Hybrid Genetic Algorithm Based on Gene Fragment Competition for Polyphonic Music Transcription

Gustavo Reis[1], Nuno Fonseca[1], Francisco Fernández de Vega[2], and Anibal Ferreira[3]

[1] School of Technology and Management
Polytechnic Institute of Leiria, Portugal
{gustavo.reis,nfonseca}@estg.ipleiria.pt
[2] University of Extremadura, Spain
fcofdez@unex.es
[3] University of Porto, Portugal
ajf@fe.up.pt

Abstract. This paper presents the Gene Fragment Competition concept that can be used with Hybrid Genetic Algorithms specially in signal and image processing. Memetic Algorithms have shown great success in real-life problems by adding local search operators to improve the quality of the already achieved "good" solutions during the evolutionary process. Nevertheless these traditional local search operators don't perform well in highly demanding evaluation processes. This stresses the need for a new semi-local non-exhaustive method. Our proposed approach sits as a tradeoff between classical Genetic Algorithms and traditional Memetic Algorithms, performing a quasi-global/quasi-local search by means of gene fragment evaluation and selection. The applicability of this hybrid Genetic Algorithm to the signal processing problem of Polyphonic Music Transcription is shown. The results obtained show the feasibility of the approach.

Keywords: Polyphonic Music Transcription, Evolutionary Algorithms, Genetic Algorithms, Memetic Algorithms, Intelligent Recombination Operator, Gene Fragment Competition.

1 Introduction

Although Genetic Algorithms (GAs) are very good at rapidly identifying good areas of the search space (exploration), they are often less good at refining near-optimal solutions (exploitation). For instance: when a Genetic Algorithm (GA) is applied to the "OneMax" problem[1], near-optimal solutions are quickly found but convergence to the optimal solution is slow. Therefore hybrid GAs using local

[1] The OneMax problem is a binary maximization problem, where the fitness function is simply the count of the number of genes set to "1".

M. Giacobini et al. (Eds.): EvoWorkshops 2008, LNCS 4974, pp. 305–314, 2008.

search can search more efficiently by incorporating a more systematic search in the vicinity of "good" solutions[1]. For instance: a bit-flipping hill-climber could be quickly applied within each generation for the OneMax to ensure fast convergence. Memetic Algorithms (MAs) are a class of stochastic global search heuristics in which Evolutionary Algorithms-based approaches are combined with local search techniques to improve the quality of the solutions created by evolution[1]. This means that Memetic Algorithms go a further step by combining the robustness of GAs on identifying good areas of the search space with local search for refining near-optimal solutions. Recent studies on MAs have revealed their successes on a wide variety of real world problems[1]. Particularly, they not only converge to high quality solutions, but also search more efficiently than their conventional counterparts. In diverse contexts, MAs are also commonly known as hybrid EAs, Baldwinian EAs, Lamarkian EAs, cultural algorithms and genetic local search.

But a new problem with traditional local search operators arises due to the cost of evaluation. If the calculation of the fitness function is heavy, having local search operators changing each individual several times means lots of individual evaluations thus lots of computational cost. Therefore in problems that demand high computational cost in fitness evaluation this might be a prohibitive solution. Fitness evaluation allows us to measure the whole "quality" of each individual, and in many cases, it is obtained by simply combining the values of the evaluations of each gene or gene fragment. But when the evaluation of gene fragments is possible, these fragment values are not taken on consideration during recombination.

This paper presents a Gene Fragment Competition, a different approach to recombination, that takes advantage of gene/fragment evaluation and gene/fragment selection as a way to speed up the process, especially when evaluation of individuals is a demanding computational task. The presented method arises from the work of the authors on the field of Automatic Music Transcription using Genetic Algorithms. In this kind of approach the individuals evaluation demands lots of digital signal processing based on a high number of FFTs, which results on a heavy computational cost making impracticable the use of traditional memetic algorithms, but still needing a special operator to increase significantly the performance. The rest of the paper is structured in the following way: Section 2 presents our proposed approach, Section 3 describes the Polyphonic Music Transcription problem, the Genetic Algorithm approach to the problem and how gene fragment competition is applied, Section 4 presents our experiments and results and finally Section 5 summarizes our conclusions and future work.

2 Proposed Approach

2.1 Gene Fragment Competition

In the traditional GA approach (see Fig. 1a), genetic algorithms are based on a cycle made of evaluation, selection, recombination and mutation - individuals are

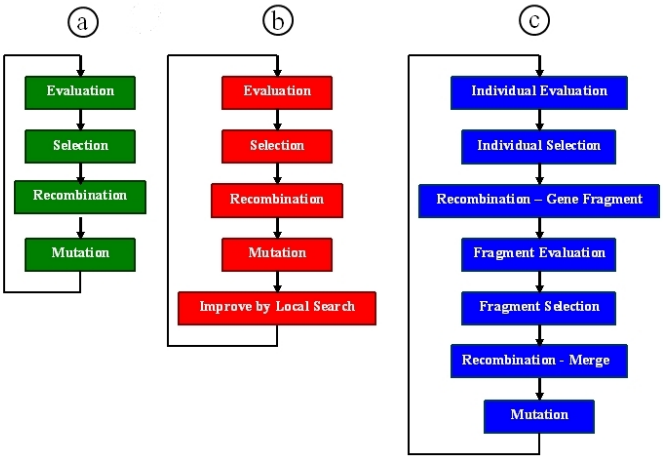

Fig. 1. Classic Genetic Algorithm approach (a) vs Traditional Memetic Algorithm approach (b) vs Gene Fragment Competition (c)

evaluated, based on their evaluation parents are selected for recombination, creating new individuals that are subject to mutation. On the other hand classical MAs apply a new local search operator in each individual just after the mutation (or even after recombination), looking for better solutions in the neighborhood of already found good solutions. Gene Fragment Competition uses a different kind of global/local search approach. Instead of using separate operations for global and local search, like the Memetic Algorithms, a different type of recombination is proposed which is responsible for a semi-global/semi-local search.

We can consider that traditional recombination operators are made of two operations: fragmentation (parent's genes are divided on two or more fragments), and merging (these gene fragments are merged to create new individuals). The main idea of the proposed method is to add two additional steps inside recombination: gene fragment evaluation and gene fragment selection. Parent genes are split on n fragments, each fragment is evaluated and then a selection method is applied to choose the best gene fragments, which will be merged to create a new born individual. To split the parents in n fragments two methods can be applied: static fragment size, on which equally sized fragments are created or dynamic fragment size where are created fragments with random sizes. For selecting gene fragments classic selection methods also apply (roulette, tournament, etc.). Although standard recombination operators breed two individuals from two parents, our method breeds only one individual from two or more parents.

As can be seen, a very important requirement must be fulfilled: it must be possible to evaluate gene fragments. If this is not possible, the method cannot be applied. This does not mean that the system must be able to evaluate individual genes, or every kind of group of genes. In some applications it simply means that some special type of fragmentation must be applied to make possible the evaluation of its fragment (see section 3.3). For instance, in signal processing

applications or image processing applications we can fragment the individual in time fragments or spatial fragments. In the cases when evaluation is a complex operation a cache feature is highly desirable, for quick evaluation of gene fragments, since its only a matter of adding partial fitness values.

2.2 Simple Example

Imagine that our goal is to create an individual which is an exact copy of a target sequence of integer numbers. Therefore the individual's encoding could be an array in integers, where each gene would correspond to each sequence number (see Fig. 2).

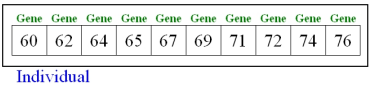

Fig. 2. Individual's encoding on the "Find the sequence problem"

Let's consider that the fitness value is obtained by the sum of the absolute differences (Equation 1) of each individual's gene ($X(i)$) and the our target individual ($O(i)$) (see Fig. 3).

$$Fitness = \sum_{i=1}^{genes} |O(i) - X(i)| \tag{1}$$

Important note: since the best individuals are the ones closer to our target and the fitness function is measuring that distance, the best individuals are the ones who have less fitness values.

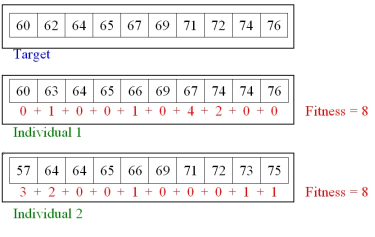

Fig. 3. Fitness values of Individual1 and Individual2. The fitness value of each individual is calculated by the sum of the absolute difference between the values of their genes and the target individual's genes.

If we want to breed a new Individual from the parents Individual1 and Individual2 with 2 random points of cut (dynamic fragment size) our intelligent recombination operator will calculate the fitness value of each fragment and then choose the fragment for the new born individual with the best fitness (see Fig. 4).

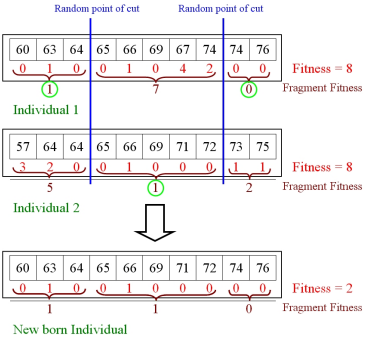

Fig. 4. Breeding of a new born individual. The best fragments of each father are inherited to the new born individual.

3 Polyphonic Music Transcription Problem

3.1 Automatic Music Transcription

Automatic Music Transcription is the process in which a computer program writes the instrument's partitures of a given song or an audio signal. Hence, automatic music transcription from polyphonic audio recordings is the automatic transcription of music in which there is more than one sound occurring at the same time: multiple notes on a single instrument (like a piano), single notes in multiple instruments, etc. (usually, only pitched musical instruments are considered). Music transcription is a very difficult problem, not only from the computational point of view but also in a musical view since it can only by addressed by the most skilled musicians.

Traditional approaches to Automatic Music Transcription try to extract the information directly from audio source signal (using frequency analysis, autocorrelation functions [2,3], and other digital signal processing techniques [4,5,6,7,8,9,10]). Nevertheless Automatic Music Transcription can be considered as a search space problem where the goal is to find the sequence of notes that best models our audio signal[11]. Instead of trying to deconstruct the audio signal, a search space approach will try to find a solution which allows the creation of a similar audio signal. Usually search space approaches are not addressed in music transcription problems due to the huge size of the search space. Nevertheless genetic algorithms[12] have proven to be an excellent tool to find solutions in extremely large search spaces, since they only need to use a very small subset of the entire search space.

3.2 Genetic Algorithm Approach

To apply genetic algorithm to the music transcription problem there are some important considerations regarding: the encoding of the individuals, the creation of the initial population, recombination and mutation and also the fitness function (as reviewed by Reis et al. [13]).

Each individual (chromosome) will correspond to a candidate solution, and is made of a sequence of note events (the number of notes will most likely be different from individual to individual, i.e.: the number of genes is not fixed). Each note, acting as a gene, will have the information needed to represent that note event (e.g. note, start time, duration, and dynamics).

For the starting population, an individual is created based on the peaks of the FFT - in each time frame, the frequency with the highest peak creates (or maintains) a note with the same fundamental frequency. The additional individuals of the population (199 of 200) are created based on the initial individual and after 10 forced mutations. Fig. 5 illustrates this process.

Fig. 5. Starting Population Process

There are several mutation operators: note change (±1 octave, ± 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 with a silent between), event remove, new event (random or event duplication with different note). Besides these event mutations, there are 2 mutation operators (with lower probability) that are applied equally to all events: velocity change (up to ± 4 in a scale of 128) and duration (from 50% to 150%).

To evaluate an individual, a synthesizer is used to convert the note sequence into an audio stream, which will be compared with the original audio stream. Therefore each note has to pass through an internal synthesizer which in our case is made of a simple sampler, using piano samples.

The original stream and the synthesized one are compared in the frequency domain. The streams are segmented in time frames with 4096 samples (fs = 44.100kHz) and an overlapping of 75%. A FFT is calculated for each frame and the differences are computed (Equation 2).

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

Table 1. Algorithm parameters

Parameters	**Values**
Population	100
Survivor Selection	Best 100 individuals (population size)
Crossover probability	75%
Mutation probability	1%
Note minimal duration	20 ms

The $|O(t, f)|$ is the magnitude of frequency f at time frame t in the source audio signal, and $|X(t, f)|$ is the same for each individual. The division by f acts as a frequency normalization. Fitness is computed from frame slot 0 to $tmax$, traversing all time from the beginning to the end, and from $fmin = 27,5$ Hz (corresponding to the first note of the piano keyboard) to $fmax = 22050$ Hz, which is half of the sample rate - 44100 Hz.

The GA approach is resumed on Table 1.

3.3 Applying Gene Fragment Competition to Music Transcription

To apply the proposed operator to the music transcription approach presented in [13] and summarized in section 3.2, there are some remarks. The requirement needed for gene fragment competition is that it must be possible to evaluate gene or gene fragments. In the music transcription approach presented before, each gene represents a musical note, and it is not possible to evaluate each note, especially in polyphonic parts. Nevertheless, there is a solution for that. The overall fitness of each individual is obtained by adding the FFT differences over the different time frames, which means that although it is not possible to evaluate note by note, it is possible to evaluate time frames (for instance, it is possible to evaluate the behavior of an individual on a time fragment between time=2.0s and time=4.0s). Then it is possible to map that time interval to the genes (notes) acting on that time fragment. If some notes on that fragment began before or end after the time frontiers, the note is split, and only the inside part are considered. Later, during the recombination merge phase, if a note ends on the exact same time that a similar one begins, notes are merged as one, since the algorithm considers that a previous split happened.

For global selection, the "deterministic tournament" method was used, but in fragment selection a "non-deterministic tournament" method was used as a means to preserve some biodiversity. Regarding fragment size, that in this case is measure in seconds since we consider time fragments, the value of 5 seconds was used. Increasing the fragment size should decrease the impact of the operator, and decreasing fragment size has the side effect of splitting too much the notes.

For each individual, a cache feature was implemented, that stored the fitness values for each time frame, which means that evaluating a time fragment is simply done by adding fitness values of its internal time frames.

4 Experiments and Results

To analyze the impact of the presented method, several tests were made. The proposed method was applied on our music transcription approach on an audio file with the first 30 seconds of the piano performance of the Schubert's Impromptu No.2 in E Minor. Each bank of tests was created with 1000 generations, and with at least 4 different runs[2] (the values presented correspond to the average values).

[2] The value differences between runs are very small ($< 3\%$).

Table 2. New parameters

Classic GA	
Selection	Deterministic Tournament (5 individuals)
Recombination	1-point time crossover (with eventual note split)
Gene Fragment	
Individual Selection	Deterministic Tournament (5 individuals)
Fragment Selection	Tournament (5 individuals)
Fragment size	5 seconds[3]

Table 3. Average fitness values over 1000 generations using classic GA, using static GFC and dynamic GFC

Generation	**GA (1%)**	**Dyn GFC**	**Static GFC**	**GA (0,5)%**	**GA (5%)**
250	1,808 E12	1,670 E12	1,718 E12	1,839 E12	1,882 E12
500	1,690 E12	1,554 E12	1,580 E12	1,708 E12	1,811 E12
750	1,621 E12	1,492 E12	1,512 E12	1,624 E12	1,765 E12
1000	1,571 E12	1,455 E12	1,464 E12	1,563 E12	1,728 E12

In the first bank of tests, there were 3 different scenarios: classic GA approach, static fragment size and dynamic fragment size of the proposed method (see Table 2). In these tests (and on the following ones), the presented values are the fitness values. Since the goal of the paper is not to evaluate our music transcription approach, presenting other results (% of transcribed notes, etc) would remove the focus from the goal of the paper.

Table 3 and left part of the Fig. 6 shows the fitness evolution (average fitness values over several different runs) over 1000 generations. Once again, it is important to recall that in our implementation, since fitness measure the error between FFT's, the lower values of fitness, means better individuals.

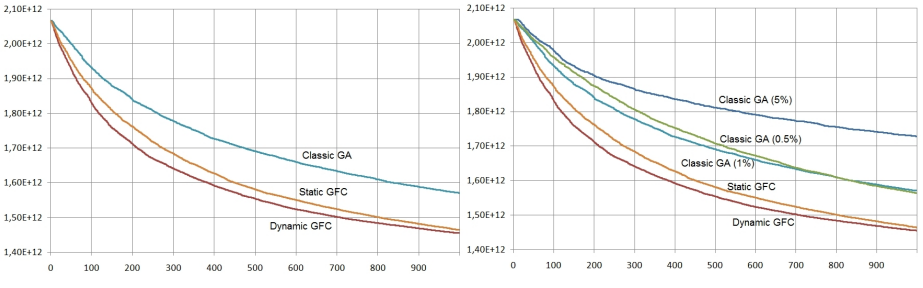

Fig. 6. Fitness values over 1000 generations using classic GA, using static fragment size gene competition and dynamic fragment size gene competition (left) and with different values of mutation probabilities in generic GA approach (right)

[3] Resulting in 6 fragments on 30 seconds audio files.

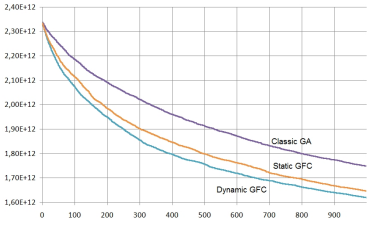

Fig. 7. Average fitness values over 1000 generations using classic GA, using static fragment size gene competition and dynamic fragment size gene competition for the Mozart's Piano Sonata

The first test, shows us that the proposed method achieves a better performance in this scenario. Nevertheless, there are other situations that needed to be tested in order to discard other hypothesis. One question that could rise is regarding the "Tournament" vs. "Deterministic Tournament" selection. A new run of tests were made using the classic GA, but changing the selection mode from "Deterministic Tournament" to "tournament". The obtained results were identical within a range $> 0.1\%$. The other question that could rise is related to mutation probabilities. Since the proposed method fragments the genes, could it be that changing the mutation probabilities of the classic GA could result in much better results? A new bank of tests was made with classic GA approach with different mutation probabilities (5%, 1% and 0.05%). Fig. 6(right) and Table 3 show the results. Using different mutation probabilities above (5%) or bellow (0.5%) didn't present significantly results.

Tests were made also with another audio file to confirm the earlier results. A 30s seconds audio file of Mozart's Piano Sonata n. 17 in B flat K570 was used and is presented in Fig. 7. Once again the proposed method presents an increase of performance. In this test the performance difference between static and dynamic fragment size also increases comparatively with the initial test.

It is important to say that in both tests by applying our operator with dynamic size, we have achieved in only 500 generations the same results the classical GA achieves in 1000 generations, which is a very significative gain in performance.

5 Conclusions and Future Work

This paper has presented a Gene Fragment Competition as a new technique for improving quality of results when applying GAs to several signal and image processing problems.

Although the proposed method presents some requirements that are not fulfilled on several GA applications (capability of evaluating fragment of genes), it is shown that at least in some scenarios can achieve an important performance increase.

In the future, additional tests must be made to have a better understanding of the impact of the operator applied on different problem applications.

Acknowledgments

This work was partially funded by National Nohnes project TIN2007-68083-C02-01 Spanish MCE.

References

1. Hart, W., Krasnogor, N., Smith, J.: Memetic Evolutionary Algorithms. In: Recent Advances in Memetic Algorithms, Springer, Heidelberg (2004)
2. Klapuri, A.P.: Qualitative and quantitative aspects in the design of periodicity estimation algorithms. In: Proceedings of the European Signal Processing Conference (2000)
3. Klapuri, A.P.: Automatic music transcription as we know it today. Journal of New Music Research 33(3), 269–282 (2004)
4. Marolt, M.: On finding melodic lines in audio recordings (2004)
5. Dixon, S.: On the computer recognition of solo piano music (2000)
6. Bello, J.P.: Towards the automated analysis of simple polyphonic music: A knowledge-based approach. PhD thesis, University of London, London, UK (2003)
7. Walmsley, P., Godsill, S., Rayner, P.: Bayesian graphical models for polyphonic pitch tracking (1999)
8. Walmsley, P., Godsill, S., Rayner, P.: Polyphonic pitch tracking using joint bayesian estimation of multiple frame parameters (1999)
9. Goto, M.: A robust predominant-f0 estimation method for real-time detection of melody and bass lines in cd recordings.
10. Gómez, E., Klaupuri, A., Meudic, B.: Melody description and extraction in the context of music content processing. Journal of New Music Research 32(1) (2003)
11. Lu, D.: Automatic music transcription using genetic algorithms and electronic synthesis. Computer Science Undergraduate Research, University of Rochester, New York, USA
12. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The MIT Press, Cambridge (1992)
13. Reis, G., Fonseca, N., Fernandez, F.: Genetic algorithm approach to polyphonic music transcription. In: Proceedings of WISP 2007 IEEE International Symposium on Intelligent Signal Processing, pp. 321–326 (2007)