Sampling Markov Models under Binary Equality Constraints Is Hard

Stephane Rivaud François Pachet Pierre Roy

Sony CSL, 6 rue amyot, 75005 Paris

srivaud972@gmail.com
pachetcsl@gmail.com
roypie@gmail.com

Abstract

Many content generation applications in sequential domains such as music or text are based on sampling from statistical models. In practice, users need solutions that satisfy specific, additional properties not captured by the model. For instance, music generation often requires long-range, structural constraints to be satisfied, and these are typically ignored by statistical models such as Markov models. Several works have shown that it is possible to sample, in polynomial time, Markov models subject to specific types of constraints. In this paper, we address the problem of sampling Markovian sequences satisfying equality constraints between distant variables. Such constraints are motivated by the need to enforce high-level structure, such as the repetition of patterns. Our contribution is twofold. First, we show that exact sampling is, in this case, #P-hard. Second, we show empirically that a random walk approach augmented by a filtering algorithm leads to efficient sampling, but with a significant bias in the output's distribution. Finally, we show that specific configurations of equality constraints can be handled in polynomial time. We illustrate this result on a chord sequence generation problem.

Keywords

Markov Model, Binary constraint, Equality, Complexity, Sampling.

1 Introduction

Recent years have seen an increase in style imitation applications in sequential domains such as text and music. Although *style* remains an elusive notion, it is most often modeled as a statistical object, usually estimated from a representative corpus of sequences. Once a statistical model is estimated, new sequences can be sampled according to the underlying distribution of the model [2].

However, in practice, statistical models do not capture all the properties of the corpus from which they have been trained. For instance, meter, harmony, and more generally properties involving long-range correlations are not well captured by statistical models that involve only local features of the corpus at stake. It is therefore interesting to use the technology of constraint programming to enforce *a posteriori* these properties and to somehow combine the statistical power of the model with the filtering of such global constraints.

Several results have been obtained recently in this direction, for instance to enforce meter [17], nogoods (anti-plagiarism) [12] or $1/f$ spectrum properties [8] in sequences generated by Markov models, in polynomial time.

Such an endeavor follows a trend in constraint programming consisting in reformulating existing statistical algorithms as constraint satisfaction problems (CSP), notably for Markov chains [7, 9], neural networks [6, 1], and more generally statistical constraints [14, 15]. This work can be seen as yet another bridge between stochastic algorithms and finite-domain, discrete constraint satisfaction.

2 Markov Models under Binary Equality Constraints

The problem we address in this paper was first highlighted in [3]. It consists in sampling a sequence from a Markov model of chord progressions, that additionally satisfies a set of *binary equality constraints*. In the cited publication, this model was estimated from a corpus of chord sequences in the Trance music style. Binary equality constraints hold on specific indexes of the sequence, i.e. if we consider a sequence to be sampled as a set of variables $x_1, x_2, \ldots, x_n$, then each constraint $C_{i,j}$ states that $x_i = x_j$. An arbitrary number of such equalities can be set on a sequence. Obviously, the number of equality is bounded by $\binom{n}{2} = \frac{n(n-1)}{2}$ if n is the size of the sequence.

The motivation for imposing such binary equalities is to simulate *structural properties* observed in real sequences, a notoriously difficult problem in statistical modeling. Indeed, enforcing binary constraints is a way to specify repeating patterns of arbitrary lengths in a

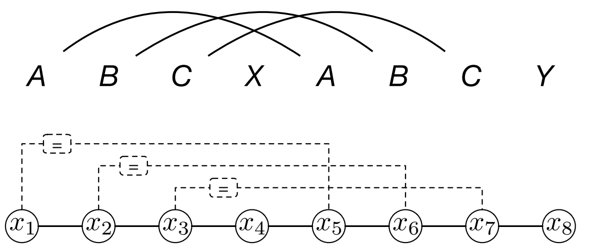

FIGURE 1 – A typical configuration of binary equalities, imposed on a Markov sequence to specify the repetition of a pattern. Here the three first values should be repeated at a specific position in the sequence

finite sequence. A set of binary equalities can be used to specify a pattern of arbitrary size to be repeated at specific locations of the sequence as illustrated in Figure 1.

In the original publication, a modified random walk procedure was proposed. In general, a simple random walk is impractical and a random walk augmented with filtering introduces a statistical bias in the output's distribution. Binary equalities introduce cycles and dependencies in the corresponding models that need to be somehow propagated.

If there is a fixed small number of such equalities, the corresponding language defined by the set of constraints can be considered as regular. In that case, the result described in [11] can be applied. However, the size of the automaton grows exponentially with the number of constraints and quickly becomes intractable.

In this paper, we address the problem in its full generality, and study the complexity of the problem from a sampling perspective.

The rest of the paper is organized as follows. In Section 1, we show that the problem of deciding if there is a solution to the Markov + binary equality constraints with a non zero probability is NP-hard. This proof is based on a parcimonious reduction of binary CSPs to a Markov + binary equality problem. We also show that the corresponding inference problem is #P-hard. In Section 2, we show empirically that using a filtering procedure in a random walk scheme does enable us to find non zero probabilities solutions quickly, but that these probabilities are wrong. In Section 3, we identify a number of specific cases corresponding to configurations of the binary equalities (including our motivating example), for which we exhibit a polynomial sampling algorithm, and we report on experiments conducted on a music chord sequence generation problem.

3 Complexity of Decision and Sampling Problems

A CSP is fully represented by its constraint graph, i.e. a graph where the vertices represent the variables, and each edge (x, y) is labeled by a function $C(x, y)$ taking value 1 when (x, y) satisfies the constraint and 0 otherwise. Factor graphs are a generalization of constraint graph, i.e. $C(x, y)$ can take any real value. They allows us to answer queries about the CSP in a probabilistic framework. Two fundamental queries are, for example, is the CSP satisfiable (decision problem), and if so, how many solutions does it have (counting problem). Let G be the factor graph associated with the CSP, and $\mathbb{P}_G$ the unnormalized measure associated with the factor graph. It can easily be shown that $\mathbb{P}_G$ is a uniform distribution over the solution of the CSP, where each solution have measure 1. The decision problem thus consists in determining whether or not this distribution is the null distribution, and the counting one consists in computing the normalizing constant $Z = \sum_X \mathbb{P}_G(X)$.

Proposition 1. *The decision (resp. counting) problem associated with a Markov model subjet to a set of equality constraints is NP-complete (resp. #P-complete), i.e. deciding if there is an (resp. counting the number of) assignment that have non zero probability.*

Démonstration. Let A be a $(d, 2)$-CSP and G its associated factor graph, i.e. A is a binary CSP where the maximum size of a variable domain is d. We will show that the decision (resp. counting) problem associated to A can be reduced to the decision (resp. inference) problem in a Markov model subject to a set of equality constraints.

We unwrap the factor graph G into a chain G' with a set of equality constraints $\mathcal{C}_{\text{eq}} = \{C_1, \ldots, C_{n_{\text{eq}}}\}$ such that the contraction of G' with respect to $\mathcal{C}_{\text{eq}}$ is G [[13], p.231]. One property that the linear chain must follow is that the set of binary constraints labeling its edges must be the same set of constraints labeling the original graph, regardless of constraints allowing all pairs of values (dummy edge). In order to build the unwrapped graph G' one can use the Algorithm 1 [1]. An example is shown in Figure 2.

As the factor graph G' is a chain, we can factorize $\mathbb{P}_{G'}(X) = \mathbb{P}_{G'}(X_1) \prod_{n=2}^{N} \mathbb{P}_{G'}(X_n | X_{n-1})$ because of the independence assumptions it must satisfies. The conditional probabilities can be obtained by belief propagation on G', leaving us with the Markov model $\mathcal{M}$ [5]. By construction, there exists a non zero probability sequence satisfying the equality constraints if and only if A has a solution.

[1] We do not discuss the efficiency of the unwrapping method. We only need it to be polynomial, and the method we propose is polynomial.

Algorithme 1 : Linearizing graphs

```
Data : $G = (V, E)$ − graph
Result : $G' = (V', E')$ − linear chain
         $\mathcal{C}$ − set of binary equality constraints
$E' \leftarrow \emptyset$;
$V' \leftarrow \emptyset$;
$\mathcal{C} \leftarrow \emptyset$
// initializing $E'$
(left, right) $\leftarrow$ first edge of $E$;
add (left, right) to $E'$ ;
add left to $V'$;
add right to $V'$;
remove (left, right) from $E$
while $E$ is not empty do
    (left, right) $\leftarrow$ first edge of $E$;
    last $\leftarrow$ last vertex of $V'$;
    if last == right then
        // we only need to add right
        add (last, right) to $E'$;
        add right to $V'$;
        remove (left, right) from $E$;
        if right was already in $V'$ then
            // add an equality constraint
            $i \leftarrow$ last occurence of right in $V'$ before current position;
            $j \leftarrow$ current position in $V'$;
            Add EqualConst($V(i), V(j)$) to $\mathcal{C}$;
    else
        // we need to add $e_1$ and $e_2$
        edge $\leftarrow$ dummyEdge(last, left);
        add edge to $E'$;
        add left to $V'$;
        add (left, right) to $E'$;
        remove (left, right) from $E$;
        if left was already in $V'$ then
            // add an equality constraint
            $i \leftarrow$ last occurence left in $V'$ before current position;
            $j \leftarrow$ current position in $V'$;
            Add EqualConst($V(i), V(j)$) to $\mathcal{C}$;
        if right was already in $V'$ then
            // add an equality constraint
            $i \leftarrow$ last occurence right in $V'$ before current position;
            $j \leftarrow$ current position in $V'$;
            Add EqualConst($V(i), V(j)$) to $\mathcal{C}$;
$G' \leftarrow (V', E')$;
```

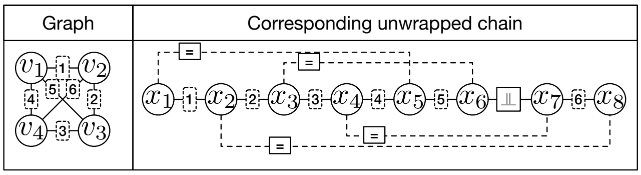

FIGURE 2 – Left : constraint graph. Right : unwrapped chain with additionnal equalities
Mapping : $(v_1 \rightarrow x_1, x_5)$, $(v_2 \rightarrow x_2, x_8)$, $(v_3 \rightarrow x_3, x_6)$, $(v_4 \rightarrow x_4, x_7)$.
The edge between x_6 and x_7 represent a constraint allowing every pair of values in order to avoid redundancy.

Since solving A is NP-hard in general [4], the decision problem associated with $\mathcal{M}$ and $\mathcal{C}_{\text{eq}}$ is NP-hard.

We can write the following equation

$$\mathbb{P}_{\mathcal{M}}\left(C_{\text{eq}}\right) = \mathbb{P}_{\mathcal{M}}\left(C_1\right) \prod_{n=2}^{n_{\text{eq}}} \mathbb{P}_{\mathcal{M}}\left(C_n | C_{n-1}, \ldots, C_1\right)$$

where $\mathbb{P}_{\mathcal{M}}\left(C_{\text{eq}}\right)$ is the probability that the constraints in C_{eq} are satisfied, according to the Markov model. Suppose that C_n is an equality between X_i and X_j, then $\mathbb{P}_{\mathcal{M}}\left(C_n | C_{n-1}, \ldots, C_1\right) = \sum_{x=1}^{d} \mathbb{P}_{\mathcal{M}}\left(X_i = x, X_j = x | C_{n-1}, \ldots, C_1\right)$
So if we can perform inference with respect to $\mathbb{P}_{\mathcal{M}}\left(\cdot | C_n, \ldots, C_1\right)$ for all $1 \leqslant n \leqslant n_{\text{eq}}$, where $\mathbb{P}_{\mathcal{M}}\left(\cdot | C_n, \ldots, C_1\right)$ stands for the distribution $X \mapsto \mathbb{P}_{\mathcal{M}}\left(X | C_n, \ldots, C_1\right)$, we can compute $\mathbb{P}_{\mathcal{M}}\left(C_{\text{eq}}\right)$. Let us recall that :

- $\mathbb{P}_{\mathcal{M}}$ is the normalization of $\mathbb{P}_{G'}(\cdot)$, i.e. $\mathbb{P}_{\mathcal{M}}(\cdot) = \frac{1}{Z(G')} \mathbb{P}_{G'}(\cdot)$.
- $\mathbb{P}_{\mathcal{M}}\left(\cdot | \mathcal{C}_{\text{eq}}\right)$ is the normalization of $\mathbb{P}_{\mathcal{M}}\left(\cdot, \mathcal{C}_{\text{eq}}\right)$, i.e. $\mathbb{P}_{\mathcal{M}}(\cdot) = \frac{1}{\mathbb{P}_{\mathcal{M}}(C_{\text{eq}})} \mathbb{P}_{\mathcal{M}}\left(\cdot, \mathcal{C}_{\text{eq}}\right)$.

Therefore $\mathbb{P}_{\mathcal{M}}\left(\mathcal{C}_{\text{eq}}\right) = \frac{Z(A')}{Z(G')}$ where $Z(A')$ is the number of satisfying assignment for A'. So if we can perform inference with respect to $\mathbb{P}_{\mathcal{M}}\left(\cdot | \mathcal{C}_{\text{eq}}\right)$, we can also compute $Z(A')$. Since computing $Z(A')$ is #P-hard [16], we have proved the #P-completeness of the inference problem. □

4 Empirical Results with a Filtering Approach

In this Section we illustrate the problem on a concrete example : the generation of a chord sequence in a given style. We consider a corpus of 232 song by the Beatles, taken from the LSDB database [10]. The size of the alphabet here is 56 (different chords, e.g. C, Eb min, etc.). For example, the song *Hey Jude* begins with the sequence of chords *F, C, C7, F, Bb, F, C7, F,...* .
We build a first-order Markov model from this corpus, and we consider the following problem :

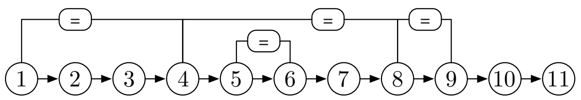

Figure 3 – Constrained markov model associated with *Problem#1*

Problem#1 := Generate a sequence of 11 chords $X_1, \dots, X_{11}$ with the set of equalities $\{X_1 = X_4 = X_8 = X_9\}$ and $\{X_6 = X_7\}$.

The filtering approach consists in a random walk where constraints are propagated after each instantiation :

1. **Arc-consistency :** filter out values forbidden by Markovian transitions and equality constraints until a fixed point is reached.
2. **Instantiation :** instantiate the first uninstantiated variable according to its prior probability restricted to the filtered domain.

We then plot the probabilities estimated from sampling versus their actual probabilities in the Markov model. Such a plot should look like a straight line if we perform a perfect sampling. The bias in the output's distribution is represented by the deviation from such an ideal plot. Figure 4 represents *Problem#1*, and the probability of the model is not normalized according to the equality constraints, due to the complexity of computing the normalizing constant i.e. model's probability is $\mathbb{P}_{\mathcal{M}}(X)$ instead of $\mathbb{P}_{\mathcal{M}}(X|\mathcal{C}_{\text{eq}})$.

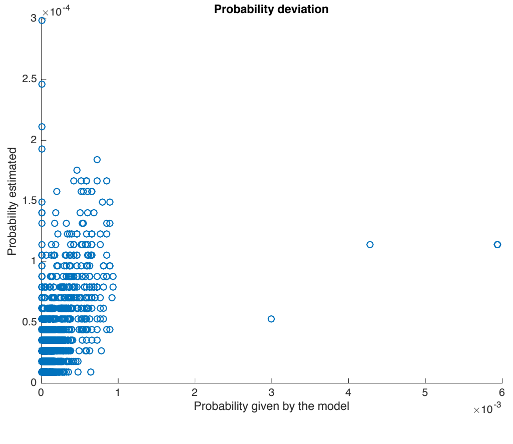

Figure 4 – Sampling of a Markov sequence under 4 binary equality constraints, using a filtering approach. The most outputed sequence is *G5, G5, G5, G5, G5, G5, G5, G5, G5, G5, G5, G*.

5 Specific Cases

We have shown that the general problem of sampling a Markov chain under an arbitrary set of binary equality constraints is hard. Experiments using a simple filtering approach do find non zero probability solutions but with a strong bias. The difficulty of the general problem is intrinsically related to the structure of the associated binary CSP introduced in Section 3, and in particular its tree-width. We exhibit here two specific cases in which the problem can be solved in polynomial time, by exploiting the specific structure of the associated binary CSP. The general algorithm we propose is summarized in Algorithm 2.

```
Algorithme 2 : Constrained sampling
Data : $\mathcal{M}$ – Markov model
$N$ – length of the sequence
$\mathcal{C}$ – Set of binary equality constraints
Result : $X$ – Markovian sequence sampled with
         probability $\mathbb{P}_{\mathcal{M}}(X|\mathcal{C})$
1 $G \leftarrow$ factor graph associated with $\mathcal{M}$ and $N$
2 $G' \leftarrow G$
3 $G' \leftarrow$ marginalize out unconstrained nodes in $G'$ according to $\mathcal{C}$
4 $G' \leftarrow$ contraction of $G'$ according to $\mathcal{C}$
5 $X_{\mathcal{C}} \leftarrow$ exact sampling from $G'$
6 $G \leftarrow$ instantiate constrained variable in $G$ with $X_{\mathcal{C}}$
7 $X \leftarrow$ sample $G$ with belief propagation
```

The algorithm can be decomposed into two steps :

1. Sampling from constrained nodes.
2. Performing belief propagation given the constrained nodes.

The major complexity of the problem lies in the first step. In the rest of this section, we first recall how to marginalize nodes in a linear chain, then we exhibit two cases where the first step of our sampling algorithm can be done in polynomial time.

5.1 Marginalizing in a Linear Chain

Let G be a linear chain representing the distribution $\mathbb{P}(X_1, X_2, X_3) = \frac{1}{Z} f_1(X_1, X_2) f_2(X_2, X_3)$. The factor graph obtained after marginalizing X_2 is still a chain. It is obtained by removing node X_2 and merging factors f_1 and f_2 into a factor f such that

$$f(X_1, X_3) = \sum_{X_2} f_1(X_1, X_2) f_2(X_2, X_3)$$

The marginalization step in a linear chain has complexity $\mathcal{O}(d^3)$ for each marginalized node where d is the size of the variable's domain. Figure 5 summarizes this procedure.

5.2 Non-crossing Equalities

When the binary equality constraints do not cross each other, the factor graph obtained at line 5 of Algorithm 2 is a linear chain with an additional unary factor for each node. In Figure 6, the first factor is $f_1(y_1) = \mathbb{P}_{\mathcal{M}}(X_1 = y_1, X_2 = y_1)$, the second factor is

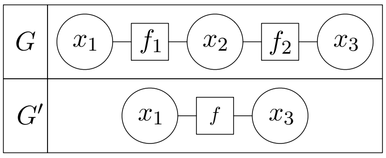

FIGURE 5 – Marginalizing node X_2 from the graph G on the top gives the graph G' on the bottom.

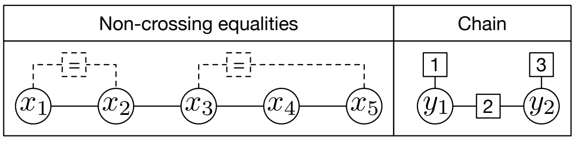

FIGURE 6 – Left : Markov model $\mathcal{M}$ with equality constraints. Right : Quotient graph obtained after marginalizing over x_4. Mapping : $(x_1, x_2 \to y_1)$, $(x_3, x_5 \to y_2)$.

$f_2(y_1, y_2) = \mathbb{P}_{\mathcal{M}}(X_3 = y_2 | X_2 = y_1)$, and the third factor is $f_3(y_2) = \mathbb{P}_{\mathcal{M}}(X_5 = y_2 | X_3 = y_2)$. We can apply belief propagation to perform perfect sampling. The overall procedure takes $\mathcal{O}(d^3 N)$ where N is the umber of nodes, as the marginalization step takes $\mathcal{O}(d^3)$ for each marginalized node.

5.3 Repeating Patterns

Another interesting case is when binary equalities correspond to a repeating pattern, as in our motivating example in Figure 1. In that case, the quotient graph obtained at line 4 of Algorithm 2 can be shown to be a simple cycle. Figure 7 shows the quotient graph associated with the example shown in Figure 1, where we have marginalized variables X and Y.

Let us note $X_1, \ldots, X_n$ the variables of a simple cycle factor graph. It can be easily unfolded into a linear chain with nodes $Y_1, \ldots, Y_{n+1}$, with the additional constraint $\{Y_1 = Y_{n+1}\}$. In this linear chain, Y_1 and Y_{n+1} correspond to X_1, and for $2 \leqslant i \leqslant n-1$, Y_i correspond to X_i. We need to compute the marginal probability $\mathbb{P}(X_1 = x) = \mathbb{P}(Y_1 = x | Y_1 = Y_{n+1})$ for all x in the variable domain. To do so, we instantiate $Y_1 = x$ and $Y_{n+1} = x$, then we perform belief propagation to compute the normalization constant $Z(x)$ of the linear chain where Y_1 and Y_{n+1} instantiated. We can compute this normalisation constant in $\mathcal{O}(d^2 n)$ and we have that $Z(x) = \mathbb{P}(X_1 = x) = \mathbb{P}(Y_1 = x | Y_1 = Y_{n+1})$. Once we have computed $\mathbb{P}(X_1 = x)$ for all x, the sampling scheme is the following :

- Sample X_1 according to $\mathbb{P}(X_1) = \mathbb{P}(Y_1 | Y_1 = Y_{n+1})$.
- Sample X_i for $1 \leqslant i \leqslant n$ using belief propagation algorithm.

The overall procedure is in $\mathcal{O}(d^3 n)$. Once we have sampled all the values of the quotient graph, we can follow the rest of Algorithm 2 to sample the unconstrained variables.

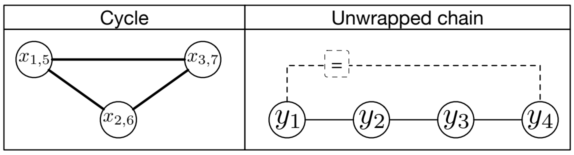

FIGURE 7 – Left : Quotient constraint graph of example shown in Figure 1, after marginalization of unconstrained variables. Right : Unwrapped chain correpsonding to the quotient graph. Mapping : $(x_{1,5} \to y_1, y_4)$, $(x_{2,6} \to y_2)$, $(x_{3,7} \to y_3)$.

The complexity of this case does not depend on the number of binary equality constraint. We performed a sampling experiment with a first order Markov model trained on a corpus of 129 songs composed by Charlie Parker, made of 51 different chords. The generated sequences contains 12 chords and must satisfy $\{X_1 = X_9, X_2 = X_{10}, X_3 = X_{11}, X_4 = X_{12}\}$, as in the example of Figure 1. Figure 8 plot the estimated probabilities vs the model's probability normalized with respect to the set of equality, i.e. $\mathbb{P}_{\mathcal{M}}(X | \mathcal{C}_{\text{eq}})$.

The most frequent instantiation of this pattern of 12 chords in the corpus is *Bb, Gm, Cm, F, Bb, G, Cm, F, Bb, Gm, Cm, F*. The second most outputed pattern by our algorithm is *Bb, Cm, F, Bb, Bb, Cm, F, Bb, Bb, Cm, F, Bb*, which is close to the most frequent one in the corpus [2], i.e. the constrained variables almost take the desired values.

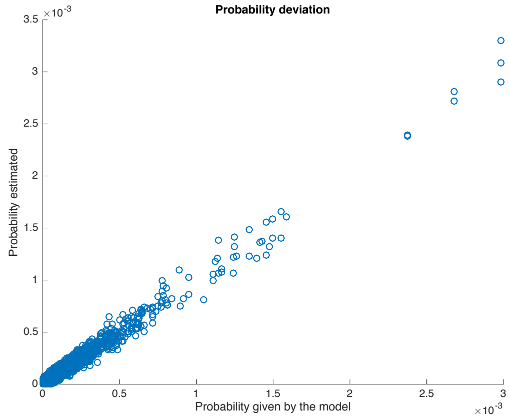

FIGURE 8 – Sampling of a Markov sequence under 4 binary equality constraints, using a belief propagation approach on the quotient graph. Sampling is here perfect.

6 Discussion

We have introduced a sampling problem originating from a music generation problem : sampling Mar-

[2] The most outputed sequence being *Bb, Cm, F, Bb, Bb, Bb, Cm, F, Bb, Cm, F, Bb*.

kov sequences satisfying a set of binary equality constraints. We have shown that the problem in general is #P-hard. The proof is based on the fact that a binary CSP can be transformed into a Markov model subject to equality constraints. We have empirically shown that it is possible to sample such sequences using filtering at the price of a significant statistical bias. We have shown that for specific configurations of binary equalities, polynomial solutions can be found. When used without additionnal constraint, these algorithms do not give satisfactory results, however they can easily be mixed with all regular constraints because it is based on a belief propagation scheme.
This problem is an instance of a *Markov + X* problem where X is a hard constraint, which turns out to be NP-hard in general, as opposed to previous results in this domain (e.g. regular or nogood constraints). More generally we believe that the combination of statistical models and hard constraints is an interesting approach to follow.

7 Acknowledgments

This research is conducted within the Flow Machines project funded by the ERC under the European Unions 7th Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 291156 and the Lrn2Cre8 project (FET grant agreement no. 610859).

Références

[1] Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. Neuron constraints to model complex real-world problems. In *Principles and Practice of Constraint Programming - CP 2011, Perugia, Italy*, pages 115–129, 2011.

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

[3] Darrell Conklin. Chord sequence generation with semiotic patterns. In *MML 2015 : 8th International Workshop on Machine Learning and Music*, pages 1–3, 2015.

[4] Michael R Garey, David S. Johnson, and Larry Stockmeyer. Some simplified np-complete graph problems. *Theoretical computer science*, 1(3) :237–267, 1976.

[5] Daphne Koller and Nir Friedman. *Probabilistic graphical models : principles and techniques.* MIT press, 2009.

[6] Michele Lombardi and Stefano Gualandi. A new propagator for two-layer neural networks in empirical model learning. In *Principles and Practice of Constraint Programming - 19th International Conference, CP 2013, Uppsala, Sweden, September 16-20, 2013. Proceedings*, pages 448–463, 2013.

[7] Michael Morin and Claude-Guy Quimper. The markov transition constraint. In *Integration of AI and OR Techniques in Constraint Programming - 11th International Conference, CPAIOR 2014, Cork, Ireland, May 19-23, 2014. Proceedings*, pages 405–421, 2014.

[8] F. Pachet, P. Roy, A. Papadopoulos, and J. Sakellariou. Generating 1/f noise sequences as constraint satisfaction : the voss constraint. Buenos Aires (Argentina), July 2015. 24th International Joint Conference on Artificial Intelligence (IJCAI 2015).

[9] François Pachet and Pierre Roy. Markov constraints : steerable generation of Markov sequences. *Constraints*, 16(2), 2011.

[10] François Pachet, Jeff Suzda, and Dani Martinez. A comprehensive online database of machine-readable lead-sheets for jazz standards. In *ISMIR*, pages 275–280, 2013.

[11] A. Papadopoulos, F. Pachet, P. Roy, and J. Sakellariou. Exact sampling for regular and markov constraints with belief propagation. Cork (Ireland), 2015. 21th Principles and Practice of Constraint Programming Conference (CP 2015).

[12] Alexandre Papadopoulos, Pierre Roy, and François Pachet. Avoiding plagiarism in markov sequence generation. In *Proc. of AAAI*, pages 2731–2737, 2014.

[13] Sriram Pemmaraju and Steven Skiena. *Computational Discrete Mathematics : Combinatorics and Graph Theory with Mathematica ®*. Cambridge University Press, New York, NY, USA, 2003.

[14] Steven David Prestwich, Roberto Rossi, and S. Armagan Tarim. Randomness as a constraint. In Gilles Pesant, editor, *Principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings*, volume 9255 of *Lecture Notes in Computer Science*, pages 351–366. Springer, 2015.

[15] Roberto Rossi, Steven David Prestwich, and S. Armagan Tarim. Statistical constraints. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 777–782. IOS Press, 2014.

[16] Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1) :273–302, 1996.

[17] Pierre Roy and François Pachet. Enforcing meter in finite-length markov sequences. In *Proc. of AAAI*, 2013.