

# A Generative Representation for the Evolution of Jazz Solos

Kjell Bäckman\*, Palle Dahlstedt  
IT University, Gothenburg, Sweden

**Abstract.** This paper describes a system developed to create computer based jazz improvisation solos. The generation of the improvisation material uses interactive evolution, based on a dual genetic representation: a basic melody line representation, with energy constraints (“rubber band”) and a hierarchic structure of operators that processes the various parts of this basic melody. To be able to listen to and evaluate the result in a fair way, the computer generated solos have been imported into a musical environment to form a complete jazz composition. The focus of this paper is on the data representations developed for this specific type of music. This is the first published part of an ongoing research project in generative jazz, based on probabilistic and evolutionary strategies.

## 1 Introduction

The most important feature of a good jazz musician is to be able to keep an entire solo together as an entity, i.e. to build up the solo phrase by phrase in collaboration with the other musicians, where each phrase is a natural continuation of the previous one and leads up to a climax of intensity. After the climax the solo should be rounded off. A longer solo might contain several climaxes, but they should then be organized in a musically meaningful way. A good improviser is not expected to drop the focus and give way to meaningless cascades of notes or producing routine phrases for lack of artistic ideas. The challenge is to be able to plan the structure of the solo already from start, and then stick to the plan during the entire solo. There are some excellent examples in the history of jazz with this ability, such as John Coltrane, Miles Davies, Keith Jarrett, Bill Evans, Claes Crona and possibly some more.

This project aims at making the computer build up a solo based on these principles. This is done using evolutionary principles on a genome structure consisting of a raw melody line split up into small melody fragments (delta phrases) and a structure of operators applied hierarchically on the delta phrases. Initially, the raw melody line is built up according to a “rubber band” principle, where each pitch interval is constructed using energy constraints much like the tension of a stretched rubber band. After application of the operators, the delta phrases will be somewhat different, however hopefully preserving some kind of musical idea. The aim is to ensure logical development of a consistent material during the solo and thus reflect the feature of a well-planned solo.

There are others working with similar concepts. Al Biles [1] has developed a system, GenJam, which uses phrases played by the “master” soloist as basis for the

computer played solo. GenJam has three types of improvisation: whole chorus, chase improvising and collective improvising. The two last types are typical for older jazz forms like New Orleans jazz and the Swing era, while the first type is more relevant to modern jazz forms from the second half of the 20<sup>th</sup> century. In GenJam the function of listening to motives from fellow musicians has been solved by means of an Analog-To-MIDI converter device. However, GenJam does not have the grand format principle of building up a solo from a low intensity level to a climax and rounding it off at the end, which is the long-term aim of our research.

Francois Pachet [9] has developed a system where the user plays a melodic material, from which the system builds its solo according to Markov chain probability calculation. The user can at any time introduce new melodic material to which the system responds. The sounding result is remarkably good and does not suffer from any technical instrumental constraints. Pachet's system sounds like a well-trained musician.

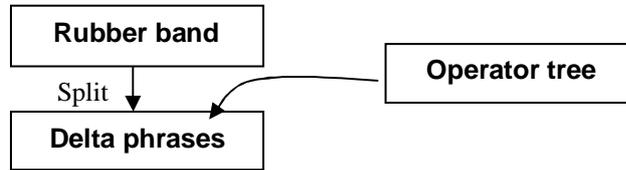
Dahlstedt [3][4] uses small melody fragments and combines them using operators in a recursively generative tree structure. Thywissen [13] uses generative music grammars in his GeNotator project. Dahlstedt and Thywissen apply their theories to classical music composition. In our experiment we try to apply similar theories to improvised jazz music. Also Manning's [8] exploration of MIDI technologies and Dean's [5] work on hyperimprovisation have been valuable.

Robert Rowe has in his two volumes, *Interactive Music Systems* [10] and *Machine Musicianship* [11] some interesting features such as scales connected to certain chord types, which have been valuable for the development of this system.

## 2 The Algorithmic Process

A jazz solo in this project consists of the melodic raw material and a hierarchical structure of operators. The melodic raw material can be auditioned separately. It is split into small portions, delta phrases, to allow processing at a lower level. Each operator of the operator structure processes a single delta phrase. For instance, one operator can add a note to the delta phrase, another operator can transpose the delta phrase a stipulated interval, and still another operator can invert a delta phrase. After processing of the delta phrases by the operators the result will be a modified melody possible to play back. The operators are applied hierarchically, i.e., one operator is applied to the whole melodic material, then one to each half, etc.

Fig. 1 shows how the raw material is split into delta phrases and then processed by the operators of the operator tree.



**Fig. 1.** Delta phrases are created by splitting the rubber band. The operator tree then processes the delta phrases.

The creation of the raw material and the application of the operators to the delta phrases are described in the subsequent sections.

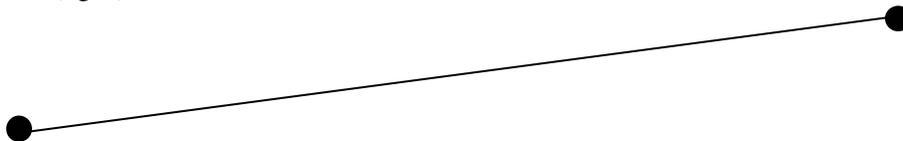
## 2.1 Rubber Band Principle

The rubber band principle utilizes the contrapuntal [12] aspect of consuming a certain amount of energy to make an interval jump. The larger interval, the more energy is required. This is also to some extent depending on the length and volume of the destination note.

A maximum amount of energy for the whole melody line is allowed to be consumed. Having spent much energy, the melody can collect new energy by making less energy-consuming movements for a while, which is accumulated to the available energy reservoir. This will make the melody go up and down in intensity.

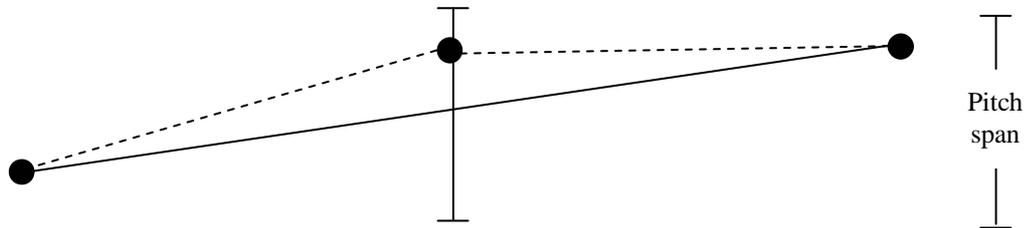
The creation of the raw material uses a similar technique as used in the mid-point displacement algorithm for landscape generation. It is created by originating from start and end pitches, then dividing the interval recursively. The middle pitch is stored for each interval division. It is represented as a deviation from the mean between the start and end pitch. So the representation is a binary tree of deviation figures from the mean line.

The raw material is created as follows. First we calculate a start pitch and an end pitch (fig. 2).



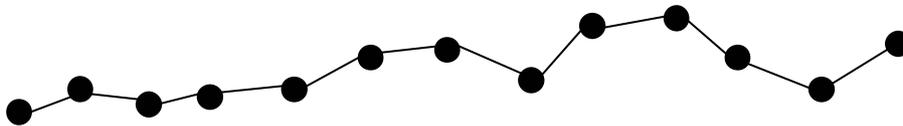
**Fig. 2.** The start and end points of the rubber band.

Then we generate a pitch in the middle of the time span, which is allowed to deviate from the mean pitch line by a maximum pitch span (fig. 3) achieved by experimentation.



**Fig. 3.** The middle point is created within the allowed span.

Then we split the time span into two equal time spans and repeat the process of generating a new note in the middle of each time interval (fig. 4).



**Fig. 4.** The complete rubber band.

For each recursive subdivision, the maximum allowed pitch span is reduced by a certain factor, also achieved by experimentation.

For each new note the required energy is calculated, which is dependent on the deviation from the mean pitch line, the length of the new note, and the volume of the new note. Notes with a high energy are delayed some microseconds to reflect the situation when a real musician prepares himself prior to making the big pitch jump. A big pitch jump is also prepared, according to contrapuntal theory [12], by playing some ornamentation around the source note before making the big jump. The bigger the pitch jump, the more ornamentation is played. Also, according to contrapuntal theory, the gap between the source and destination pitches is filled with further notes after the big jump in order not to leave an empty hole in the melody.

The pitches can be accommodated to a given chord progression, which is created by a separate evolutionary process, to be described elsewhere.

So far we have talked about the rubber band principle in connection with pitches. But the rubber band principle is also applied to note lengths and volumes. For instance, a start note length and an end note length are generated. The middle note length of the melody interval is selected with a deviation from the mean length. The deviation must be within the allowed length span, which also is reduced by a certain factor each time the interval is divided. This has the effect that a series of notes will have about the same length, however by modifying the length span factor, this can be adjusted to achieve sudden burst outs of short notes.

The calculation of note lengths is not quantized to the standard rhythmical values whole notes, half notes, fourths, eighths, triplets etc. The lengths can have any MIDI ticks value. The reason for this is to not being tied up to traditional musical thinking concerning rhythm, but to concentrate on melody shapes and intensity fluctuations. This will however provide a free-rhythmic feeling separated from any beat. As an option, when applying the melodies to a jam session situation, we have created a

function for accommodation of the rhythms to standard values of fourths, eighths, sixteenths, triplets etc.

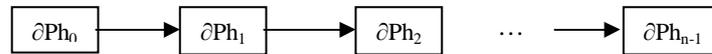
For dynamics, the rubber band principle is applied similarly; a start note volume and an end note volume are generated. The middle note volume of the melody interval is selected with a deviation from the mean volume. The deviation must be within the allowed volume span, which also is reduced by a certain factor each time the interval is divided. By modifying the volume span factor you can achieve more or less smooth volume shapes.

By combining the rubber band principles for pitch, length and volume we achieve pitch shapes, length shapes and volume shapes operating independent of each other.

The technical representation of the raw material is MIDI pitch, length and volume for the start note and end note. The contour is represented as a tree structure of relative values of pitch, duration and volume (relative to the mean of the end points of the current time subdivision), which when applied recursively will recreate the exact contour.

## 2.2 Delta Phrases

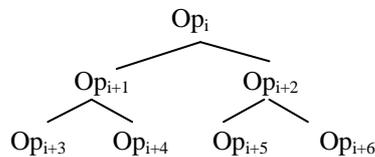
When the raw material has been created it is split into delta phrases. A delta phrase is a series of notes with pitch, length and volume. The number of notes per delta phrase is given by the number of notes in the rubber band divided by the number of delta phrases, which is a global parameter. Suppose we get  $n$  delta phrases,  $\partial Ph_0 - \partial Ph_{n-1}$  (fig. 5).



**Fig. 5.** Division of the rubber band in delta phrases

## 2.3 Operator Tree

The purpose of organizing operators hierarchically into an operator tree is to allow each delta phrase to be processed hierarchically by a series of operators. Fig. 6 shows the structure of an operator tree.



**Fig. 6.** Structure of an operator tree.

The operator at the top level is applied to all delta phrases. The operators at level 2 are applied to half of the delta phrases each. The 4 operators at level 3 are applied to  $\frac{1}{4}$  of the delta phrases. The division by 2 for each level is continued until there is one

single operator to each delta phrase. The effect of this is that each delta phrase is processed by a series of operators from top to bottom of the operator tree, one operator per level. Since each operator performs the same operation to each delta phrase, this process introduces conformity over the whole solo, and as the recursive process branches out, variation is introduced between sections.

Each operator modifies a delta phrase in one particular way. The options are given in table 1.

**Table 1.** Operator options.

Copy, leave the delta phrase unmodified.
Transposition of the entire delta phrase by a random number of halftones
Transposition of a random single note by a random number of halftones
Addition of a note at a random position of the delta phrase within the pitch interval given by the delta phrase
Removal of a randomly selected note in the middle of the delta phrase
Augmentation of each interval in the delta phrase by a random number of halftones
Diminuation of each interval in the delta phrase by a random number of halftones
Retrograde, the delta phrase is reversed
Inversion, the pitches of the delta phrase are mirrored around its average pitch
Rhythm modification of a randomly selected tone. The amount of time to add/delete is randomly selected
Volume modification, the volume of a loud note is decreased and vice versa
Note length modification, the length of a long note is decreased and vice versa
Insertion of a rest of random length at the end of the delta phrase
Repetition of part of the delta phrase. Delta phrase is divided into three segments, and one of them is repeated
Polyphony, the highest pitch of the delta phrase is calculated, then some notes later an extra note is added, a random number of halftones higher than that note.
Pitch bend, if a slope of five ascending intervals is followed by three descending intervals, the top note will be subject to pitch bend, which is performed by starting a half tone below and sliding up to the top pitch

The composition of operators in the operator tree is randomly created based on probability percents per each type of operator in table 1.

After all operator levels have been processed, each delta phrase has been modified by a series of operators from top to bottom. The effect is that two adjacent delta phrases have been processed by a similar series of operators and consequently should have some features in common.

The purpose of a process we call operator tree imbalance is to acquire a more varied application of operators to the delta phrases. It has been implemented by moving a part of the operator structure to another node of the tree, thus achieving a deeper operator level in some parts of the operator tree (fig. 7). The amount of imbalance (how many pair of operators to be moved) is controlled by a global parameter.



Fig. 7. Operator tree imbalance.

With the aim of acquiring melody fragments according to some kind of ABA form, the experiment has been equipped with “trinary” functionality besides the binary structure of the operator tree, which means grouping of operators three by three, where the third operator is set equal to the first. This means that the first operator in a pair of operators is set equal to the first operator in the previous pair. This is made at the bottom level of the operator tree only. The amount of “trinary” (the number of times to do this) is controlled by a global parameter.

To incur more operator processing to a delta phrase than accomplished by one series of operators from top to bottom of the operator tree, a delta phrase is allowed with a certain probability to “jump back” in the operator tree and follow another branch of operators (fig. 8). The frequency of doing this is controlled by a global parameter.

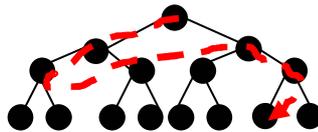


Fig. 8. The operator tree jump-back mechanism.

## 2.4 Evolution of solos

The generative representation described above has been used in an interactive evolutionary application, where a population of solos (typically about ten) are auditioned, selected and reproduced. The selection is a manual process using our personal preference, which stems from our background as jazz musicians. Solos can be saved at any time as MIDI files. Stored solos can be brought back into the process for further breeding. The genetic operators are described below.

A genome consists of the raw melody material and the operator tree. For each new generation two genomes (raw material melodies and the corresponding parent operator trees) are combined by crossover and mutated to generate a set of child genomes.

The two parent raw material melodies are combined by selecting start and end notes from the two parents interchangeably. Child no 1 will then get the deviation figures for pitch, length and volume for the first of two halves from parent 1 and for the last of two halves from parent 2. Child no 2 will get the figures for the first of two halves from parent 2 and for the last of two halves from parent 1. Child no 3 and 4 will get their figures from parent 1 and 2 interchangeably per each of 4 parts. Child no

5 and 6 will get their figures per each of 8 parts and child no 7 and 8 will get their figures per each of 16 parts.

A mutation is performed to the child raw material melody by increasing the deviation figure for pitch by 3 halftones up or down depending on whether the deviation is positive or negative, the deviation figure for MIDI note length by 12 ticks and the MIDI volume by 28 units, depending on whether the deviations are positive or negative. This is done for six notes in a sequence starting on a random position of the rubber band.

The new operator trees are combined by extracting branches from the two parent trees. This is done by originating from the parent 1 operator tree and copying random branches from parent 2 operator tree into parent 1 operator tree until 50% of the operators have been copied. When an operator in the operator tree of parent 2 is selected, all operators of the sub-tree under that operator will also be copied.

Mutations are applied to the child operator trees by modifying a random operator by the “jump back” series of operators, and applying slight random changes to the parameters of the operators (transposition, rest length, etc.), with a certain probability.

This gives a specific operator tree and a specific raw material melody per child.

### 3 Results

Some sounding examples are provided, where the title of each sound file gives an indication of the basic parameter setting. The manual selection of children has been carried out based on personal preference, which in turn springs from our background as jazz musicians. About 5-15 generations have been processed for each parameter setting with a population of 10 children per generation.

The sounding output is a continuous flow of small motives hooked onto each other, now and then interrupted by small rests inserted into the flow. You can trace the rubber band-like shapes of pitch, note length and volume individually, which in combination cause the intensity fluctuations. Thus the melody has some kind of intensity curves rolling up and down, trying to imitate melody curves of good music in general and jazz improvisation solos in particular.

The sounding output provides an interesting sequence of thematic material according to a slowly developing process, giving a feeling of recognition since each delta phrase has some kind of resemblance with adjacent delta phrases.

There is also an interesting polyrhythmic feature sometimes caused by a repetitive sequence of an odd number of notes not matching the natural beat (3 against 4, 5 against 4 etc.), and sometimes also caused by accentual volume effects to individual notes.

The “trinary” functionality did not provide the expected result to acquire some sort of ABA form, so the functionality could probably be dismissed without any loss of musical quality. The tree imbalance functionality also has a doubtful impact on the quality and could probably be excluded without any recognizable negative impact. The “jump back” functionality, on the other hand, provides a richer variation of the sounding output and deserves to be kept and further developed.

A set of unaccompanied sound examples can be heard at this link; in General MIDI format:

<http://oden.ei.hv.se/genmel>

A link to an example with computer generated drums and bass accompaniment is also available:

[http://oden.ei.hv.se/genmel/midi\\_evolv1.mid](http://oden.ei.hv.se/genmel/midi_evolv1.mid)

## 4 Conclusions

Does this system provide any valuable artistic material? Yes, at least some sounding examples are of interest, maybe not of high professional musician class, but provide interesting and unexpected artistic output.

An improviser often uses standard phrases and motives trained during a long time of practicing and concerting. He relies on routines built up through repeated usage of similar muscular movements that are well accommodated to the physical design of the instrument. Some musicians are very strongly tied up to this behaviour, which makes them sound somewhat cliché-like and limited. There are some examples of this kind of musician in jazz history. They use the same kind of motives whatever style they play in, and may sound technically very brilliant and swinging, but if you transcribe their solos and try to play them with a good fingering, you will realize that they are astonishingly easy to learn to play.

The main purpose of using computers to produce jazz improvisation is that it opens your mind to new thinking and frees you from old habitual paces of playing. Hopefully it can enrich your improvisation style with new kinds of musical material.

The process of listening to many children in each generation may be boring after some time and it is easy to loose your concentration of separating between musically meaningful melodies and not so meaningful. Another problem is that the musical properties you are looking for might not be exactly the same in the beginning of a session as it is later in the session. Therefore, an intelligent setting of basic parameters and consistent evaluation conditions are required. The basic idea with evolution is that it works over many generations, but a concentrated listening to 10 children of more than 10 generations is probably not possible. So an automated fitness procedure would be valuable. The work of developing a computer based automatic fitness procedure has been started, but is not included in this paper.

Since the rubber band principle is applied to note lengths with no restrictions as concerns the traditional note values of whole notes, half notes, fourths, eighths etc., you cannot trace any particular beat or tempo, which is critical to all jazz music performed in jam session groups. This is easily remedied by quantization of the note lengths to the nearest traditional note value according to some logic, and thereby maybe also taking bar lengths in account. But by omitting this regulation, it is easier to concentrate on the real melodic value and not be distracted by side effects like swing or rhythmic effects.

## 5 Future Work

As mentioned, an automated fitness function would enable us to utilize the full strength of the evolutionary process, including a large population and many generations. Development has started, with promising results, which will be published elsewhere in the near future.

Since the a final goal of this project is to use the generated melodies as improvised solos in jam sessions with both virtual instruments and acoustic instruments, it will be necessary to accommodate the rhythm to the beat of the tune being played, and also to the periodicity and chorus lengths.

Evolutionary algorithms can also be used to produce new harmonies, drum rhythms, walking bass figures, piano and guitar accompaniment chord arrangement, and basic tune themes. This work has already been initiated, and will be published in the future.

Communication between musicians is very important in live jazz music. It would be possible to implement this in a computer generated jam session by letting the soloist, the drummer and the accompanying pianist “listen” to each other and reuse motives and rhythmic accents. Hopefully this will render not only a communicative feature but also a feeling of collective improvisation, where no particular soloist is leading the others but instead a situation where all musicians have the same value and contribute to the musical result on an equality basis.

## 6 References

1. Biles, J.A. (1994) GenJam: a genetic algorithm for generating jazz solos. Proceedings of the 1994 International Computer Music Conference. ICMA, San Fransisco, pp. 131-137.
2. (anonymized) (2006) Evolutionary Jazz Improvisation. Masters thesis.
3. Dahlstedt, P. (2004) Sounds Unheard of – Evolutionary algorithms as creative tools for the contemporary composer. PhD thesis, Chalmers University of Technology, Gothenburg.
4. Dahlstedt, P. Autonomous Evolution of Complete Piano Pieces and Performances, MusicAL Workshop, ECAL 2007, Lisbon, Portugal, September 10-14, 2007, Proceedings (Workshop CDROM), 2007
5. Dean, T. (2003) Hyperimprovisation: Computer-Interactive Sound Improvisation. A-R Editions Inc., Middleton, Wisconsin.
6. Levine, M. (1989) The Jazz Piano Book. SHER MUSIC CO. Petaluma, CA, USA.
7. Levine, M. (1995) The Jazz Theory Book. SHER MUSIC CO. Petaluma, CA, USA.
8. Manning, P. (2004) *Electronic and Computer Music*. Oxford University Press, New York, USA.
9. Pachet, F. (2002) Interacting with a Musical Learning System: The Continuator. SONY-CSL, Paris, France. <http://www.csl.sony.fr/~pachet> (Accessed 2 March 2006).
10. Rowe, R. (1993) Interactive Music Systems. The MIT Press, Cambridge, Massachusetts, USA.
11. Rowe, R. (2001) Machine Musicianship. The MIT Press, Cambridge, Massachusetts, USA.
12. Söderholm, V. (1980) Arbetsbok i kontrapunkt. Eriks. Cop., Stockholm
13. Thywissen, K. (1996) GeNotator: An environment for investigating the application of generic algorithms in computer assisted composition. In *Proceedings of International Computer Music Conference 1996 (ICMC96)*, pp. 274-277, Hong Kong.