Association for Constraint Programming

Flow-Machines: CP techniques to model style in music and text

François PACHET, Sony CSL-Paris and LIP6/UPMC.

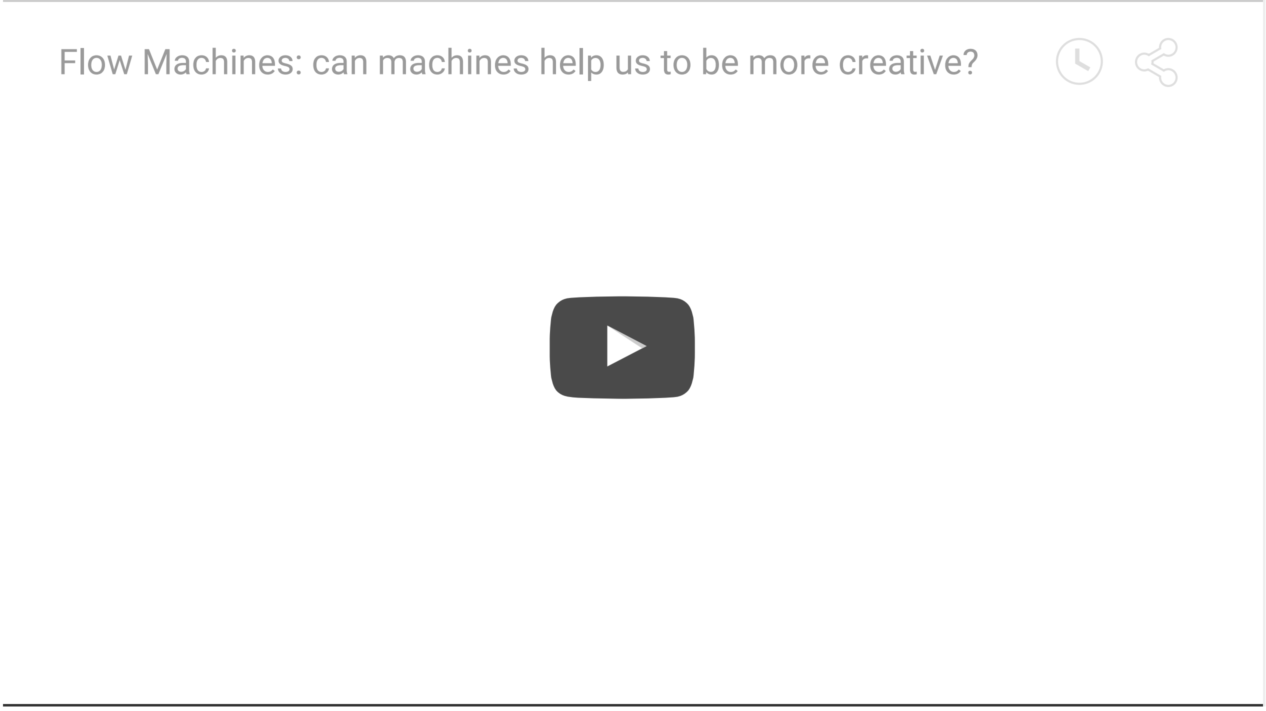
Creativity and innovation are the new motto of the modern world. We are constantly exhorted to be creative, to think out of the box, to push our limits. Not only individuals, but also companies need to show that they are always pushing boundaries, distinguishing themselves. A look at today's big brand logos is revealing: Sony's "like.no.other" and "go create", NEC "empowered by innovation", Hummer "like nothing else", Mercury cars "New doors opened", Dodge "Different", Mercedes-Benz "Unlike any other", Toshiba "Power of innovation", HP "think beyond", Intel's "leap ahead", Samsung "Be creative" or Apple "Think different". Everything around us reminds us to always be innovative, create new things and be singular. We are used to such vacuous advertisement and most of us learned not to take notice anymore.

You get back to that goddamn cubicle and start thinking outside the box!
CREATIVITY CORPORATION
Mr.fish

In the Flow-Machines project, however, we are trying to take these exhortations seriously. Flow machines (funded by the European Research Council (ERC) and coordinated by François Pachet) are content creation software aiming at enhancing individual creativity.

The vision behind flow machines is the concept of "reflexive interactions", a concept introduced some time ago by our team at Sony CSL. Reflexive Interactions are human-machine interactions with a system that attempts to imitate the user's behavior. This idea was first introduced with **the Continuator system (http://www.francoispachet.fr/continuator/continuator.html)** , and experimented with numerous interactions with jazz musicians, as well as children (see **the Miror project (http://www.francoispachet.fr/miror/miror.html)**). We showed indeed that interacting with a system that manipulates images of oneself creates novel and effective ways to boost creativity. The Flow Machines project pushes these ideas further by proposing a radically new way of looking at content creation tools.

The key technical idea of Flow Machines is to relate the notion of creativity to the notion of "style". Style is what makes an author (composer, writer, painter, etc.) recognizable, different from his peers. The assumption we make is that the key to being different is to invent a unique way to do things, a singular style. Of course, inventing a style is a difficult, life-long process, and so far only few individuals managed to invent styles that stand out in their respective fields: Picasso, Shakespeare, Mozart, Paul McCartney to name a few. Studying how these people came to invent their style suggests that this is basically a process of manipulating the styles of other creators to create new objects until something interesting emerges: Picasso has drawn many conventional bulls before he invented his abstract style which is so prevalent today (see the video about **the project**).



How can a machine understand style and turn it into a computational object? An object that users can manipulate in turn to create artefacts in that style, that also satisfy their own constraints. Technically we have been investigating new combinatorial optimization techniques to this aim, including constraint programming. We started from the well-known Markov models, routinely used to model statistical properties of temporal sequences. Markov models are used everywhere, from economics to Google ranking algorithms, and are good at capturing local properties of temporal sequences and abstract them into well-known mathematical concepts (e.g., a transition matrix). A Markov model can easily be estimated from a corpus of sequences in a given style to represent information about how musical notes follow each other. This model can then be used to produce new sequences that will sound more or less similar to the initial sequences. Generation is extremely simple and consists in so-called random walks (also called drunken walks): starting from a random state, transitions are drawn randomly from the model to build a sequence step by step. However, the remarkable simplicity of random walks in Markov models meets its limitations as soon as one tries to impose specific constraints on the generated sequences. The difficulty arises when one wants to impose simple properties that cannot be expressed as local transition probabilities. For instance, imposing that the last note equals the first one, or that the notes follow some pattern, or that the total duration be fixed in advance. Even more difficult, how to impose that the generated melody is nicely "balanced", for instance exhibiting a well-known 1/f property [1], characteristic of natural phenomena?

The initial idea was to use CP precisely to represent Markovianity, so that other, additional properties could also be stated as constraints. This is the main idea of constraint programming: constraints can be added at will to the problem definition, without changing the solver, at least in principle. The idea of representing Markovianity as a global constraint is detailed in (*Pachet and Roy, 2011*). We reformulate Markov generation as a constrained sequence generation problem. This idea works in practice and enabled us to produce remarkable examples. For instance, the AllDifferent constraint (introduced by J.-C. Régin), added to a Markov constraint could produce our "Boulez Blues": **a Blues in the style of Charlie Parker so that all chords are different!** However, this approach is costly, and we did not propose any boundaries on the worst-case complexity. So we started to look for efficient solutions for specific cases.

The first result we obtained concerned unary constraints: we showed they could be handled in polynomial time. The capacity to impose specific values (or domains) at specific locations in the sequence and get solutions with their correct probabilities (sampling) in polynomial time (*Pachet et al., 2011*) enabled us to do many interesting things: **the Virtuoso system**, as well as **harmonizations of arbitrary leadsheets**, and lyric generation (satisfying rhymes and prosody constraints, *Barbieri et al., 2012*). The Miror software is also based on this algorithm.

The next problem to solve was *meter*: notes have durations and there are many constraints in music concerning the sum of these durations. Meter cannot be expressed easily by index positions in a sequence. As a consequence, one cannot simply generate Markov sequences with a fixed duration for instance, which is problematic for music composition. Thanks to a theorem by Khovanskii in additive number theory, we found a pseudo-polynomial solution for meter and Markov sequences. Meter enables the generation of all Markov sequences satisfying constraints on partial sums of durations. This important result (*Roy and Pachet, 2013*) is heavily used in our systems, notably for lead sheet generation "in the style of" with arbitrary constraints (*Pachet and Roy, 2014*).

Another step was to address a recurring problems of Markov chains: increasing the order leads to solutions which contain large copies of the corpus: how to limit this effect? MaxOrder was introduced in (*Papadopoulos et al., 2014*) precisely to solve this problem. The solution consists in reformulating the problem as an automaton, using the framework of regular constraints. We proposed a polynomial algorithm to build this automaton (max order imposes a set of no goods to the Markov sequences).

An intermediate but beautiful result was to revisit the classical result of Voss & Clarke concerning 1/f distribution in music. We looked for a constraint that biases a sequence so as to its spectrum is in 1/f. We showed that the stochastic, dice-based algorithm proposed by Voss can be expressed as a tree of ternary sum constraints, leading to an efficient implementation (*Pachet et al., 2015*). For the first time, one can generate meter sequence in 1/f. Note that the examples in the original Voss paper did not have bars, understandably: now we can add them!

The next class of problems we address now is sampling: how to get not only all the solutions of the constraint problem, but a distribution of typical sequences, and for more powerful graphical models than Markov, notably to handle polyphonic sequences.

Works on sampling led to a remarkable result: all regular constraints added to Markov constraints can be sampled in polynomial time (*Papadopoulos et al, 2015b*). Moreover, Meter (and maxOrder) can be expressed as a regular constraint (as introduced by Pesant). This result somehow closes the chapter "Markov + X" global constraints initiated in 2011, since most interesting constraints in music and text can be expressed as a regular constraint.

Concerning the extension to more powerful models, we are exploring the use of maximum entropy models for music (*Sakellariou et al., 2015*). This model is based on binary relationships between events, thereby preventing issues related to parameter explosion inherent to higher-order Markov models. Departing from a pure filtering approach, parameter estimation is performed using high-dimension gradient search, and generation using a Metropolis algorithm. This model also leads to easier extensions as it is not limited to sequences. Notably, current investigations with multi-voice harmonization (in the style of Bach chorals) are under way, with already beautiful results (see the Bach choral orchestration of Ode to Joy in **the video**).

Many other aspects of style capture and generation are considered in this project, such as the generation of audio accompaniments in "in the style of". A dynamic programming approach is used in conjunction with a smart "audio gluing" mechanism that preserves groove, i.e. small deviations in the onset of events that characterize the style. An example can be heard in the Bossa nova orchestration of Ode to joy in the video (the 2nd one in **the video**). This result started an ongoing collaboration with Brazilian colleagues to capture **Brazilian guitar styles**.

Tool-wise, we are now finishing the implementation of authoring tools for musical composition and text writing that enable people to generate content by manipulating the style of an existing author, possibly themselves (see **a video** about a song composed in the style of Bill Evans). Suppose you want to create a song; that is, a melody with an underlying harmony. You can imagine that you choose some notes or pattern you like, then you can ask the system to “fill in the blanks” with the style of your preferred composer, such as Paul McCartney or Tom Jobim. The system proposes a sequence in the style of the author you chose and that satisfies your constraints (preferred notes, imposed harmonies, etc.). If you don’t like the proposed sequence the system suggests another one, and so on until you are satisfied with the result.

This fascinating and challenging project has now passed its mid-term. After having investigated core algorithmic problems, we are now entering an experimental stage in music and text. Current projects involve the production of several music albums in different styles, so stayed tuned!

Selected bibliography:

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

- *[Pachet et al., 2011] Pachet, F., Roy, P. and Barbieri, G.*
  Finite-Length Markov Processes with Constraints.
  Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI, pages 635-642, Barcelona, Spain, July 2011 ↩

- *[Barbieri et al., 2012] Barbieri, G., Pachet, F., Roy, P. Degli Esposti, M.*
  Markov Constraints for Generating Lyrics with Style.
  In Luc De Raedt et al., editor, Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012), pages 115-120, 2012 ↩

- *[Roy and Pachet, 2013] Roy, P. and Pachet, F.*
  Enforcing Meter in Finite-Length Markov Sequences.
  Bellevue, Washington (USA), June 2013 27th Conference on Artificial Intelligence (AAAI 2013) ↩

- *[Pachet and Roy, 2014] Pachet, F. and Roy, P.*
  Imitative Leadsheet Generation with User Constraints.
  21st European Conference on Artificial Intelligence (ECAI 2014), pages 1077-1078, Prague (Czech Republic), August 2014 ↩

- *[Papadopoulos et al, 2015a] Papadopoulos, A., Pachet, F. and Roy, P.*
  Generating non-plagiaristic Markov sequences with max order Sampling.
  Creativity and Universality in Language, Springer. 2015 ↩

- *[Papadopoulos et al., 2014] Papadopoulos, A., Roy, P. and Pachet, F.*
  Avoiding Plagiarism in Markov Sequence Generation
  July 2014 28th Conference on Artificial Intelligence (AAAI 2014), pages 2731-2737, Quebec (Canada) ↩

- *[Pachet et al., 2015] Pachet, F., Roy, P., Papadopoulos, A. and Sakellariou, J. 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)* ↩

- *[Papadopoulos et al, 2015b] Papadopoulos, A., Pachet, F., Roy, P. and Sakellariou, J. Exact Sampling for Regular and Markov Constraints with Belief Propagation. Cork (Ireland), 2015 21th Principles and Practice of Constraint Programming Conference (CP 2015)* ↩

- *[Sakellariou et al., 2015] Sakellariou, J., Tria, F., Loreto, V. and Pachet, F. Maximum Entropy Model for Melodic Patterns. Paris (France), July 2015 ICML Workshop on Constructive Machine Learning* ↩

References:

- The project's web site: **www-flow-machines.com** .
- A page on Markov constraints: **Markov constraints.**
- The video of the 7 reorchestrations of "Ode to Joy", presented at the European Parliament for the signature of the 5000th ERC grantee: **video.**

Pink noise or 1/f noise is a random signal, filtered to have equal energy per octave. It occurs in sound, music and countless human artefacts or natural events, from the fluctuations of the flood levels of the Nile to movements of the stock market. ↩