewpoints* method [7, 10] is used to model harmonic and melodic aspects of counterpoint.



$$\begin{bmatrix}72\\60\end{bmatrix} \begin{bmatrix}77\\62\end{bmatrix} \begin{bmatrix}76\\64\end{bmatrix} \begin{bmatrix}74\\65\end{bmatrix} \begin{bmatrix}71\\67\end{bmatrix} \begin{bmatrix}72\\64\end{bmatrix} \begin{bmatrix}69\\65\end{bmatrix} \begin{bmatrix}71\\62\end{bmatrix} \begin{bmatrix}72\\60\end{bmatrix}$$

Figure 1: First species counterpoint example [33] and its dyad representation.

A simple linked viewpoint is used whereby every dyad is represented by three linked features (see Figure 2): two melodic pitch class intervals between the two melodic lines, and a vertical pitch class interval within the dyad. With this representation, the second dyad b in Figure 2 is given by the compound feature $\tau(b \mid a) = \langle 2, 5, 3\rangle$.



$$\begin{bmatrix}60\\48\end{bmatrix} \begin{bmatrix}65\\50\end{bmatrix}$$

Figure 2: Features (on the arrows) derived from two consecutive dyads a and b (bottom) form compound feature $\tau(b \mid a) = \langle 2, 5, 3\rangle$.

Following this transformation, dyad sequences in a corpus are transformed to more general feature sequences, which are less sparse than the concrete dyad sequence for obtaining statistics from a corpus. In the following it is described how to create a simple first order transition matrix (TM) over dyads from these statistics, which can immediately be applied in any optimization algorithm (see Section 3.1).

Following the method of Conklin [9], let $v = \tau(b \mid a)$

$$\begin{tikzcd} & v = \tau(b \mid a) \arrow[d] \\ a \arrow[r] & b \end{tikzcd}$$

Figure 3: The probabilistic dependencies in the vertical viewpoint model.

be the feature assigned by a viewpoint τ to dyad b, in the context of the preceding dyad a. Assuming the probabilistic graphical model of Figure 3, the probability $P(b \mid a)$ of dyad b following dyad a can be derived as follows:

$$\begin{aligned}P(b \mid a) &= P(a,b)/P(a) \quad \text{conditional probability}\\ &= P(b,v,a)/P(a) \quad \text{because } P(v \mid b,a) = 1\\ &= P(b \mid v,a) \times P(v,a)/P(a) \quad \text{chain rule}\\ &= P(b \mid a,v) \times P(v) \quad \text{independence of } a \text{ and } v\end{aligned}$$

with the second term $P(v)$ estimated from the corpus:

$$P(v) = c(v)/n$$

where n is the number of dyads in the corpus and $c(v)$ is the number of dyads in the corpus having the feature v. To further reduce the number of parameters for training simply to the quantities $P(v)$, the first term $P(b \mid a,v)$ is modelled with a uniform distribution

$$P(b \mid a,v) = |\{x : \tau(x \mid a) = v\}|^{-1}$$

where x ranges over all 110 possible dyads.

As an example of the calculation of $P(b \mid a)$, referring to the first two dyads of Figure 2, consider the probability of the second dyad $b = [65\ 50]$ following the first dyad $a = [60\ 48]$. Suppose that $P(\langle 2,5,3\rangle) = 0.01687$. Given the space of possible dyads, we have

$$|\{x : \tau(x \mid [60\ 48]) = \langle 2,5,3\rangle\}| = 6$$

that is, there are 6 possible dyads

$$\{[65\ 48], [65\ 50], [65\ 62], [60\ 48], [60\ 50], [60\ 62]\}$$

that have the feature $\langle 2,5,3\rangle$ in the context of dyad [60 48]. Therefore for this example:

$$P([65\ 50] \mid [60\ 48]) = 1/6 \times 0.01687 = 0.0028$$

A complete statistical model is created by filling a transition matrix of dimension 110×110 with these quantities for all possible pairs of dyads.

Given a first order transition matrix over dyads, the probability $P(s)$ of a sequence $s = e_1, \ldots, e_\ell$ consisting of a sequence of ℓ dyads is given by

$$P(s) = \prod_{i=2}^{\ell} P(e_i \mid e_{i-1}) \tag{1}$$

This probability will be used to create an objective function, as discussed in the following section.

3. SAMPLING SOLUTIONS FROM A STATISTICAL MODEL

In this research generating counterpoint music is seen as a combinatorial optimization problem, whereby the best combination of notes needs to be found in order to produce music that adheres to a certain style as well as possible. Since generating dyad sequences with the best possible objective function is a computationally hard problem, a variable neighbourhood search algorithm is used as it is an efficient optimization method. Variable neighbourhood search has been successfully applied to a wide range of combinatorial problems [19] including vehicle routing [26], graph colouring [3] and project scheduling [16]. Hansen et al. [21] find that VNS outperforms existing heuristics and is able to find the best solution in moderate computing time for several problems.

In this paper, the VNS previously developed by Herremans and Sörensen [22] is adapted to work with a learned objective function. The VNS method is then compared with two sampling methods i.e., random walk, and Gibbs sampling. One of the reasons for using a first order Markov model to represent first species counterpoint is that it is possible to compute the Viterbi optimum for this problem. This allows a thorough comparison of the sampling methods relative to each other and the optimum solution (see Section 4.2).

3.1 Objective Function

In Section 2 a Markov model with vertical viewpoints was described for learning the characteristics of a corpus of musical pieces. This statistical model is transformed into an objective function that can be used to indicate the quality of a generated fragment. High solution quality corresponds to high probability sequences in the model.

The probability $P(s)$ from Equation 1 is transformed into cross-entropy since it is more convenient to use logarithms. The sum of the logarithms is normalised by the sequence length to obtain the *cross-entropy* $f(s)$:

$$f(s) = -\frac{1}{\ell - 1} \sum_{i=2}^{\ell} \log P(e_i \mid e_{i-1}) \tag{2}$$

The quality of a counterpoint 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.

The Viterbi solution, the minimal cross-entropy solution, can be computed directly from the transition matrix. This is done by a dynamic programming algorithm, which fills a solution matrix of dimension $110 \times \ell$ columnwise, accumulating the best partial path ending at each dyad and sequence position in each cell. The minimal cross-entropy is given by the minimum value within the last column of the solution matrix.

3.2 Variable neighbourhood search

Variable neighbourhood search, or VNS, is a local search based metaheuristic. The structure of the implemented VNS is represented in Figure 4. The algorithm starts from an initial fragment, in this case a fragment with randomly assigned dyads from the set of allowed dyads. From this starting fragment the VNS iteratively makes small improvements (called *moves*) in order to find a better one,

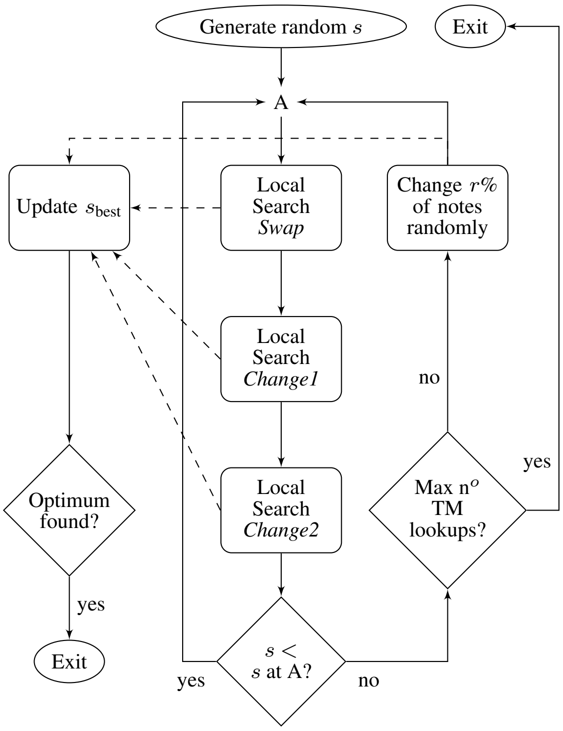


Figure 4: Overview of the VNS.

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 dyad to any other allowed dyad. The last move, *change2*, is an extension of the previous one whereby two sequential dyads are changed simultaneously to all possible allowed dyads.

The set of all possible fragments s' that can be reached from the *current* fragment by a move type is called the *neighbourhood*. The local search uses a steepest descent strategy, whereby the best fragment is selected from the entire neighbourhood. This strategy will quickly steer the algorithm 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 [20]. 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 [22]. The VNS algorithm was implemented in C++ and the source code is available online [1] .

3.3 Random walk

The random walk method [8] is a simple and common way to generate a sequence from a Markov model. The initial dyad is fixed (see Section 4). After that, successive dyads are generated by sampling from the probability distribution given by the relevant row of the transition matrix (based on the previous dyad). That is, at each position i the next dyad e_i is selected with probability $P(e_i \mid e_{i-1})$. If there is no next dyad with non-zero probability, the entire process is restarted from the beginning of the sequence. Iterated multiple times, on every successful iteration, the cross-entropy of the solution is noted if it is better than the best score so far.

3.4 Gibbs sampling

Gibbs sampling is a popular method used in a wide variety of statistical problems for generating random variables from a (marginal) distribution indirectly, without having to calculate the density [6]. The algorithm is given a random piece s generated by the random walk method above. The following process is iterated: a random location in the piece s is chosen and all valid dyads are substituted into that position, each substitution producing a new piece s' having probability $P(s')$. This distribution over all modified pieces is normalized, and one is sampled from this distribution. This process is iterated with s set to the sampled piece. Iterated multiple times, on every iteration the cross-entropy of the solution is noted if it is better than the best score so far.

4. EXPERIMENT

In order to compare the efficiency of the VNS with other techniques an experiment was set up. Since there are no large available corpora restricted to first species counterpoint, 1000 pieces were generated by means of the algorithm with a rule-based objective function [22]. All pieces consist of 64 dyads. These pieces were used to train the Markov model discussed in section 2.

A number of hard constraints are imposed to better define and limit the problem. Firstly, as discussed in Section 2, the range is restricted to 110 dyads. Secondly, the cantus firmus is specified and cannot be changed by the algorithm (thus, the three methods in Section 3 consider only those dyads compatible with the specified cantus firmus). Based on music theory rules specified by Salzer and Schachter [33], a third hard constraint fixes the first dyad to [60 48] and the last two dyads to [59 50] and [60 48]. This brings the number of possible solutions to 10^{61}. The Viterbi solution for this problem has a cross-entropy of 3.22410 (see Table 1).

4.1 Distribution of random walk

Figure 5 shows the distribution of cross-entropy of musical sequences sampled by random walk. A total of 10^7 iterations of random walk sampling were performed, and the cross-entropies (excluding those solutions which led to a dead end during the random walk) were plotted. The

[1] `http://antor.ua.ac.be/musicvns`

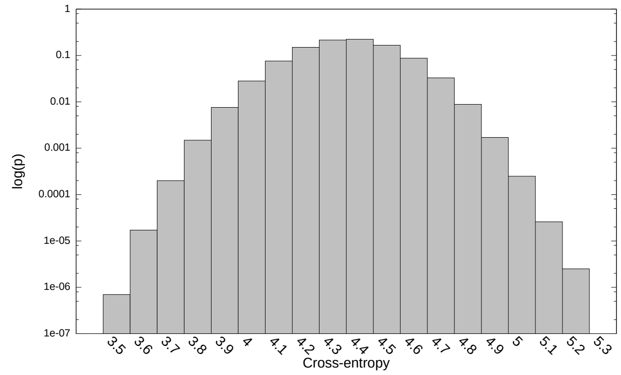


Figure 5: The distribution of cross-entropy according to random walk, over 10^7 iterations.

plot therefore shows the probability of random walk producing some solution within the indicated cross-entropy bin. The difficulty of sampling high probability solutions is immediately apparent from the graph. For example, in order to generate one solution in the cross-entropy range of $3.5 - 3.6$ (a solution still worse than the Viterbi solution), approximately one million solutions should be sampled with random walk. Figure 5 also shows that even with the large number of random walk samples taken (10^7), the Viterbi solution is not found.

4.2 Performance of the algorithms

To evaluate the relative performance of the methods, the number of transition matrix lookups (TM lookups) is used as a complexity measure in order to compare the VNS with random walk and Gibbs sampling. A total of 100 runs are performed with a cut-off point of 3×10^7 TM lookups or alternatively until the Viterbi solution is reached.

The average of the best scores of each of the 100 runs are displayed in Table 1 per algorithm. A one-sided Mann-Whitney-Wilcoxon test was performed to test if the results attained by the VNS are significantly better than the ones from the random walk and Gibbs sampling. Since the p-values of the latter algorithms are both $< 2.2 \times 10^{-16}$ we can accept the alternative hypothesis which states that the results from the VNS are lower (i.e., better) than the ones for both random walk (RW) and Gibbs sampling (GS). The VNS was able to find the optimal fragment ($f(s) = 3.2241$) before the cut-off point of 3×10^7 TM lookups in 51% of the cases. Neither GS nor RW were able to reach the optimum in any of the iterations. The best cross-entropy values reached by all three of the algorithms during the 100 runs are displayed in Table 1.

Figure 6 shows the evolution of the average value of the objective function for the best fragment found by the algorithms over 100 runs. The ribbons on the graph indicate the best and worst run of each algorithm. The Viterbi optimum is displayed as the lower horizontal line. It is clear from the graph that VNS outperforms both GS and RW. All three algorithms seem to start with a very steep descent in the very beginning of the run, but GS and RW converge faster. Gibbs sampling does perform slightly better than random walk, but the best run is still worse than the worst run of the VNS.

Figure 7 focuses on the first 50,000 TM lookups displayed in Figure 6. In the very beginning of the runs, VNS is outperformed by the two simpler algorithms. This is probably due to the fact that the VNS starts from a random initial solution that allows zero-probability transitions. Even so, the algorithm is able to quickly improve these solutions. A combination of VNS with an initial starting solution generated by a random walk could even further improve its efficiency.

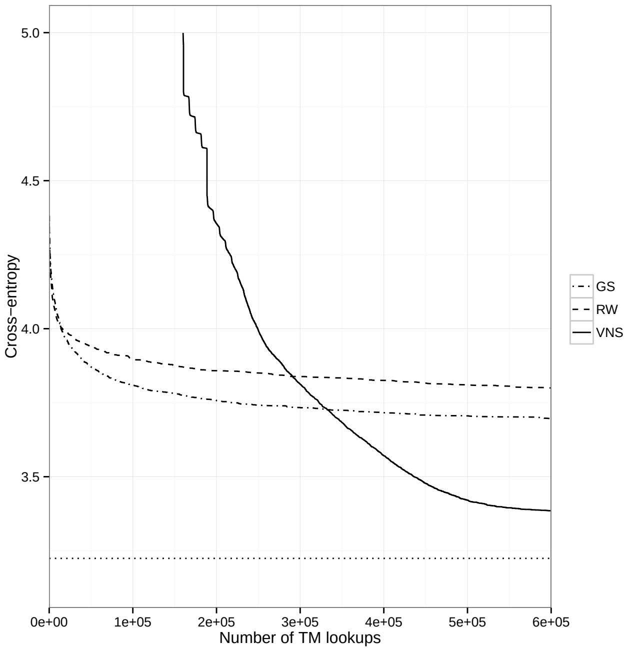


Figure 7: Evolution of the best fragment found zoomed in on the beginning of the runs.

5. CONCLUSION

The approach used in this research shows the possibilities of combining *music generation* with *machine learning* and provides us with an efficient method to generate music from styles whose rules are not documented in music theory. The proposed VNS algorithm is a valid and flexible optimization method that is able to find the fragment with the best dyad transitions according to a learned model. It outperforms both random walk and Gibbs sampling in terms of sampling of high probability solutions. The focus of this paper is on high probability (low cross-entropy) regions, but the VNS can just as easily be applied to sample low probability regions. In addition to the VNS contribution, in this paper we confirmed that random walk does not practically (only in the theoretical limit of iterations) sample from the extrema (i.e., sampling the highest probability pieces), from even a simple Markov model.

It must be mentioned that the absolute cross-entropy results presented in this paper possibly have some bias towards the VNS method, because the moves used to generate the training data for the creation of the statistical model

	Viterbi	random walk	Gibbs sampling	VNS
average	n/a	3.64788	3.53426	3.24453
best	3.22410	3.44837	3.39871	3.22410

Table 1: Average and best results of 100 runs after 30 million transition matrix lookups.

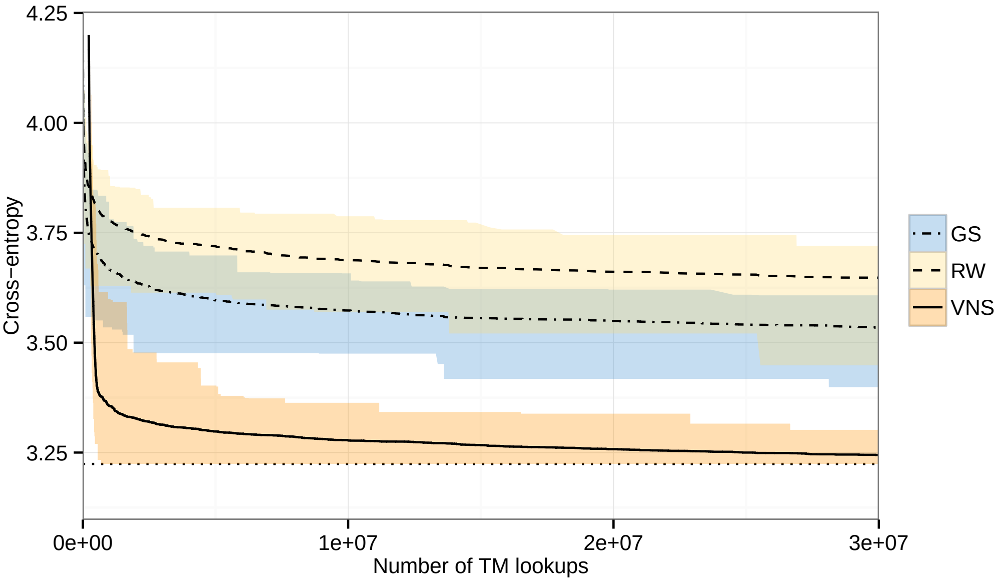


Figure 6: Evolution of the best fragment found.

are in fact the same as those used by VNS during the search of the solution space. Therefore, we plan to test in future research if the relative performances of the sampling methods hold up under independent training data.

The results are promising as the VNS method converges to a good solution within relatively little computing time. The described VNS is a valid and flexible sampling method and has been successfully combined with the vertical viewpoints method. In future research, these methods will be applied to higher species counterpoint [23, 38, 10] with the multiple viewpoint method [9], using more complex learned statistical models. When generating more complex music, new move types should be added to the VNS in order to escape local optima. The consideration of more complex contrapuntal textures will also permit the use of a real corpus.

Acknowledgments

This research is supported by the project Lrn2Cre8 which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859. This research is also partially supported by the Interuniversity Attraction Poles (IAP) Programme initiated by the Belgian Science Policy Office (COMEX project).

6. REFERENCES

[1] M. Allan and C. Williams. Harmonising chorales by probabilistic inference. In *Advances in Neural Information Processing Systems*, 2004.

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

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

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

[5] E. A. Bowles. Musicke's handmaiden: Of technology in the service of the arts. In H. B. Lincoln, editor, *The Computer and Music*. Cornell University Press, Ithaca, NY: Cornell ed., 1970.

[6] G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

[7] D. Conklin. Representation and discovery of vertical patterns in music. In C. Anagnostopoulou, M. Ferrand, and A. Smaill, editors, *Music and Artificial Intelligence: Lecture Notes in Artificial Intelligence 2445*, pages 32–42. Springer-Verlag, 2002.

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

[9] D. Conklin. Multiple viewpoint systems for music classification. *Journal of New Music Research*, 42(1):19–26, 2013.

[10] D. Conklin and M. Bergeron. Discovery of contrapuntal patterns. In *ISMIR 2010: 11th International Society for Music Information Retrieval Conference*, pages 201–206, Utrecht, The Netherlands, 2010.

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

[12] S. Davismoon and J. Eccles. Combining musical constraints with Markov transition probabilities to improve the generation of creative musical structures. In *Applications of Evolutionary Computation*, pages 361–370. Springer, 2010.

[13] P. Donnelly and J. Sheppard. Evolving four-part harmony using genetic algorithms. In *EvoApplications (2)*, volume 6625 of *Lecture Notes in Computer Science*, pages 273–282. Springer, 2011.

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

[15] M. Farbood and B. Schoner. Analysis and synthesis of Palestrina-style counterpoint using Markov chains. In *Proceedings of the International Computer Music Conference*, pages 471–474, 2001.

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

[17] M. Geis and M. Middendorf. An ant colony optimizer for melody creation with Baroque harmony. In *IEEE Congress on Evolutionary Computation*, pages 461–468, 2007.

[18] D. Gürer. Pioneering women in computer science. *ACM SIGCSE Bulletin*, 34(2):175–180, 2002.

[19] P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.

[20] P. Hansen and N. Mladenović. Variable neighborhood search. In *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 145–184. Springer US, 2003.

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

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

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

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

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

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

[27] M. Y. Lo and S. M Lucas. Evolving musical sequences with n-gram based trainable fitness functions. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 601–608. IEEE, 2006.

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

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

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

[31] G. Papadopoulos and G. Wiggins. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposium on Musical Creativity*, pages 110–117. Edinburgh, UK, 1999.

[32] S. Raczyński, S. Fukayama, and E. Vincent. Melody harmonization with interpolated probabilistic models. *Journal of New Music Research*, 42(3):223–235, 2013.

[33] F. Salzer and C. Schachter. *Counterpoint in composition: the study of voice leading*. Columbia University Press, 1969.

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

[35] N. Tokui and H. Iba. Music composition with interactive evolutionary computation. In *Proceedings of the Third International Conference on Generative Art*, volume 17:2, pages 215–226, 2000.

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

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

[38] R. P. Whorley, G. A. Wiggins, C. Rhodes, and M. T. Pearce. 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, 2013.

[39] I. Xenakis. *Formalized Music: Thought and mathematics in composition*. Number 6. Pendragon Pr, 1992.