

Evolutionary Algorithms and Automatic Transcription of Music

Gustavo Reis
School of Technology and
Management
Polytechnic Institute of Leiria
Portugal
gustavo.reis@ipleiria.pt

Francisco Fernández
University of Extremadura
Spain
fcofdez@unex.es

Aníbal Ferreira
University of Porto
Portugal
ajf@fe.up.pt

ABSTRACT

The main problem behind Automatic Transcription (Multiple Fundamental Frequency - F0 - Estimation) relies on its complexity. Harmonic collision and partial overlapping create a frequency lattice that is almost impossible to deconstruct. Although traditional approaches to this problem rely mainly in Digital Signal Processing (DSP) techniques, evolutionary algorithms have been applied recently to this problem and achieved competitive results. We describe all evolutionary approaches to the problem of automatic music transcription and how some were improved so they could achieve competitive results. Finally, we show how the best evolutionary approach performs on piano transcription, when compared with the state-of-the-art.

Categories and Subject Descriptors

H.5.5 [Sound and Music Computing]: Signal analysis, synthesis and processing;; I.2.m [Artificial Intelligence]: Miscellaneous

General Terms

Algorithms

Keywords

Automatic Music Transcription, Pitch Estimation, Multiple F0 Estimation, Genetic Algorithms

1. INTRODUCTION

Automatic Music Transcription, often called *Multiple Fundamental Frequency (F0) Estimation* or *Multi-Pitch Estimation*, consists of having an algorithm that extracts and identifies all the musical notes from a given acoustic signal. This is very difficult problem that only the most skilled musicians can address. Since the first works by Moorer [13] and Piszczalski & Galler [15], polyphonic music transcription systems

almost always rely on the analysis of information present in the frequency domain. Klapuri [7], for instance, uses iterative calculation of predominant F0s in separate frequency bands and Martin [10] uses blackboard systems. There have also been applied techniques that use the principles of human auditory organization for pitch analysis, as the work of Kashino et al. [6] by means of a Bayesian probability network, where bottom-up signal analysis can be integrated with temporal and musical predictions, and Wamsley et al. [22, 23], who use the Bayesian probabilistic framework to estimate the harmonic model parameters jointly for a certain number of frames. The usage of a Hidden Markov Model and spectral feature vectors was proposed by Raphael [16] to describe chord sequences in piano music signals. Carreras et al. [2] used Neural Networks for spectral-based harmonic decompositions of signals. Marolt [9] used networks of adaptive oscillators to track partials over time. Ortiz et al. [14] used a physical model of the piano to generate spectral patterns and compare them to the incoming spectral data.

Although there have been several applications of Genetic Algorithms [5] to Signal Processing [1], Evolutionary Algorithms have almost no applications to Automatic Music Transcription.

The rest of this document is structured as follows: Section 2 overviews the Evolutionary Approaches to the problem of Automatic Transcription of Music and Section 3 presents our conclusions..

2. EVOLUTIONARY APPROACH

It is important to emphasize that the main idea behind a Genetic Algorithm [5] is to have a set of candidate solutions (individuals) to a problem evolving towards the desired solution. In each generation those individuals are evaluated according to their quality (fitness). The worst individuals are then discarded and the best will generate new individuals resulting from the combination of their parent's characteristics (genes) and minor variations (mutation). This way, individuals with better quality tend to live longer and to generate better and fitter offspring, thus improving the robustness of the algorithm. Moreover, when addressing a genetic algorithm to a problem there are several aspects that must be taken into account:

Genotype How to encode each individual or candidate solution to the problem.

Fitness Function How to evaluate the quality of each candidate solution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.

Copyright 2012 ACM 978-1-4503-1178-6/12/07...\$10.00.

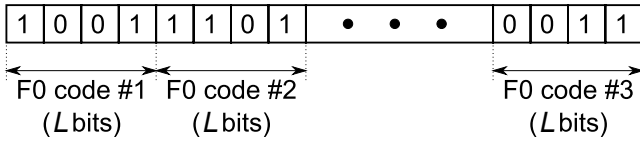


Figure 1: Garcia's approach chromosome structure with $L = 4$ bits.

Selection How individuals are selected from the population to breed.

Recombination How to employ recombination: given two individuals, how to exchange genetic material between them to breed two new individuals (offspring).

Mutation What kind of mutations we should take into account, according to the problem being solved.

Initialization How the first population is generated.

Survivor Selection How survivors are selected from one generation to the next.

During the entire section of this document we will describe how each approach found in the literature addresses these topics.

2.1 First Genetic Algorithm approach to Polyphonic Pitch Detection

The first work in the literature using Genetic Algorithms for polyphonic pitch detection appears in 2001 by Garcia [3]. Garcia claims that polyphonic pitch detection can be considered as a search space problem where the goal is to find the pitches that compose a polyphonic acoustic signal. This way, it makes sense to use genetic algorithms since they perform very well in search problems [4].

Genotype.

Garcia's approach encodes each chromosome as a binary string with variable length (see Fig. 1). The chromosome's structure is a concatenation of N substrings of L bits each. Each substring encodes one F0 value using binary fixed-point representation. Although the length of the L substrings is fixed since F0 range and resolution is specified as an input parameter, the length of the chromosome is variable because no assumption is made about the number of F0s in the signal. The length of the substrings is defined according to the frequency range, $\Delta F0$, and frequency resolution, $dF0$, as the minimum integer L , where:

$$2^L \geq \frac{\Delta F0}{dF0} \quad (1)$$

Fitness Function.

The fitness measure of Garcia's approach [3], $f(s)$, for a given string or chromosome, s , is based upon a correlation between the input spectrum and a comb spectrum defined in [11]. The partial fitness value $f_p(s, j)$ is computed for each fundamental frequency value, $F0^j$, coded by substring j in string s , as the correlation between the input magnitude spectrum $|X(\omega)|$ and a reference comb spectrum with exponentially decreasing amplitudes $e^{-\alpha h}$, where h is the harmonic index and α a specified input parameter:

$$f_p(s, j) = \sum_h |X(2\pi h F0^j)| \cdot e^{-\alpha h} \quad (2)$$

After the partial fitness $f_p(s, j)$ is computed for a substring j , the input DFT bins used in the correlation sum are zeroed for the remaining partial fitness evaluations of the string. This way, each spectral bin is constrained to belong to only one harmonic series. This strategy penalizes strings or chromosomes that contain correct F0 values along with spurious multiples or submultiples. For each chromosome a raw fitness value, f_{raw} , is then calculated as the sum of partial fitnesses over all its j substrings:

$$f_{raw}(s) = \sum_{j=1}^{N_s} f_p(s, j) \quad (3)$$

The chromosome fitness, $f(s)$, is then computed from the raw fitness as:

$$f(s) = f_{raw}(s) - N_s \bar{f}_p \quad (4)$$

where $\bar{f}_p$ is the mean partial fitness over the whole population:

$$\bar{f}_p = \frac{\sum_{s,j} f_p(s, j)}{\sum_s N_s} \quad (5)$$

and where N_s is the number of F0s or substrings in the chromosome. The subtraction by $N_s \bar{f}_p$ in Equation 4 is a way to penalize strings with too many F0 codes (it is equivalent to subtracting the average partial fitness from each partial fitness) since substrings with partial fitness values smaller than average will become negative and then will penalize the global fitness of the chromosome. Strings with any F0 value outside the allowed range are assigned null fitness.

A final fitness correction step is applied to prevent the premature convergence of the genetic algorithm. This is employed by imposing a fitness floor value F_{min} , such as:

$$F_{min} = \frac{F_{max}}{\beta} \quad (6)$$

where F_{max} is the maximum fitness in the current generation, and β is an input positive constant. Individuals whose $f(s) < F_{min}$ have their fitness reset at $f(s) = F_{min}$.

Selection.

Each individual is selected for breeding according to the roulette wheel [4] selection operator: for each individual in the population a roulette wheel slot is assigned, which size is proportional to its fitness $f(s)$. Garcia implements the roulette wheel as an array of partial cumulative fitnesses:

$$f_c = \sum_{j=1}^s f(j) \quad (7)$$

where a uniformly distributed random number r between zero and the total cumulative fitness $f_c(M)$ is drawn and then minimum string index s that satisfies the $f_c(s) > r$ condition is chosen.

Recombination.

As recombination operator, Garcia uses the single-point crossover. This operator is designed as follows: two differ-

ent points of cut are selected - one per individual - since the number of encoded F0s can differ from individuals. This way, two individuals with different chromosome sizes can breed and generate two offspring also, with different chromosome sizes. This operator also ensures that the chromosomes length of the offspring are always multiples of L .

Mutation.

The mutation operator consists on flipping single bit in the whole genome of an individual. The probability of mutation (P_n) is given by:

$$P_n = (1 - P_m)^{(N \cdot L)} \quad (8)$$

where P_m is the probability of mutation per bit and $(N \cdot L)$ is the chromosome length.

Initialization.

The initial population is composed by randomly generated individuals: random number of F0s, each with a random F0 value. Both maximum number of F0s and F0 frequency range are specified inputs.

Survivor Selection.

Each new generation consists of individuals selected from the previous generation. This selection is made using the roulette wheel selection operator. Afterwards both recombination (one point crossover) and mutation are applied. Finally, if the current best individual is not as fit as the best individual of the previous generation, the current worst individual is replaced by the best from the previous generation. This strategy is called elitism [4].

Additional Constraints.

Note that this approach does not have in consideration: onset, offset and also dynamics. The algorithm can tell which are the fundamental frequencies present on an audio signal but is unable to detect where those pitches start, where do they end and which are their dynamics.

2.2 Moving from Polyphonic Pitch Detection to Automatic Music Transcription

Despite Garcia's approach [3] being able to work with almost any frequency and resolution, Lu [8] considers that a polyphonic audio signal is made of by the 128 possible pitches (from the low C, frequency 8.18 Hz to a high G, 12543.88 Hz) defined in MIDI specification [12], therefore an audio signal can have up to 128 specific frequencies.

Genotype.

Traditionally, solutions or chromosomes are represented as binary strings of 0s and 1s (such as in Garcia's [3] approach), but other encodings are also possible. Lu encoded each individual as a hierarchical structure, which is not far from the internal representation of a MIDI file [12] (see Fig. 2). Each individual is made of several sequences of notes, organized as tracks, according to each instrument. Each note has frequency, a start time and length. Both start time and length are truncated to time slices. For instance: for transcribing a set of eighth notes, the eighth note should be the time slice.

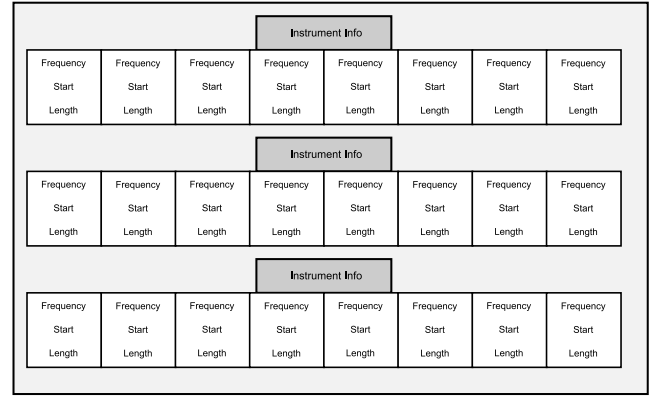


Figure 2: Genotype of Lu's approach: notes are separated according to each instrument/track. Each note has frequency, start time and length. Start and length are truncated to time slices.

Fitness Function.

To evaluate each individual or transcription each transcription is rendered using additive synthesis into an audio signal, which will be compared with the original audio. The result of the comparison (distance) is the fitness value of the corresponding individual. To avoid problems like phase, Lu [8] proposes that the distance between each individual's transcription and the original audio should be measured in the frequency domain. The fitness function, similar with the euclidean distance, is defined as:

$$Fitness = \frac{1 - \sum_{t=0}^{tmax} \sum_{f=fmin}^{Rochelim} (O(t, f) - X(t, f))^2}{\sigma} \quad (9)$$

where $O(t, f)$ is the magnitude of frequency f at time t of the original audio, $X(f, t)$ is the same for the individual's transcription and σ acts as a scaling factor, equivalent to the first worst transcription, putting all fitnesses values between [0,1].

Selection.

Although the author of this approach claims that he is addressing music transcription using genetic algorithms, his approach does not use recombination, which is the main pillar of genetic algorithms [4]. The approach relies exclusively on mutations. This way, individuals are not selected for reproduction.

Recombination.

The author of this approach claims that "the genetic material found inside high-fitness individuals is good enough such that most of the material is at least partially correct". According to Lu, the removal of this material for addition into another individual is detrimental for the donating individual. This way, recombination was not included in the approach.

Mutation.

Lu applies a roulette selection to determine which mutation will be applied to each individual. The main purpose of this roulette wheel is having some mutations being performed more often than others. The proportions of the

roulette wheel also change over time so that mutations that perform small incrementational changes are more often applied during the last generations of the algorithm.

This approach uses the following mutations:

Irradiate Randomly changes one feature (pitch, start time or end time) of a gene.

Nudge Similar to Irradiate, except that changes are on the smallest amount possible: pitch is changed by one semitone, and both start and end time are changed by one time slice.

Lengthen Adds a random musical note to the chromosome.

Split Inserts silence into an encoded musical note. This mutation is capable of deleting a note by inserting a silence with the length of the selected note, shortening a note by inserting the silence on its end or even split the note into two notes by inserting the silence in the middle.

Reclassify Moves a section of the chromosome to a different spot in the chromosome. This mutation allows a set of multiple notes being changed from one instrument to another.

Assimilate Takes a section of the chromosome from one individual and copies it to another individual.

Initialization.

The initial population is generated by randomly generated individuals: random number of notes, each with a random start and duration.

Survivor Selection.

The top third of the population are copied and then mutations are applied on those copies. These new individuals replace the bottom third (less fit) of the population.

Additional Constraints.

Note that this approach does not have into account the dynamics of each note. For the synthesis process, instead of using sample based techniques as Reis et al.[20], Lu[8] uses simple and very-well known mathematical models like the sine, square, sawtooth and triangle waves. Therefore, this approach is only able to deal with sounds generated by those mathematical models. Also, the input audio files are MIDI files synthesized with the same synthesizers used inside the genetic algorithm, which makes turns the transcriptions much easier to find and without the problem of harmonic overfitting [20].

2.3 Automatic Music Transcription using Synthesized Instruments

In 2007 Reis and Fernandez [17] proposed a new genetic algorithm approach to automatic music transcription, using synthesized instruments. Unlike Lu's method [8] the synthesized instruments were not simple mathematical models (sine, sawtooth and triangle waves) but, instead, synthesized instruments (piano and vibraphone). Reis and Fernandez [17], similarly as Garcia [3], proposed a genetic algorithm with recombination, mutation and crossover operators for pitch detection. The latter approach only takes into account

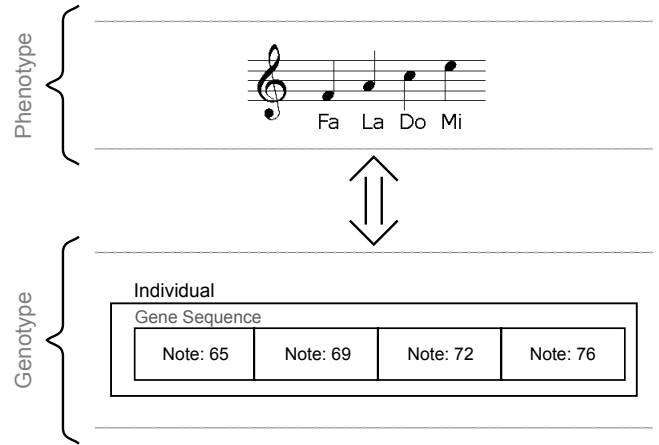


Figure 3: Reis and Fernandez encoding for monophonic transcription. The individual is divided in time frames, where each time frame has can be one of 128 possible MIDI pitches plus the option of silence.

the possible 128 MIDI pitches, just as it happens with Lu's [8] algorithm.

Genotype.

Reis and Fernandez [17] started proposing a system for monophonic pitch detection and later upgraded it to support polyphonic audio signals, just with minor adjustments. The encoding for the monophonic transcription task is based on the assumption that a signal is divided in several time frames, where there can be one of the 128 possible pitches plus the option of having a silence (see Fig. 3). For polyphonic transcription of music, the authors extended the previous encoding to support several pitches at the same time, as shown of Fig. 4.

Fitness Function.

Similarly as Lu's proposal [8], the evaluation of each individual is done in the frequency domain, to avoid phase problems, using the STFT. To compare each MIDI-like individual with the target acoustic signal, each individual is rendered into an audio signal using additive sound synthesis. Reis and Fernandez [17] implemented a synthesizer with the respective oscillator and envelope for the synthesis process. The fitness evaluator renders each MIDI-like individual converting it in an audio signal and then computes it's fitness value by summing the difference between each frequency in each time slice of the song:

$$Fitness = \sum_{t=0}^{tmax} \sum_{f=0}^{fmax} (|O(t, f)| - |X(t, f)|)^2 \quad (10)$$

where $O(t, f)$ is the magnitude of frequency f at time slot t in the acoustic audio signal, and $X(t, f)$ is the same for each individual. Fitness is computed from time slot 0 to $tmax$, traversing all time from the beginning to the end, and from $fmin = 0$ Hz to $fmax = 22050$ Hz, which is the nyquist frequency of 44100 Hz sample rate.

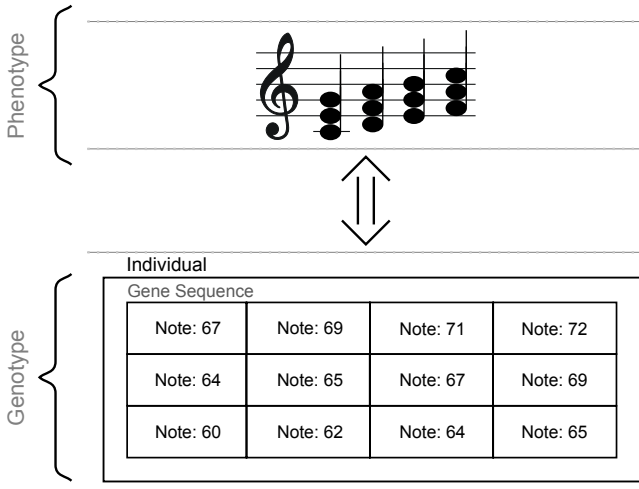


Figure 4: Reis and Fernandez encoding for polyphonic transcription. The individual is divided in time frames, where each time frame has can be one of 128 possible MIDI pitches plus the option of silence.

Selection.

Individuals are selected for breeding with the deterministic tournament [4]. The size of the tournament is 5.

Recombination.

The offspring is generated by applying the classic one-point crossover [4] on each pair of parents.

Mutation.

The authors only implemented a simple mutation that changes the pitch of a random note by -1,1 semitone.

Initialization.

As in all the previous works by other authors [3, 8] the initial population consists of random generated individuals.

Survivor Selection.

The new individuals generated by recombination and mutation are added to the population. Then, the N most fit individuals (where N is the initial population size) are selected for the next generation.

Additional Constraints.

Although this is able to deal with polyphony, it is not able to deal with multiple instruments as Lu's [8] algorithm. This approach cannot also work with note dynamics.

2.4 First Approach on Real Audio Recordings

The first genetic algorithm approach for polyphonic music transcription in the literature dealing with real audio data and real instruments appeared in late 2007 by Reis et al. [20]. Since there were, to the moment, three different genetic algorithm approaches to automatic music transcription [8, 17] and polyphonic pitch estimation [3] Reis et al. [20] decided to propose a standard and generic genetic algorithm approach to the problem, which emphasized several important considerations like: genotype, fitness eval-

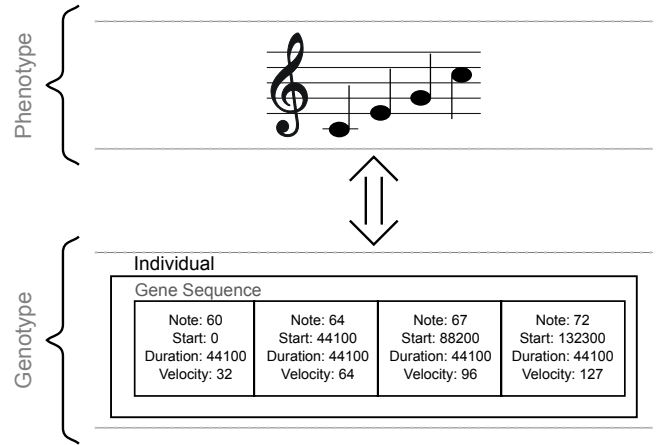


Figure 5: Encoding of the individuals proposed by Reis et al. 2007 [20]. The individual is encoded as a set note events. Each event has a pitch, start time, duration and velocity.

uation, recombination, mutation and how to generate the initial population.

Genotype.

Each individual or chromosome corresponds to a candidate solution (transcription), therefore it is made of a sequence of note events. The number of genes (note events) varies from one individual to another. Each gene has all the information needed to represent that note event: note onset, duration, dynamics and also an instrument/timbre associated with that event, if necessary. Fig. 5 shows the proposed encoding by Reis et al.

Fitness Function.

For the evaluation of an individual, each note event passed through an internal synthesizer which consisted using previously recorded piano 30 seconds samples from a Korg SP100 Piano Keyboard at the made MIDI velocity: 64. The release of the note (decay after releasing the note key) was created applying the following equation:

$$R(t) = \max \left(0, \frac{2000.0 - \frac{t}{36}}{2000.0 + t} \right) \quad (11)$$

Both original and synthesized streams are cut in time frames with 4096 samples, with an overlap of 75%. To decrease the spectrum leakage, a Hanning window was applied on each frame, before the STFT. The frequency spectrum was limited to the range from F0 of the first note to the F0 of last note of the piano's keyboard (from MIDI-note 21 - 27,5 Hz - to MIDI-note 108 - 4186Hz). The fitness function used by Reis et al. is defined by:

$$Fitness = \sum_{t=0}^{tmax} \sum_{f=Note21}^{Note108} ||O(t, f)| - |X(t, f)|| \quad (12)$$

Selection.

As in Reis and Fernández 07 [17], the selection for breed-

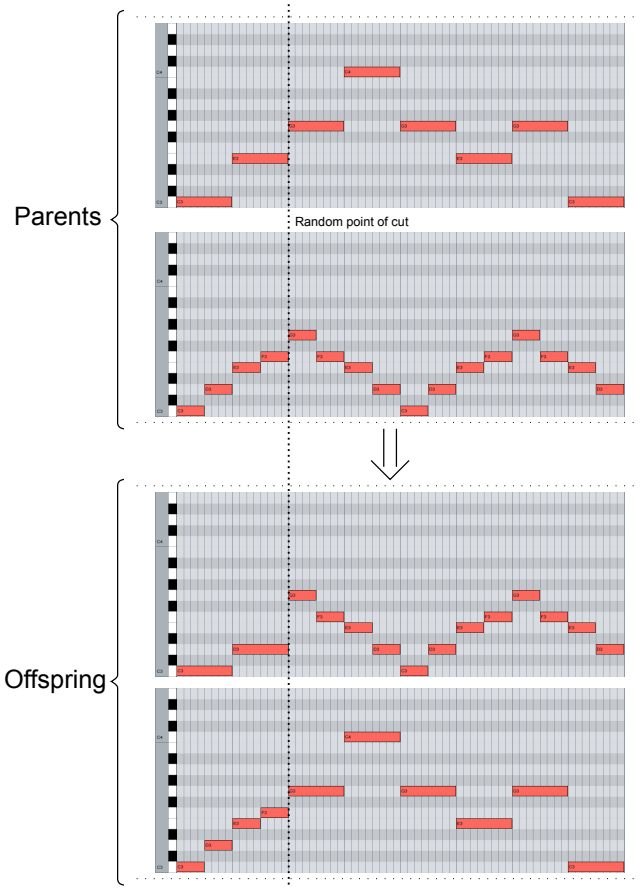


Figure 6: One point crossover performed on temporal dimension.

ing is based on the deterministic tournament, with size of 5.

Recombination.

The recombination operator proposed by these authors is based on the classic one point crossover [4]. Instead of choosing a random point of cut in the individuals chromosome, this recombination operator, chooses a random point of cut on temporal dimension. This happens because despite the individuals might differ in the number of genes, they have the same temporal length. The randomly selected point of cut in time will split any note events that cross the chosen time value (see Fig. 6).

Mutation.

Reis et al. implemented several mutation operators:

- note change ($\pm$ octave, $\pm$ half tone);
- start position (up to ± 0.5 second change);
- duration (from 50% to 150%);
- velocity (up to ± 16 in a scale of 128);
- event split (split in two events with a silence between);
- event remove;

- new event (random event or duplication with different note).

Initialization.

For the starting population, these authors created a first individual based on the highest peaks of the STFT on the original audio signal. The highest peaks, during each time frame, create (or maintain) a musical note with the corresponding Fundamental-frequencies (F0). Afterwards, this individual goes through an hill-climber process that changes all the events equally in terms of velocity, duration and start time to adjust durations and velocity to overcome decay and level differences between the original instrument and the internal synthesizer. Each additional individual in the initial population is created based on the initial individual after 10 forced mutations.

Survivor Selection.

5% of each new generation are created based on mutations of the best individual of the previous generation. All remaining individuals are the most fit of the previous generation.

2.5 Reducing the Harmonic Overfitting

In the previous proposed Genetic Algorithm approaches to Polyphonic Music Transcription, Reis et al. [20] noticed that the genetic algorithm tends to create additional notes (with lower amplitudes) in harmonic locations of the original notes to overcome the timbre differences between the internal samples and original piano sounds. Despite the fitness values continues to decrease through generations, the quality of their results started to decrease after some point, mainly because of a harmonic overfitting. Detected notes continued there (shown by recall values) but many additional notes begin to emerge, dropping the precision value.

The fact of these additional notes have low amplitude and are in harmonic locations, many times even with similar onsets, strongly decreases their impact from the perception point of view. Nevertheless, for the metrics or in situations where the dynamic information is discarded (for instance: creating music sheets), these errors are very undesirable.

Genotype.

To avoid the problem of harmonic overfitting, Reis et al. [21] extended their previous approach [20] by creating harmonic gains that boost or cut the value of the 20 first harmonic peaks of each synthesized note. Those gains act almost like an equalizer but instead of operating in fixed frequency bands, they operate on each note harmonic. In practice, this is not done with real filters but instead by applying different weights on the STFT bins belonging to the note harmonic series. This way, each individual, besides having a sequence of note events, as its candidate solution to the problem, also includes additional parameters to help the internal synthesizer to get a timbre more similar with the original instrument (see Fig. 7). The gain of the fundamental frequency of each note - F0 - is always set to 1 and its deviation is always set to 0. As for inharmonicity, the amount of shifting for each harmonic of the harmonic structure was also encoded within the Individual's genotype. This enables each individual to have its own synthesizer, with a complete evolving harmonic structure. This way the har-

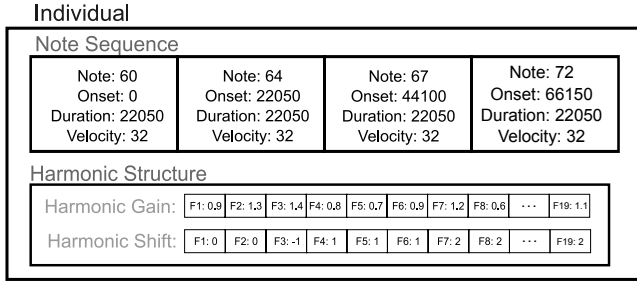


Figure 7: Encoding of the Individual with the Harmonic Structure.

monic structure of the internal synthesizer will evolve until it matches the synthesizer played on the original audio, and the note events will evolve towards the original song’s notes.

Fitness Function.

Due to the natural logarithmic scale of musical notes, STFT bins are not equally distributed by all octaves (eg.: the highest octave occupies the highest half of frequency bins). To reduce the higher impact of higher notes and, thus, reduce the harmonic overfitting, the authors performed a division by f for frequency normalization:

$$Fitness = \sum_{t=0}^{tmax} \sum_{f=27.5Hz}^{\frac{fs}{2}} \frac{||O(t, f)| - |X(t, f)||}{f} \quad (13)$$

Recombination.

By extending the individual genotype for inclusion of the harmonic gains and shifts, the recombination operator had also to be extended to support these additional chromosomes: the note events are still recombined using the one point crossover on the temporal dimension and both harmonic series and harmonic shifting are recombined using the classic one point crossover [4].

Mutation.

Two new mutations were included to support the additional chromosomes:

- harmonic change (up to ± 0.50 gain);
- inharmonicity deviation (up to ± 3 frequency bins).

Note discard.

Another feature proposed by Reis et al. [21] as a means to avoid the harmonic overfitting is note discard. Note discard is based on the assumption that most notes have similar dynamics. By considering that each note has dynamic scale between 1 and 128 (MIDI velocity range), this feature discards all notes present that have a dynamic difference of 20 between the their dynamics and the dynamics of the other notes existing during the note duration.

Dynamic Range.

Harmonic overfitting can also happen due to noise, weak harmonics or even frequency neighborhood. Dynamic range

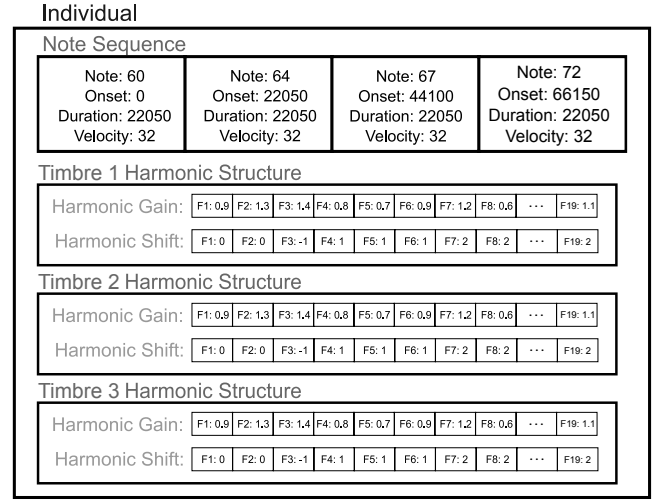


Figure 8: Encoding of the Individual with the Harmonic Structure for Multi-Timbre support.

feature uses the highest value of the STFT bins of the current frame as a reference, and sets all bins of the same frame with values 40bBs below this reference to 0.

2.6 Automatic Music Transcription of Multi-Timbral Music

The approach presented by Reis et al. [21] was for transcription of polyphonic piano music. In 2009, the same authors extended their previous approach to deal with other kinds of pitched instruments, such as: trumpet, saxophone, clarinet and trombone. Basically, both approaches are exactly the same, except that the individuals chromosome now includes the spectral envelope and its inharmonicity deviation for each different timbre.

Genotype.

The harmonic structure of each internal synthesizer was encoded inside the individuals genome (see Fig. 8) to avoid the harmonic overfitting in each instrument or voice.

2.7 Genetic Algorithm Achieves State-of-the-Art Results

In 2012, Reis et al. used all their knowledge on applying Genetic Algorithms to the Automation Transcription of Music [17, 20, 19, 21, 18] to implement a new and written from scratch Genetic Algorithm for polyphonic piano music transcription. This approach takes advantage of spectral envelope modeling and dynamic noise level estimation to aid the transcription process: while the noise is dynamically estimated, the spectral envelope of the internal synthesizers is adapted to best match the piano played in the input signals.

The authors also performed a benchmark where different state-of-the art algorithms were evaluated and their results was compared. This comparison was made using three different metrics. The Genetic Algorithm proposed by Reis et al. ranked as 2nd best algorithm on 2 metrics and as best algorithm on the remaining metric. Results of the benchmark can be seen and heard on:

<http://www.estg.ipleiria.pt/~gustavo.reis/benchmark>.

3. CONCLUSION

Although traditional approaches to Automatic Transcription of Music do not rely on Evolutionary Algorithms, we have shown that Genetic Algorithms are fit to the problem and can achieve competitive results among the state-of-the-art.

4. ACKNOWLEDGMENT

The authors acknowledge the support of Spanish Ministry of Science and Innovation under project ANYSELF (TIN2011-28627-C04), Gobierno de Extremadura, under projects GRU09105, GR10029 and Municipality of Almendralejo.

5. REFERENCES

- [1] J. T. Alender. An indexed bibliography of genetic algorithms in signal and image processing. report 94-1-SIGNAL, University of Vaasa, Department of Information Technology and Production Economics, 1995.
- [2] C. F. Automatic harmonic description of musical signals using schema-based chord decomposition. *Journal of New Music Research*, 28:310–333(24), December 1999.
- [3] G. Garcia. A genetic search technique for polyphonic pitch detection. In *Proceedings of the International Computer Music Conference (ICMC)*, Havana, Cuba, Sept. 2001.
- [4] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.
- [5] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, April 1992.
- [6] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka. Organization of hierarchical perceptual sounds: Music scene analysis with autonomous processing modules and a quantitative information integration mechanism. In *Proc. International Joint Conf. on Artificial Intelligence*, pages 158–164, 1995.
- [7] A. Klapuri. Multiplitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Transactions on Audio and Language Processing*, 16(2):255–264, February 2008.
- [8] D. Lu. Automatic music transcription using genetic algorithms and electronic synthesis. Computer Science Undergraduate Research, University of Rochester, New York, USA.
- [9] M. Marolt. Networks of adaptive oscillators for partial tracking and transcription of music recordings. *Journal of New Music Research*, 33:49–59(11), March 01, 2004.
- [10] K. D. Martin. A blackboard system for automatic transcription of simple polyphonic music. Technical report, Tech. Rep. 385, MIT Media Lab, Perceptual Computing Section, July 1996.
- [11] P. Martin. Mesure de la fréquence fondamentale par intercorrélation avec une fonction peigne. Technical report, JEP, 1981.
- [12] MIDI Manufacturers Association. *The Complete MIDI 1.0 Detailed Specification*, September 1995.
- [13] J. A. Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, 1(4):32–38, 1977.
- [14] L. Ortiz-Berenguer, F. Casajus-Quiros, and S. Torres-Guijarro. Multiple piano note identification using a spectral matching method with derived patterns. *Journal of the Audio Engineering Society*, 53(1/2):32–43, January/February 2005.
- [15] M. Piszczalski and B. A. Galler. Automatic music transcription. *Computer Music Journal*, 1(4):24–31, 1977.
- [16] C. Raphael. Automatic transcription of piano music. In *Proceedings of ISMIR 2002*, 2002.
- [17] G. Reis and F. Fernandez. Electronic synthesis using genetic algorithms for automatic music transcription. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1959–1966, New York, NY, USA, 2007. ACM Press.
- [18] G. Reis, F. Fernandez, and A. Ferreira. Transcripción de música multi-timbre mediante algoritmos genéticos. In *MAEB 2009 VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, February 2009.
- [19] G. Reis, N. Fonseca, F. F. de Vega, and A. Ferreira. Hybrid genetic algorithm based on gene fragment competition for polyphonic music transcription. In *EvoWorkshops*, volume 4974 of *Lecture Notes in Computer Science*, pages 305–314. Springer, 2008.
- [20] G. Reis, N. Fonseca, and F. Fernandez. Genetic algorithm approach to polyphonic music transcription. *Proceedings of WISP 2007 IEEE International Symposium on Intelligent Signal Processing*, pages 321–326, 2007.
- [21] G. Reis, N. Fonseca, F. Fernandez, and A. Ferreira. A genetic algorithm approach with harmonic structure evolution for polyphonic music transcription. In *The 8th IEEE International Symposium on Signal Processing and Information Technology*, December 2008.
- [22] P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner. Bayesian graphical models for polyphonic pitch tracking. In *In Diderot Forum*, pages 1–26. Morgan Kaufmann, 1999.
- [23] P. J. Walmsley, S. J. Godsill, and P. J. W. Rayner. Polyphonic pitch tracking using joint bayesian estimation of multiple frame parameters. In *in Proc. IEEE Workshop on Audio and Acoustics, Mohonk*, 1999.