A COMPARISON OF EVOLUTION STRATEGY-BASED METHODS FOR FREQUENCY MODULATED MUSICAL TONE TIMBRE MATCHING

Thomas J. Mitchell and Anthony G. Pipe
Faculty of Computing, Engineering & Mathematical Sciences, University of the West of England, Frenchay, Bristol, BS16 1QY, UK
thomas.mitchell@uwe.ac.uk

ABSTRACT

We evaluate two Evolution Strategy-based optimisation algorithms that are known to perform well with multi-modal fitness landscapes, comparing them with each other and a standard Evolution Strategy. We apply these algorithms to a simple FM synthesiser timbre matching problem, which exhibits rugged multi-modal characteristics. All three algorithms are shown to be capable of finding globally optimal solutions. The Fuzzy Clustering Evolution Strategy is both computationally expensive and slow to converge. However, it produces globally optimal results with very high probability, compared with the other two algorithms. In contrast, whilst both the Evolution Strategy and the Cooperative Co-Evolution Strategy are significantly less computationally expensive, the Cooperative Co-Evolution Strategy is significantly quicker to converge than the other two.

1. INTRODUCTION

There are now many complex synthesis methodologies, each of which is capable of producing a diverse range of timbres. Normally, the synthesiser interface is a reflection of the underlying synthesis process, rarely do the controls relate to sound in human terms. Consequently, there is often a large discrepancy between the dimensions of the synthesiser parameter space and the perceived sound space. This discrepancy renders most synthesisers difficult to control and often unintuitive to learn. Inexperienced users/programmers would benefit from a timbre-specifying procedure that relates to their mental picture of the sound. In other words, a synthesiser user could greatly benefit from being able to produce sounds that are like those desired, and then have the synthesiser search of the parameters that best match that sound, leaving the user to tune the sound thereafter. A process is therefore required that is able to map a generated sound onto sound synthesis parameters.

Earlier attempts have been made to transfer synthesiser control into a more intuitive domain (e.g. [3] [15]). The most promising recent developments utilise the optimisation principles of Evolutionary Computation (EC) for sound navigation and exploration [4] [5] [9] [12] [13] [14]. When EC is used, assistance is generally provided in one of two forms: interactive evolution, where the user controls the direction of the search as evolution takes place, or sound matching, where the evolutionary search attempts to find a close match to a given target sound. It is the latter that is of interest here.

In this paper, we build on our previous work [16] [17] in employing advanced forms of optimisation algorithms based on the Evolution Strategy (ES) in this application domain. More generally, our work builds upon the previous work, presented by Horner [6] [7], and has wider implications as a platform for a generic synthesiser interface that is not specific to the underlying synthesis type. In some of our other work, we have dealt with dynamic audio domains, i.e., those in which the sound to be matched varies with time [17]. However, for this paper, to ease accurate performance measurement of, and comparison between, three presented algorithms we have imposed the following limitations to the application domain;

- Only Frequency Modulation (FM) synthesiser parameters will be optimised.
- We will deal with only static-spectra
- We will generate the target sounds themselves, also with an FM synthesiser; we call these *contrived* sounds.

The work is, of course, not limited to such special cases; in real-world application it would be desirable to extract matches of non-contrived, time-varying tones from a variety of synthesis types. The limits are imposed to help us investigate the performance of the evolutionary algorithms in this domain.

2. EVOLUTIONARY SOUND MATCHING USING FREQUENCY MODULATION

2.1 Frequency Modulation Synthesis

FM audio synthesis, presented originally by Chowning [2], provides a synthesis method by which complex spectra can be created simply and efficiently. In what is termed simple FM, the instantaneous frequency of one oscillator is modulated by another, to produce a tone with multiple frequency partials. The amplitude function for simple FM is given by the formula;

$$e = Asin(\,_{c}t + Isin\,_{m}t) \qquad -(1)$$

Where e is the modulated carrier amplitude, A the peak amplitude of the carrier, $_c$ and $_m$ the carrier and modulator angular frequency, and I the modulation index, given by the ratio of the frequency deviation to

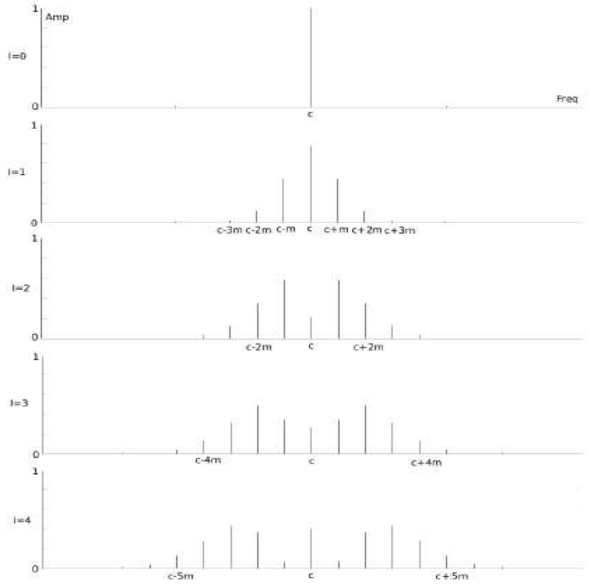

Fig. 1: Synthesised FM spectra with increasing modulation index *I*

the modulating frequency. Modulation produces side-bands in the frequency domain, with partials deviating from the carrier at integer multiples of the modulating frequency. The bandwidth of the output signal increases as the modulation index is raised, as can be observed in Fig. 1. Notice that as I is raised the amplitude of each partial varies according to a non-linear (Bessel) function. This can make it hard to achieve a target sound when altering parameters by hand. For further reading into the spectral decomposition of FM signals, the reader is referred to [11]. Chowning's basic FM arrangement is the one used for the experiments described here.

2.2 Evolutionary Frequency Modulation Sound Matching

To facilitate the matching of acoustic instrument tones, Horner's algorithm [6] [7] optimises a set of static basis-spectra generated via FM, which are recombined to simulate a given harmonic target tone. The synthesis process is therefore very close to that of wavetable synthesis, with FM used only in the production of basis-spectra. The basis-spectra are generated by a simple FM arrangement in which the modulator is tied to the fundamental frequency, and the carrier frequency is set to integer multiples thereof, known as formant FM. This arrangement is excellent for use in conjunction with wavetable synthesis, as bands of the target spectrum can be reproduced by separate basis-spectra, from which the optimum spectral envelopes can be established.

Restricting the carrier frequency to an integer multiple of the modulating frequency ensures that all of the basis-spectra are harmonic, and supports the use of a Genetic Algorithm (GA) for optimisation purposes: GAs perform their genetic operations on bit-strings, which make them ideal for integer based combinatorial search domains, such as this. Horner's wavetable FM model [6] cannot be applied directly to explore the sound space of regular FM, as it exploits an alternative synthesis paradigm. When the synthesis variables are not limited to integer numbers, the search may be performed across the entire parameter space, which may yield better timbre matches. This operation is a non-trivial process, as the FM object landscape is extremely complex and multi-modal.

Our early attempts with simple evolutionary optimisers, like the simple GA and basic ES proved insufficient for the FM matching problem. As a result, we have been following a programme of work in which we have been applying more specialised optimisation algorithms. In [16] we looked at the performance of an ES to this problem domain. In [17] we applied a more complex algorithm, called FCES, and describe briefly below and, in this paper, we compare performance of these two to a third algorithm we have developed, CCES, also described below.

3. DESCRIPTION OF ALGORITHMS

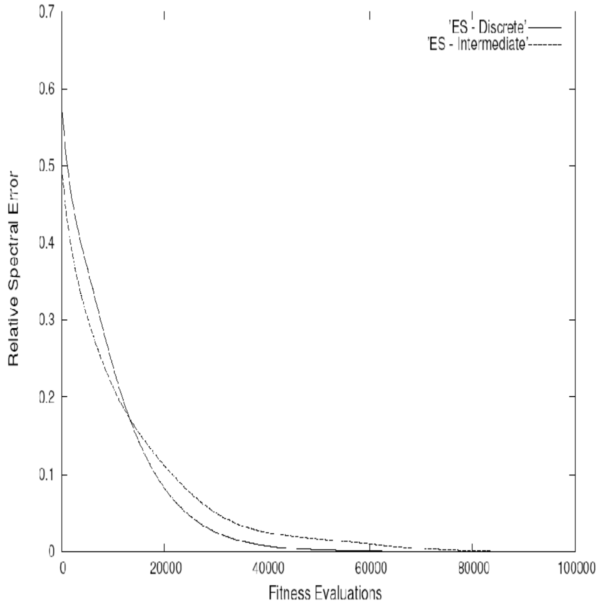

Fig. 2: ES

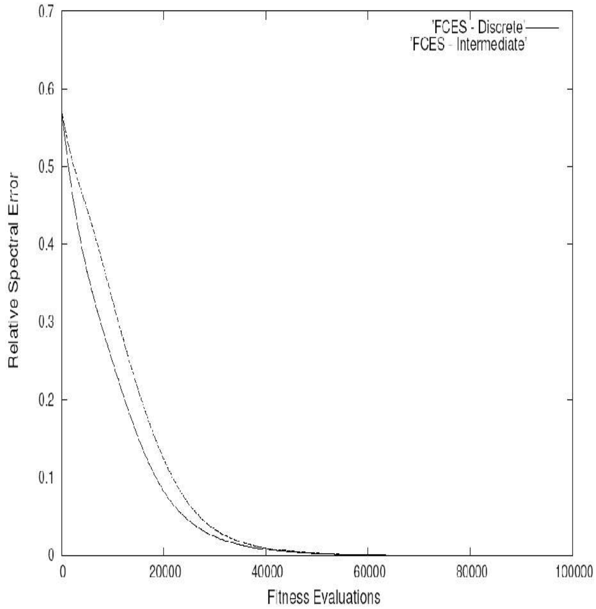

Fig. 3: FCES

The algorithms all use an Evolution Strategy (ES). Performance is tested for ES and FCES, with both discrete and intermediate recombination. However, discrete recombination is not meaningful in the case of CCES, as only a single parameter is represented in each subpopulation. All other operators and parameter values remain constant across algorithms and experiments, unless specifically stated otherwise. There is only space below for a brief description of the ES, readers are referred to [17] for a fuller treatment.

3.1 Evolution Strategy (ES)

The Evolution Strategy was originally developed in the 1960's by two students of the Technical University of Berlin [21] [22]. Presented as an automatic engineering design optimiser, shown to outperform traditional gradient oriented techniques, evolution strategies have since undergone numerous modifications and enhancements Schwefel [22]. The only ES parameter that is varied in the experiments that follow is recombination, which normally takes one of two main classes [1]:

- Intermediate - the genotype/phenotype vector of each offspring is obtained by taking the mean vector of its parents' vectors.
- Discrete - dynamic n-point crossover: each component of the genome of the offspring is produced by choosing a single vector component from the parents.

In all cases below, the ES uses a derandomised mutation operator described in [18], which is self-adaptive.

3.2 Fuzzy Clustering ES (FCES)

FCES combines the powerful local search properties of the Evolution Strategy with the strengths of Fuzzy Clustering, by partitioning the search population into fuzzy clusters that locally recombine and progress. With a sufficient number of clusters, and an adequate population size, all of the locally optimal peaks can be identified and thus, a global optimum is consistently found. Clustering, as a tool for global optimisation [23], was previously utilised to provide multiple start points for a local hill-climber optimisation. FCES follows essentially the same framework but uses a stochastic population-based search (the ES) in place of the local optimisation algorithm and proceeds by alternate application of optimisation and clustering. The aim is to achieve the reliability of clustering methods with the efficient self-adaptive search behaviour of the ES approach. The basis of the approach is that a clustering algorithm is used to form a partition of the parent population in a regular ES. The algorithm, therefore, is consistent with the standard generational model of an Evolutionary

Algorithm with global selection. Subsequent recombination blends genetic material from all parents in proportion to their degree of membership of a particular cluster (fuzzy clustering). This allows clusters, within the population, to form independently at regions of high fitness within the object landscape, preventing premature global convergence at locally optimal peaks. For a fuller description of this algorithm, and its application to a dynamic sound matching domain, see [17]. It is worthy of note that a population member constitutes a complete solution in the ES and FCES algorithms, whereas in the Cooperative Co-Evolutionary approaches, a single population member normally represents only a component of a complete solution.

3.3 Cooperative Co-ES (CCES)

When applying co-evolution to problems such as these, a standard approach is to identify a natural decomposition of the problem into subcomponents. Each component is assigned to a subpopulation, such that individuals in a given subpopulation represent potential components to the greater problem. Subsequently, each component is evolved simultaneously, but in isolation from each other. In order to evaluate the fitness of an individual from a given subpopulation, collaborators are selected from the other subpopulations in order to form a complete solution. The co-evolutionary approach adopted here arises directly from that of Potter et al [20], in developing their 'Cooperative Co-Evolutionary Algorithms' (CCGAs). However, in [20] a Genetic Algorithm is used as the evolutionary component in the architecture.

Potter et al describe two versions of CCGA. In both versions, a separate sub-population is instantiated for each functional variable in the system to be optimised. To gain an initial fitness value, each sub-population member is combined with a randomly selected individual from each of the other species. The resulting set of values is applied to the target function for evaluation. Subsequently, for CCGA-1, the fitness of any given subpopulation member is found by combining it with the current best subcomponents from the other subpopulations, that are temporarily frozen. Potter et al go on to describe an experimentally verified weakness in this credit assignment procedure, which appeared only to work well when applied to problems in which the members of the subpopulations were quite independent of each other. In fact, for some of the problems they used that displayed high inter-subpopulation dependencies, CCGA-1 performed worse than a standard GA. CCGA-2, therefore, was equipped with an enhanced credit assignment procedure. In CCGA-2, each subpopulation member is evaluated using the procedure defined for CCGA-1 *and* with a randomly selected individual from each of the other subpopulations. The fitness of the better performing combined vector of values is then returned as the individual's fitness. The result was an algorithm that performed better on those problems that CCGA-1 was not good at, at the expense of slightly lower performance on problems with independent subpopulations. They were able to present experimental results on a small selection of problems in which their approach outperformed a standard GA, in terms of both convergence speed and quality of best solutions. In the discussion section of their paper, they suggest that any evolutionary algorithm could be used in place of the GA and, indeed, we have done just this with an ES. The architectures that we call CCES-1 and CCES-2 are thus formed, having the matching credit assignment approaches as those described above for CCGA-1 and CCGA-2.

4. EXPERIMENTAL SETUP

4.1 Sound Generation

For the matching procedure to commence, the algorithm requires a target. It is possible to insert any sound into the model at this point; however, for testing purposes, it is useful to follow the methodology presented by Justice [10] and Payne [19]: matching *contrived* target sounds produced by a FM model identical in structure to the matching synthesiser. In such circumstances, a successful match will yield parameters equal to those with which the target tone was produced and, with repeated tests with a variety of targets, demonstrates that any point within the sound space is accessible via the matching process.

The FM circuit under test is constructed from two sinusoidal oscillators. Each oscillator has two input parameters. A 'Frequency' parameter controls the oscillator frequency expressed as a multiple of the synthesiser fundamental; for these experiments this fundamental is 200Hz. These are named $_c$, for the carrier oscillator and $_m$ for the modulating oscillator in equation 1. The second parameter relates to amplitude. In the case of the carrier oscillator, this parameter, A in equation 1, simply controls synthesiser amplitude; whereas for the modulating oscillator, this parameter controls the modulation index, I, described earlier. 40 random FM timbres were created by randomly generating these 4 parameters within a range of 0.0 to 8.0. For the carrier oscillator amplitude, this range is scaled to the full-scale deflection at the synthesiser output.

4.2 Algorithm Structure Details & Parameters

The underlying ES common to all of the algorithms tested was of the following form. Where intermediate crossover was used, = , i.e., all parents are used equally in recombination as described in [1]. The ES and FCES ran for 50 generations with the exogenous parameters (200,2000). In the *co-evolutionary* field, some alternative terminology is used, which is briefly described below. The iteration of one sub-population is called a generation. Of more interest is the situation when each sub-population has advanced by one generation, which is called a 'round' according to [8]. To retain parity across our experiments, CCES-1 ran for 50 rounds and CCES-2 ran for 25 rounds, because CCES-2 performs two fitness evaluations per offspring. Although the ES and FCES were operationally similar enough to use the same number of generations and exogenous parameters, for CCES they needed to be scaled by the number of subpopulations. Each subpopulation, of which there are four (one per parameter), therefore had the exogenous parameters (50,500), i.e., ¼ of the values used for ES and FCES.

4.3 Fitness Evaluation

Clearly, each of the algorithms tested required a means by which good and bad solutions can be differentiated. A metric is required to provide the 'distance' between (synthesis of) the potential solution and the target sound. The objective function identifies strong offspring, facilitating their selection as parents from which subsequent offspring can be produced. Within this work, the 'distance' is measured by calculating the normalised error, referred to as the 'relative error' in [7], which is measured between the target and candidate spectra. This error measure has proved effective in previous studies [5] [6] [7] and offers an excellent balance between detail and execution speed. The relative spectral error is given by the equation;

$$relative\ error = \sqrt{\frac{\sum_{b=0}^{N_{bin}} (T_b - S_b)^2}{\sum_{b=0}^{N_{bin}} T_{b^2}}}$$

Where T is a vector of the target spectrum amplitude coefficients, S a vector of synthesised candidate spectrum amplitude coefficients and N_{bin} the number of frequency bins produced by spectrum analysis. Each algorithm under test attempts to find a match on each of the waveforms using the relative spectral error as an objective function. In earlier experiments, we found that, in this application, an ES-based search could become trapped at locally-optimal points of the fitness landscape, a problem that any optimisation engine (including EC) would encounter for this application domain. To overcome this problem, the spectrum of the target and synthesised tones are modified to produce *windowed* spectra. *Windowing* allows spectrum error to be measured across a band, which has a smoothing effect on the object landscape. The Windowing function is more fully described in our paper [16]. The same window function and parameters are used for experiments described here, as that described in [16]. A complete cycle of the objective function is as follows:

1. Insert candidate solution into the FM model,
2. Subject the corresponding synthesised waveform to spectral analysis,
3. Calculate relative spectral error between target and synthesised candidate spectra.
4. Return this value as a fitness rating.

Clearly a perfect match yields the original parameters used to generate the sounds, and an error of 0. It is possible that non optimal matches, in these terms, could be good perceptual matches, however, this is an analytical survey; consequently, such perceptual matches are not considered.

5. RESULTS

For each algorithm type, ES, FCES and CCES, a graph is provided to illustrate performance, figs. 2 - 4. Each graph shows an algorithm's convergence rate, for successful runs. Each line on a graph shows one of the two variants of an algorithm tested, and is the average fitness of the best individual selected from the 5 successful runs over the total of 40 runs for each. Table 1 below, shows the relative performance of the algorithms and their variants that were tested here, in terms of the numbers of times that each algorithm converged to a globally optimal solution over 40 runs.

Table 1: Proportion of runs that converged to a near optimal solution

Algorithm	Converged Runs (out of 40)
ES - Intermediate Recombination	9
ES – Discrete Recombination	20
FCES Intermediate Recombination	25
FCES Discrete Recombination	38
CCES – 1	12
CCES – 2	19

6. DISCUSSION

It is clear from the graphical data, that all variants of all algorithms were able to find globally optimal solutions. It is also clear that the choice of intermediate or discrete recombination for the ES and FCES algorithms is not crucial to performance in this application in terms of convergence speed, but significantly affects convergence reliability. However, it is also clear that, although FCES does not give any clear advantage, in terms of convergence velocity compared with the ES, both variants of CCES do, with CCES-1 giving best performance in this respect. It is worth noting however, from the tabular data, that FCES most reliably finds the optimal solutions. CCES was the least computationally expensive of all the algorithms, only slightly so with respect to the ES, but 60% faster than FCES, principally because of time spent partitioning the population into clusters.

7. CONCLUSIONS

We have tested three Evolution Strategy (ES)-based optimisation algorithms for a simple FM synthesiser timbre matching problem. All three were capable of finding globally optimal solutions. The Fuzzy Clustering ES (FCES) was both computationally expensive and slow to converge. However, it produced globally optimal results with very high probability, compared with the other two algorithms. In contrast, whilst both the ES and the Cooperative Co-Evolution Strategy (CCES) are significantly less computationally expensive, CCES was significantly quicker to converge than either the ES or FCES.

REFERENCES

1. H. -G. Beyer, *Towards a theory of evolution strategies: On the benefits of sex -the (μ/μ,)-theory*, Evolutionary Computation, 3(1):81–111. 1995.
2. J. Chowning, *The Synthesis of Complex Audio Spectra by Means of Frequency Modulation*, Journal of the Audio Engineering Society, vol. 21, pp. 526-534. 1973.
3. J. Ethington, D. Punch, *SeaWave: A system for Musical Timbre Description*, J. Computer Music, 18 (1). 1994.
4. Fujinaga, and J. Vantomme. *Genetic algorithms as a method for granular synthesis regulation* Proceedings of the International Computer Music Conference, 1994.
5. R. Garcia, *Growing Sound Synthesizers using Evolutionary Methods*, European Conference in Artificial Life ECAL2001. Artificial Life Models for Musical Applications. 2001.
6. Horner, J. Beauchamp, and L. Haken, *Machine Tongues XVI: Genetic Algorithms and Their Application to FM*, Matching Synthesis, Computer Music Journal, 17(4), 17-29. 1993.
7. Horner, *Nested Modulator and Feedback FM Matching of Instrument Tones*, IEEE Transactions on Speech and Audio Processing, 6(6), 398-409. 1998.
8. T. Jansen, R. P. Wiegand, *Exploring the Explorative Advantage of the CC (1+1) EA*, Proceedings of Genetic and Evolutionary Computation Conference. Springer. 2003.
9. G. Johnson, *Exploring the sound-space of synthesis algorithms using interactive genetic algorithms*, Presented at Proceedings of the AISB Workshop on Artificial Intelligence and Musical Creativity, Edinburgh. 1999.
10. H. Justice, *Analytic Signal Processing in Music Computation*, IEEE Transactions on Acoustics, Speech, and Signal Processing,Volume ASSP-27, Number 6, 1979.
11. LeBrun, *A Derivation of the Spectrum of FM with a Complex Modulating Wave*, J. Comp. Music, 1 (4):51-52, 1977.
12. E. R. Miranda, *Striking the right note with ARTIST: an AI- based synthesiser*, M. Chemillier and F. Pachet (editors), *Recherches et applications en informatique musicale*, Paris: Editions Hermes, pp. 227- 239. 1998.
13. P. Mandelis, *Genophone: Evolving Sounds and Integral Performance Parameter Mappings*, Applications of Evolutionary Computing, EvoWorkshops2003, pp538- 549, 2003.
14. Manzolli, A. Maia, J. Fornari and F. Damiani, *The evolutionary sound synthesis method*, Proceedings of the ninth ACM international conference on Multimedia, Ottawa, Canada 2001.
15. E. R. Miranda, *On the Music of Emergent Behaviour: What can Evolutionary Computation Bring to the Musician?* Leonardo, Vol. 36, No. 1, pp. 55-58. 2003.
16. T. J. Mitchell and A. G. Pipe, *Convergence Synthesis of Dynamic Frequency Modulation Tones Using an Evolution Strategy*, EvoWorkshops2005: EvoMUSART, 3rd European Workshop on Evolutionary Music and Art, 2005.
17. T. J. Mitchell, J. C. W. Sullivan, *Frequency Modulation Tone Matching Using a Fuzzy Clustering Evolution Strategy*, Proc. Audio Engineering Society 118th Convention. 2005.
18. A. Ostermeier, A. Gawelczyk, N. Hansen, *A derandomized approach to self-adaptation of evolution strategies*, Evolutionary Computation, 2 (4):369-380. 1995.
19. R. Payne, *A microcomputer based analysis/resynthesis scheme for processing sampled sounds using FM*, Proc. Int. Computer Music Conf., San Francisco, CA, 1987, pp. 282 289. 1987.
20. M. A. Potter, K. A. De Jong, *A Cooperative Coevolutionary Approach to Function optimization*, Parallel Problem Solving from Nature -- PPSN II. 1994.

21. Rechenberg,*Cybernetic solution path of an experimental problem*, Tech. report RAE Translation 1122. Farnborough, Hants. 1965.
22. H. P. Schwefel, *Collective phenomena in evolutionary systems*. Pages 1025-1032 of: Preprints of the 31st Annual Meeting of the International Society for General Systems Research, 1987.
23. A. Torn, and A. Zilinskas, *Global Optimization*. Lec. Notes Comp. Sci., no. 350. Berlin: Springer-Verlag. 1989.