An Exploration of Evolutionary Computation Applied to Frequency Modulation Audio Synthesis Parameter Optimisation

Thomas James Mitchell

A thesis submitted in partial fulfilment of the requirements of the University of the West of England, Bristol for the degree of Doctor of Philosophy

Bristol Institute of Technology, Faculty of Environment and Technology,
University of the West of England, Bristol

May 2010

Acknowledgments

With so many people to thank who have indirectly contributed to this work I would first like to acknowledge those whose names are not mentioned here: family, friends – if you have ever said “you’re *still* doing that?” then you know who you are – I am forever indebted.

I would principally like to thank my director of studies Professor Tony Pipe who has been an endless source of inspiration and confidence from start to finish. Additionally, I would like to thank my other supervisors: Dr David Creasey, Dr Marcus Lynch and Stephen Allen whose contributions have been invaluable, especially in the drafting of this thesis. I would also like to thank Professor Alan Winfield for providing a forceful boot onto the path and also for providing funding in desperate times. I would also like to thank the Engineering and Physical Sciences Research Council for providing me with the opportunity to learn about all this stuff.

Finally, I would also like to express my gratitude to Dr Johnson Abraham, Dr Azarhar Machwe and Bhuvan Sharma for their friendship, company, humour and lengthy discussion on this mission; Professor Ian Parmee for the space; Professor Eduardo Miranda, Professor David Corne and Professor Larry Bull for reading and examining this thesis; Dr Charlie Sullivan for his help and ideas; Mike Bollen and Georgina Caldwell for their time proof reading “the tale of the derandomised mutation operator”; Rob Brown for sharing the pain; and to all the inspirational people who have demonstrated what drive, focus and dynamic thinking can achieve.

List of Publications

Some of the work in this thesis has also appeared in the following publications:

Mitchell, T. J., and Pipe, A. G. (2005) Convergence Synthesis of Dynamic Frequency Modulation Tones Using an Evolution Strategy. *Applications on Evolutionary Computing: EvoWorkshops*. 533-538.

Mitchell, T. J. and Sullivan, C. W. (2005) Frequency Modulation Tone Matching Using a Fuzzy Clustering Evolution Strategy. *In: Proceedings of the 118th Convention of the Audio Engineering Society*. Barcelona.

Mitchell, T. and Pipe, A. (2006) A Comparison of Evolution Strategy-based Methods for Frequency Modulated Musical Tone Timbre Matching. In: Proceedings of the 7th *International Conference in Adaptive Computing in Design and Manufacture*. Bristol. pp. 167-175.

Mitchel, T.J. and Creasey, D.P. (2007) Evolutionary Sound Matching: A Test Methodology and Comparative Study. *Proceedings of The Sixth International Conference on Machine Learning and Applications* (ICMLA'07).

Abstract

With the ever-increasing complexity of sound synthesisers, there is a growing demand for automated parameter estimation and sound space navigation techniques. This thesis explores the potential for evolutionary computation to automatically map known sound qualities onto the parameters of frequency modulation synthesis. Within this exploration are original contributions in the domain of synthesis parameter estimation and, within the developed system, evolutionary computation, in the form of the evolutionary algorithms that drive the underlying optimisation process. Based upon the requirement for the parameter estimation system to deliver multiple search space solutions, existing evolutionary algorithmic architectures are augmented to enable niching, while maintaining the strengths of the original algorithms. Two novel evolutionary algorithms are proposed in which cluster analysis is used to identify and maintain species within the evolving populations. A conventional evolution strategy and cooperative coevolution strategy are defined, with cluster-orientated operators that enable the simultaneous optimisation of multiple search space solutions at distinct optima. A test methodology is developed that enables components of the synthesis matching problem to be identified and isolated, enabling the performance of different optimisation techniques to be compared quantitatively. A system is consequently developed that evolves sound matches using conventional frequency modulation synthesis models, and the effectiveness of different evolutionary algorithms is assessed and compared in application to both static and time-varying sound matching problems. Performance of the system is then evaluated by interview with expert listeners. The thesis is closed with a reflection on the algorithms and systems which have been developed, discussing possibilities for the future of automated synthesis parameter estimation techniques, and how they might be employed.

Contents

List of figures

List of Tables

Chapter 1

Introduction

There is no doubt that modern technology has had a profound effect on the structure, form and performance of music. Powerful and inexpensive general-purpose computers have made electronic musical apparatus widely available to amateur and professional composers alike. The audio synthesiser has, and continues to play an important role in the development of modern music, enabling composers to electronically recreate the sound of acoustic instruments, or to explore beyond the realms of the familiar, to create sounds previously unheard. There are a wide variety of synthesis techniques which can be used to create musical sounds across a considerable range of timbres (tonal characteristics). Effective control and navigation of a synthesiser's sound space requires expert knowledge of the underlying synthesis technique, which may draw from theoretical and/or experiential knowledge. It is often the case that composers are required to defer traditional notions of musicianship to concentrate on the task of synthesiser programming: manipulating parameters to produce a desired effect. Consequently, an emerging archetype amongst contemporary electronic musicians is the composer/programmer: an individual versed not only in music, but also the inner workings of the enabling technology.

To experimental musicians lacking this technical prerequisite, the synthesiser interface can present an obstacle between artistic ideas and their expression. The parameters which are used to shape the sound character are specific to the particular synthesis architecture being employed, and rarely relate to sound in human terms. Consequently, there is a complex mapping between the dimensions of a synthesis parameter (or control) space, and the perceived sound character (or timbre) space. This can often result in an unintuitive synthesiser interface which is concerned with scientific process rather than artistic creativity.

Manufacturers attempt to sidestep this issue by providing a database of parameter settings that enable users to select from a wide range of pre-programmed sounds, known as presets. However, presets only provide access to a limited subspace of the complete synthesis sound space. If it were possible to relate the parameters of a synthesiser more directly to the user's intuitive understating of timbre, synthesiser control could become more transparently about sound creation rather than computer programming. The first step to achieving this is the development of a process which is able to map known sound qualities onto sound synthesis parameters. This requires a technique that can efficiently search a synthesis parameter space to identify configurations which achieve specific auricular characteristics. This thesis examines the use of evolutionary computation to do just this, and documents a series of experiments in which evolutionary algorithms are applied to the problem domain of sound matching with frequency modulation (FM) synthesis.

1.1 Context

One of the earliest examples in which evolutionary computation was applied to sound synthesis parameter estimation is the work of Professor Andrew Horner (1993) at the University of Illinois. In this work, Horner applied a genetic algorithm to both frequency modulation and wavetable synthesis in order to match (reproduce) acoustic musical instrument tones. As in more recent efforts (Riionheimo and Välimäki, 2003), a great deal of domain-specific knowledge was applied to reduce the complexity of the problem, by augmenting the synthesis process, and carefully encoding the parameters to make the search domain more tractable. The work presented here avoids the use of domain-specific knowledge and places emphasis on the application of existing and new evolutionary computation methods to address the parameter estimation problem while leaving the frequency modulation synthesis structures and their parameter encodings intact.

1.1.1 Evolutionary Optimisation

Artificial models of evolution have been shown to offer many advantages over more traditional optimisation techniques. For example, as evolutionary search is guided by a means of directed stochastic search, high-performance solutions are located more directly than purely random methods (Monte-Carlo search), and more efficiently than enumeration-based methods (brute force search). Maintenance of an advancing population ensures that evolutionary models are less susceptible to becoming trapped within local optima than calculus-based methods (Hill-Climbing), without the need for detailed *a priori* domain-specific knowledge.

Despite these strengths, evolutionary optimisation is not without weaknesses; in certain applications problems can arise. This thesis is concerned with a class of problem in which multiple distinct high fitness optima may be found within the problem space: the so-called *multimodal problem*. The primary reason standard evolutionary algorithms struggle within these environments is inherent in their fundamental architecture. The model combines stochastic search operators, to explore the problem space, with selective operators, to exploit profitable regions. This mechanism results in a tendency for the algorithm to focus on a single peak, which may be disadvantageous when the application domain is comprised of multiple high-fitness peaks. For the parameter estimation problem explored in this thesis, it is desirable to locate a diversity of solutions and not just one. Optimisation of multiple search space solutions enables a selection of sound matches to be optimised from which the synthesiser user is able to choose. This multiple solution proficiency has relevance to other application domains in which practitioners may require a variety of design solutions to facilitate better understanding of the underlying problem structure.

In recent times, a variety of evolution-based optimisation techniques have been designed specifically for optimising solutions to multimodal problems. This thesis describes the application of advanced developments in evolutionary computation to the problem of sound synthesis parameter estimation. In doing so, new extensions to the evolutionary algorithm model are presented in which cluster analysis is incorporated within the standard generational model of the evolution strategy. Genetic operators are also developed that enable ‘species’ to emerge within the evolving population. These extensions could be useful in wider fields of application; in particular, it is proposed that these algorithms provide a robust technique for real-valued multimodal function optimisation.

1.1.2 Frequency Modulation Audio Synthesis

FM audio synthesis, presented originally by Chowning (1973), provides a computationally efficient means of creating complex sound timbres, which has seen wide application in commercial systems. In what is termed *simple* FM, the instantaneous frequency of one sinusoidal oscillator is modulated by another. A diagram of the simple FM model is provided in figure 1.1. f_m and f_c are known as the modulator and carrier frequencies respectively, I is the modulation index, and A controls the output amplitude. The amplitude function for simple FM is given by the formula:

$$e = A\,sin\big(\omega_c t + I \sin\,(\omega_m t)\big)\,, \quad \omega_c = 2\pi f_c, \quad \omega_m = 2\pi f_m, \quad I = \frac{d}{f_m} \qquad (1.1)$$

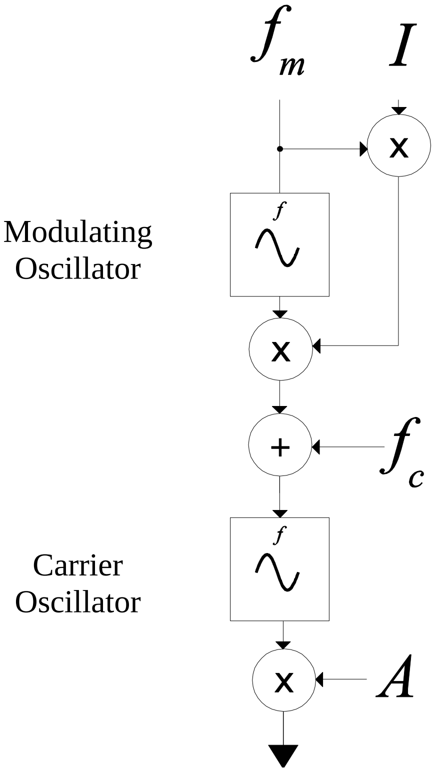


Figure 1.1: Simple FM model

In equation 1.1, e is the modulated carrier output, A is the peak output amplitude, ω_c and ω_m are the carrier and modulator angular frequencies respectively. The modulating oscillator varies the carrier frequency in the range specified by the peak frequency deviation d, which is the product of the modulation index I and the modulating frequency f_m. When I is assigned a value of zero there is no modulation of the carrier oscillator frequency, and the generated signal equates to a sine wave at frequency f_c. However, when $I > 0$, frequency partials are generated around the carrier at integer multiples of the modulating frequency as side-bands.

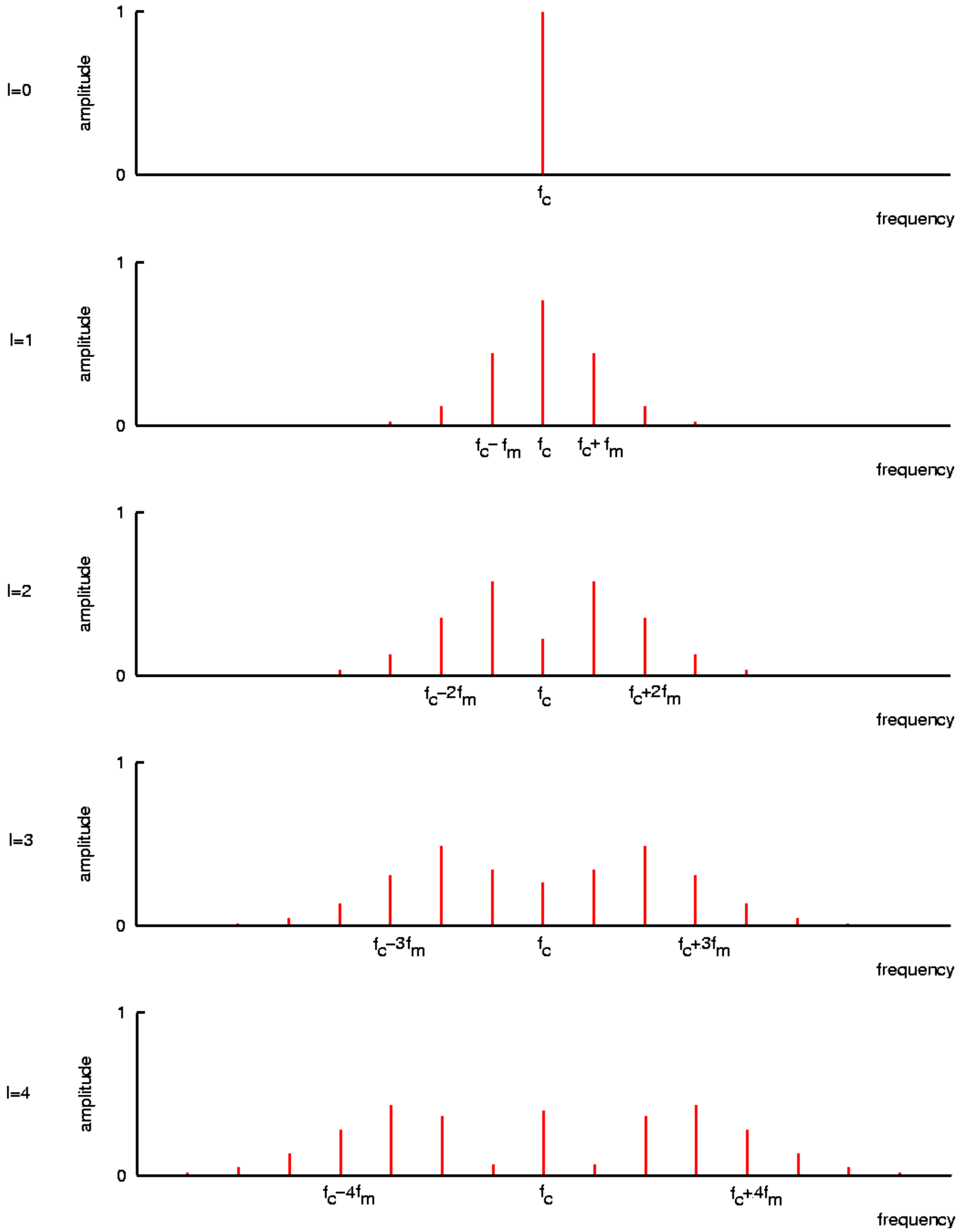


Figure 1.2: FM spectrum plots with increasing modulation index, adapted from Chowning (1973)

As illustrated in figure 1.2, the bandwidth of the modulated signal varies in proportion to the modulation index and modulator frequency. Notice, however, that there is a complex relationship between partial amplitudes and the modulation index I (the envelope of the spectrum is shaped by a non-linear function). The amplitudes of the partials are governed by Bessel functions of the first kind and order n, denoted $J_n(I)$, where the argument to the Bessel function is the modulation index. The Bessel functions are described by the following equation:

$$J_n(I) = \frac{I^n}{2^n n!}\left(1 + \sum_{k=1}^{\infty} \frac{(-1)^k I^{2k}}{\prod_{i=1}^{k} 2i(2n+2i)}\right) \tag{1.2}$$

The Bessel functions for $n \in [0,3]$ are shown in figure 1.3.

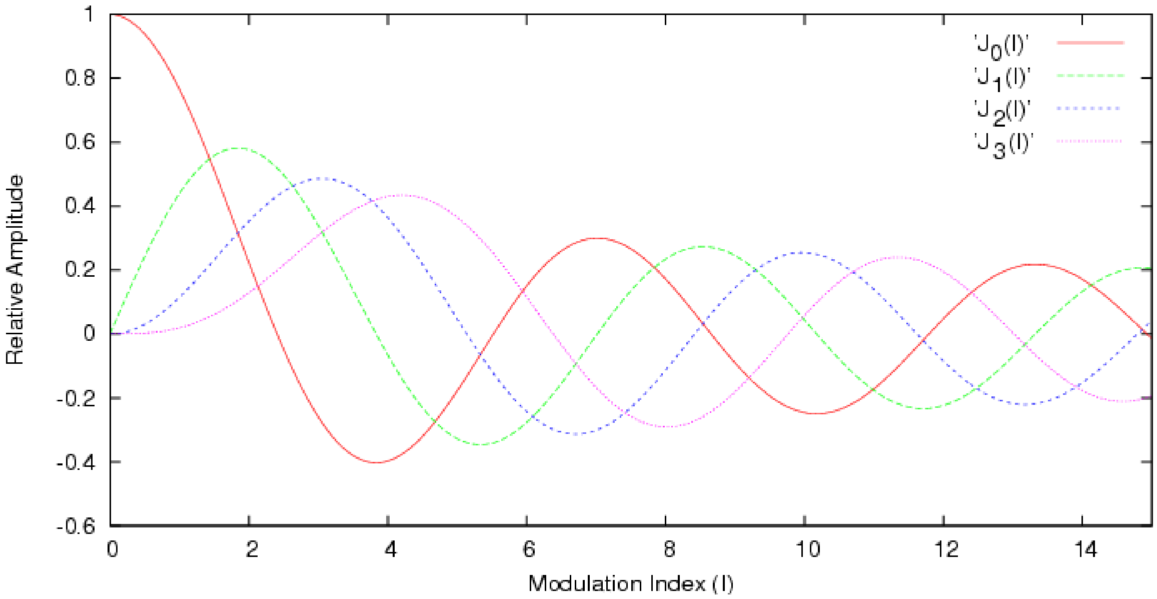


Figure 1.3: Bessel functions of the first kind and order n

The FM signal spectrum is shaped by the functions illustrated above. The amplitude of the partial at frequency f_c is scaled according to the value of $J_0(I)$, or 0^{th} order function; the amplitude of the first pair of side frequency partials are scaled according to the 1^{st} order function; the second pair of side frequency partials, by the 2^{nd} order function; and so forth. The trigonometric expansion of the simple FM function is given by the expression:

$$\begin{aligned} e = A\,\{ \quad & J_0(I)\ sin(\omega_c t) && + \\ & J_1(I)\ (sin(\omega_c + \omega_m)t - sin(\omega_c - \omega_m)t) && + \\ & J_2(I)\ (sin(\omega_c + 2\omega_m)t - sin(\omega_c - 2\omega_m)t) && + \\ & J_3(I)\ (sin(\omega_c + 3\omega_m)t - sin(\omega_c - 3\omega_m)t) && + \quad \ldots \} \end{aligned} \tag{1.3}$$

The non-linear relationship between the synthesis parameters and the spectral form of the modulated signal can often complicate the process of sound design with FM. When parameters are altered by hand it can be difficult to find specific combinations of partials to produce a particular timbre. The sound design process is complicated further by the unintuitive effects of reflected side frequencies. That is, partials synthesised with negative frequencies ($< 0Hz$) are directly mapped onto their positive values with negative phase. Reflected partials may then interact with positive components emphasising, or suppressing, coinciding partials: constructive and destructive interference respectively. If the ratio of the carrier to modulating frequency is a rational number, these reflections produce an arithmetic series of sinusoidal partials with frequencies at integer multiples of a fundamental frequency: the so-called harmonic spectrum. Conversely, when the ratio is irrational, reflections are positioned between the positive frequency components to produce a non-harmonic spectrum. With so few parameters with which to control such a wide range of timbres, combined with the non-linear effects outlined above, FM has become widely regarded as a difficult synthesis type to control (Kronland-Martinet *et al*, 2001), (Horner, 2003), (Delprat, 1997), (Payne, 1987). Consequently, a fair proportion of the work in this thesis is concerned with the development of algorithms that are designed to evolve solutions to complex real-world multimodal static optimisation problems. Thus, the fundamental research question that motivates this research is as follows:

Can evolutionary algorithms be created and employed to locate multiple distinct matches of a given target sound, with conventional frequency modulation audio synthesis structures?

1.2 Objectives

To introduce the work set out in the following chapters, the principal objectives which have directed this research are enumerated below.

1. To explore the potential for evolutionary computation as a mechanism for parameter estimation with frequency modulation synthesis.

2. To assess and develop optimisation algorithms suitable for optimising multiple sets of frequency modulation synthesis parameters that approximate a given target sounds.

3. To develop a testing method that enables algorithmic performance to be measured quantitatively in application to sound matching problems.

1.3 Contributions

In satisfying the above objectives the following contributions to knowledge are included in this thesis:

- In chapter four, a niching evolutionary algorithm is presented which incorporates *k*-means clustering into the evolutionary cycle of a conventional evolution strategy to preserve population diversity and enable solutions at multiple distinct optima to be maintained. The algorithm is named the clustering evolution strategy (CES) (Mitchell and Creasey, 2007).
- In chapter five, the CES architecture is included into the architecture of the cooperative coevolutionary algorithm, realised as a clustering cooperative coevolution strategy (CCCES), to again enable multiple distinct optima to be maintained while preserving the convergence characteristics of the standard architecture.
- In chapter six, a windowed relative spectrum error measure is developed which addresses some of the difficulties associated with comparing sounds using conventional spectrum error measures (Mitchell and Pipe, 2005).
- In chapter seven, a contrived testing method is developed which enables the optimisation component of the matching system to be analysed in isolation without interference from the synthesiser limitations. This enables effectiveness of each optimisation technique to be quantified and compared (Mitchell and Creasey, 2007).
- Also in chapter seven, the application of the developed algorithms to six standard and unsimplified continuous frequency modulation synthesisers for matching both static and dynamic sounds (Mitchell and Sullivan, 2005), (Mitchell and Pipe, 2006) and (Mitchell and Creasey, 2007). The developed matching system is then subjected to perceptual testing in chapter eight.

1.4 Methodology

The results presented in this thesis are empirical in nature. As the ultimate application domain forms a real-world problem, theoretical examination of the proposed algorithms is of limited practical use. In keeping with empirical evolutionary methods, the developed algorithms are examined comparatively in application to a variety of benchmark test functions over a number of runs. Results are plotted and results are compared for equality of means by t-test for bivariate data and ANOVA for multivariate data. In application to the parameter estimation problem, algorithm performance is measured according to the matching method described in chapter seven.

1.5 Thesis Structure

The chapters of this thesis have been organised into sections which are largely self-contained. To aid clarity, the algorithmic and application components of the system have been separated, such that chapters two–five concern the development and testing of the evolutionary algorithmic contributions of this thesis in isolation, and chapters six and seven extend their application to the real-world frequency modulation sound matching problem. In reality, the development of the matching system involved interplay between these two components, with evolutionary algorithms developed and tested in application to benchmark test functions, based upon problems that were encountered in the application domain.

Chapters two and three provide a review of evolutionary computation, the major types of evolutionary algorithm and a variety of augmentations to these algorithms which are intended to enhance performance within rugged, multimodal search domains. The evolutionary algorithmic developments of this work are described in chapters four and five, while chapters six and seven provide further review of the frequency modulation sound matching problem and the performance of traditional and developed algorithms within this domain. Chapter eight describes a set of perceptual listening tests with a panel of expert listeners in which the perceived similarity of evolved matches are juxtaposed with their target sounds.

1.6 Implementation

Experimental results provided in this thesis have been produced by applications written by the author in C/C++ using GCC under the GNU/Linux operating system. A number of different synthesis configurations were examined initially using the graphical programming environment PD (Pure Data) prior to their implementation in C/C++. The evolutionary algorithms tested herein have been built according to the specifications described in the relevant literature. Statistical analysis is performed on all results with plots and comparisons generated by the SPSS statistical analysis software. The listening tests provided in chapter eight were performed using Max/MSP patches designed specifically for the task.

Chapter 2

Background: Evolutionary Computation

This chapter provides a brief introduction to the field of evolutionary computation, including a summary of the general evolutionary model, followed by a review of the major algorithms that embody this field of research.

2.1 An Introduction to Evolutionary Computation

Evolutionary computation (EC) is a broad research area within which the principles of Darwinian evolution (Darwin, 1859) are employed to offer insight and solutions to a wide range of problem domains. A major subset of this field is concerned with the application of evolutionary algorithms (EAs), adopting the principles of EC, to optimise the parameters of static objective functions. Three independent and, in some cases, contemporaneous interpretations of the evolutionary model have been developed for this purpose. The genetic algorithm (GA), the most widely known of all EAs, was developed originally by Holland (1975), and applied to parameter optimisation problems by De Jong (1975). However, much earlier than this, an adaptive system known as evolutionary programming (EP) was developed by Fogel *et al* (1966) and was reintroduced later as a more general purpose function optimiser by Fogel (1992). Simultaneously, Rechenberg (1965) was working independently on an adaptive optimiser known as the evolution strategy (ES).

Contemporary evolution-inspired function optimisers are descended from one of these three interpretations, which have, since their introduction, been applied to an ever-increasing number of engineering problems. For a diverse list of applications the reader is directed to Schwefel and Bäck (1997), Bäck *et al* (1997a) and Rothlauf *et al* (2005).

Although these three classes of EA are not without difference, each models the processes of evolution to some degree. At an abstract level, evolution can be regarded as the mechanism by which sophisticated and well adapted biological structures have come to exist: a process of natural selection which emerges when there is a superfluity of genetic material within an environment in which individuals struggle for existence.

> *Just as a breeder chooses those individuals closest to his desired optimum, and discards the rest, so the natural environment improves the performance of a species by eliminating the less effective. Individuals possessing particular adaptations will survive better, and by virtue of the heritable nature of these adaptations, they will transmit them to their offspring. Gradually, the adaptations will spread and improve so that the species will become better suited to the environment which it inhabits.*
>
> *Parkin (1979)*

2.2 The Evolutionary Algorithm

When the principles of evolution are simulated and used to optimise solutions to engineering problems, the individuals, referred to in the above quote, are represented by a population of potential solutions. The environment, which is defined by a given objective function, quantifies the relative worth or *fitness* of each solution. Adaptations are introduced by recombining and mutating individuals from one generation to produce the offspring that form the next. The elimination of less effective genetic material is facilitated by selecting those individuals with above average fitness to partake in reproduction more frequently than those with below average fitness. This selective bias introduces the notion of natural selection, enabling well adapted genes to propagate throughout subsequent generations. A simple representation of this evolutionary model is provided in figure 2.1.

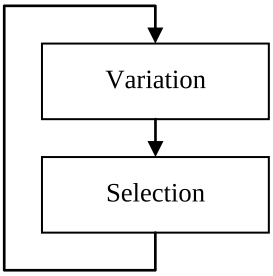


Figure 2.1: The evolutionary model

This figure illustrates the three core components of the evolutionary model: *variation*, *selection* and the *reproductive cycle*. Optimisation is achieved by maintaining a population of solutions, which are alternately subjected to variation and selection. Variation, as already stated, takes the form of *recombination* and *mutation*, and represents the birth of new individuals. Before individuals expire, they may be selected for variation based upon their performance within the test environment. The generational cycle repeats until some termination criterion is satisfied: either one or more adequate solutions are engendered, or a generational limit is reached.

To demonstrate how this metaphor of evolution might be applied to optimisation problems, an objective function of the following form is considered:

$$f: \mathbb{R}^n \rightarrow \mathbb{R} \tag{2.1}$$

In this example, the goal of the evolutionary optimiser is to find $\vec{x}$, a vector of parameters, where $\vec{x} \in \mathbb{R}^n$, such that the function f is minimised (or alternately, maximised):

$$f(\vec{x}) \rightarrow min \tag{2.2}$$

The first step in applying the evolutionary model is to decide how search points within the objective space are represented. Typically, when optimising parameters of a real-valued objective function, there are two alternatives:

- direct, real-valued representation.
- mapped, binary-coded representation.

When parameters are represented by real-valued numbers, population members directly represent solutions to the objective function. In other words, each individual contains a complete solution to the function f. The ES and EP algorithms both operate directly on the object parameters, within what is termed *phenotype* space. In contrast, traditional GAs employ a binary representation, in which object parameters are encoded into discrete, usually fixed length, bitstrings. These algorithms are said to work in *genotype* space, and require functions that map individuals between genotype and phenotype space.

Before the reproductive cycle may begin, it is necessary to initialise the system's population by generating n separate random numbers for each individual. Thereafter, genetic material from the parent population is blended via recombination to generate offspring, which are subsequently varied by means of mutation. Mutation is implemented with the random perturbation of offspring, to introduce chance positive stochastic variation. Each offspring is then evaluated as a solution to the objective function f and assigned a fitness quotient in proportion to its performance. New parents are selected based on their relative fitness, ensuring that high-performing individuals are then chosen to take part in the next round of variation more frequently than low-performing individuals.

A widely accepted viewpoint of the evolutionary process considers selection to encourage the exploitation of high-fitness regions of the solution space, while recombination and mutation facilitate the exploration of new regions which are not currently represented by population. This interplay of exploitation and exploration directs the evolving population towards higher levels of fitness, and thus, evolutionary computation has several advantages over more traditional optimisation methods. For example, enumerative and random-based optimisation techniques are only capable of exploration; consequently the process of optimisation is costly. Hill-climbing-based techniques only exploit and are therefore susceptible to becoming trapped within local optima. The implementation of both search tactics within EAs offers a heuristic optimisation method, which is both robust and efficient.

However, EAs do not provide a universal solution to all optimisation problems; there are certain problem characteristics for which evolutionary algorithms are not well suited. In his study of *epistasis*, Davidor (1991) identifies two environmental extremes for which EAs have no advantage over more traditional optimisation methods. At one extreme the problem is so well structured and easy to solve that an EA would be unable to perform better than a hill-climber. At the other extreme, the problem domain is so complex and unstructured that an EA would be unable to perform better than a random strategy. Davidor concludes that application domains with characteristics between these two extremes are the types of problem for which optimisation by EA might be advantageous.

2.3 Canonical Evolutionary Algorithms

Although the work presented throughout the later chapters of this thesis is largely built upon the theoretical framework of the ES (Beyer, 2001), it is sensible to first consider the general nature of parameter optimisation with EC. This section provides an introduction to GAs and ESs, concluding with a brief summary of their similarities and differences. Further review of EP is not included in this thesis as it is distinguished from the ES and GA, principally by the absence of recombination (Beyer, 2001, p3) (Bäck *et al*, 1993).

2.3.1 The Genetic Algorithm

The GA was originally developed by Holland to study and model the mechanisms of natural adaptive systems (Holland, 1975). Later, within his doctoral thesis, De Jong (1975) set out the framework for the application of Holland's adaptive model to the problem of parameter (function) optimisation. This application formed the precursor to a plethora of GA-based optimisers designed to improve performance when applied to new and specialised problem domains. This section will provide only a brief summary of the canonical GA; for a more comprehensive introduction with mathematical foundations, see Goldberg (1989) and Whitley (1994).

The canonical genetic algorithm models evolution at the *genotypic* level, adopting a Boolean representation for the object parameters $\vec{x}$. The choice of binary-coded representation is inspired by the way in which biological structures are encoded into the low cardinality alphabet of DNA. Within the GA architecture, individuals are constructed from a single bitstring (*chromosome*) $\vec{y} \in \{0\,,1\}^l$ which is divided into segments (*genes*) representing each object parameter. To facilitate the optimisation of the function f, GAs require functions that map between genotype and phenotype space: $g\,:\,\vec{y} \rightarrow \vec{x}$ and $g'\,:\,\vec{x} \rightarrow \vec{y}$ respectively. Often this mapping procedure is achieved by decoding each binary-represented gene from its integer value, which is then linearly scaled into the range of the corresponding object parameter.

The canonical genetic algorithm is represented by the pseudocode shown in figure 2.2.

```
t = 0;
initialise P(t);
evaluate P(t);
loop begin
      P'(t) = select(P(t));
      P(t+1) = recombine(P'(t));
      P(t+1) = mutate(P(t+1));
      evaluate(P(t+1));
      t = t + 1;
loop end;
```

Figure 2.2: Canonical GA pseudocode

Where P and P' denote the population and mating pool respectively at time t. Subsequent to initialisation (usually random), each individual of the population is decoded and applied to the objective function to retrieve a value of fitness. In the canonical GA, selection is facilitated probabilistically using the so-called *roulette wheel* selection mechanism. Each individual is represented by a sector of a notional wheel, sized in proportion to its fitness. A spin of the wheel yields a mating candidate, which is copied into a temporary mating pool (P') in preparation for variation by recombination and mutation.

The recombination operator is termed *crossover*, and provides the primary source of variation within a GA. The most basic GA recombination technique is known as single-point crossover, which operates by simply concatenating the first part of one parent string with the second part of another; where both the crossover point and the participating parent strings are chosen at random. Crossover is responsible for combining useful segments from the gene pool to form fitter solutions. This concept is otherwise known as the *building block hypothesis,* which states that short combinations of highly fit genes (building blocks) evolve simultaneously throughout the population (*implicit parallelism*). Well-adapted building blocks are assembled by recombination to create highly fit descendants (Goldberg, 1989). This also relates to Holland's *schema theory,* which states that an exponentially increasing number of trials are allocated to useful building blocks (or *schemata*) from one generation to the next (Holland, 1975).

In the theory that relates to the canonical GA, it is considered that the genes of the optimal individual are distributed throughout the population from the outset. Optimisation is then the process of correctly assembling those genes, which is achieved principally by the recombination operator. The mutation operator is widely considered to be the background source of variation (Goldberg, 1989, pp14), as it has the potential to destroy building blocks through random change. However, it is possible that a 0 or 1 positioned at a certain bit position (*locus*) may be absent or lost from the population which recombination would be unable to recover. To remedy this problem, mutation is applied by randomly inverting bit positions at a low probability, usually around 1% per bit (Schaffer *et al*, 1989), (Grefenstette, 1986).

Subsequent to fitness evaluation, descendants entirely replace their progenitors to embody the succeeding generation of individuals; this replacement approach is often referred to as *generational*. Individuals are then selected from the new population in preparation for crossover, and the reproductive cycle continues.

GA Performance and Augmentations

The earliest analysis of GA objective function optimisation was performed by De Jong (1975). De Jong compiled a suite of diverse test functions, and introduced two measures to quantify performance:

- an *on-line* measure, to indicate performance within real-world domains, where emphasis is placed on the rapid location of good results.
- an *off-line* performance measure for simulations in which many function evaluations may be performed, and the best solution saved for use at the end of a run.

The *on-line* performance is calculated from the mean average of all fitness evaluations, while the *off-line* performance is calculated from the mean average of the best solutions at each generation. De Jong also proposed numerous enhancements and modifications to the canonical GA to provide improved performance when applied to optimise a variety of different problem characteristics. These extensions included:

- an elitist strategy, in which the fittest solution at each generation is preserved and copied directly into the next.
- an expected value model, with a stable selection scheme to prevent loss of diversity throughout the early stages of evolution.
- a generalised crossover operator, to enable multi-point crossover between bitstrings.
- a crowding operator, to enhance performance in multimodal environments.

The crowding operator is of interest to this work as it presents a method for preserving diversity by encouraging the formation of *species*, a concept which will be explored further in chapter three. Diversity within the crowding model is maintained by adopting an overlapping (generational) strategy in which offspring replace their progenitors based not on fitness, but similarity in the genotype space.

In the next section of this thesis, the state-of-the-art ES is briefly examined, providing the general framework on which the algorithms presented throughout chapters four and five of this thesis are based.

2.3.2 Evolution Strategies

While EP was being developed in the U.S.A., two engineers at the Technical University of Berlin were independently developing their own evolution-inspired parameter optimisation technique known as the *evolutionsstrategie*. The earliest ES, developed by Rechenberg (1965), implemented a set of simple rules for the sequential design and analysis of real-world parametric engineering problems.

The ES models the processes of evolution at the phenotypic level. As such, search points are represented directly as *n*-dimensional vectors of (usually) real-valued object variables $\vec{x} \in \mathbb{R}^n$. As well as representing object variables, individuals (denoted I) also include a set of endogenous strategy parameters $\vec{\sigma} \in \mathbb{R}^n$, as well as a *fitness* value, equal to its objective function result $f(\vec{x}) \in \mathbb{R}$:

$$I = \left(\vec{x}, \vec{\sigma}, f(\vec{x})\right) \tag{2.4}$$

The original *two-membered* ES (the so-called $(1+1)$ ES) employs a simple mutation/selection mechanism, in which a single *parent* is mutated to produce a single *offspring*. If the mutation is found to be profitable the offspring replaces its parent, otherwise, the offspring is discarded. Later, *multi-membered* ESs were developed in which populations of parent and offspring individuals are maintained by the algorithm. The two most notable of these population-based ESs were introduced by Schwefel (1981) and constitute:

- the $(\mu + \lambda)$ strategy, in which μ parents are varied to produce λ offspring, and μ parents of the subsequent generation are selected from all $\mu + \lambda$ individuals.
- the $(\mu\,,\lambda)$ strategy, in which selection is made among only the λ offspring. Parents are systematically discarded regardless of their fitness value.

The pseudocode for the basic multi-membered ES is provided in figure 2.3.

```
t = 0;
initialise $P_\mu(t)$;
loop begin
        $P_\lambda(t)$ = recombine($P_\mu(t)$);
        $P_\lambda(t)$ = mutate($P_\lambda(t)$);
        evaluate($P_\lambda(t)$);
        P(t+1) = select($P_\lambda(t)$ ( + $P_\mu(t)$));
        t = t + 1;
loop end;
```

Figure 2.3: Multi-membered ES pseudocode

In this algorithm, P_μ and P_λ denote the parent and offspring populations respectively at time t. Following the random initialisation of μ parent individuals, the generational cycle begins. Genetic information from the parent population is blended via recombination, and then varied by mutation to engender λ offspring solutions. Thereafter, offspring are evaluated for fitness, and the top μ individuals are selected deterministically as parents from which the subsequent generation will breed.

2.3.2.1 Recombination

Recombination is the process by which the genetic information is blended to ensure that descendants inherit the characteristics of their ancestors. In the ES, recombination techniques are divisible into two major classes:

- *intermediate recombination*. In which offspring are generated with the mean average of their parents' parameters.
- *discrete/dominant recombination*. In which offspring parameters *(alleles)* are chosen at random from parent candidates[1].

Each class has *local* and *global* variants. In the former only two parents are *married* in bisexual recombination, whereas in the latter, all parents partake in multisexual recombination. Schwefel and Rudolph (1995) extended the ES to include the concept of *variable arity*, introducing the exogenous parameter ρ, to indicate the number of parents participating in the procreation of each descendant. With this generalisation, arity is controlled by the mixing number ρ which may be set to any value in the range $1 \leq \rho \leq \mu$. All variants of the ES may then be realised as special cases of the more general $(\mu/\rho \overset{+}{,} \lambda)$

[1] This technique is comparable to *uniform crossover* in the genetic algorithm (Syswerda, 1989),where each bit is chosen at random between the parental candidates.

strategy, with $\rho = 1$ indicating a strategy with no recombination and $\rho = 2$ and $\rho = \mu$ equating to local and global recombination respectively.

More formally, intermediate recombination amongst ρ parents, is given by:

$$x'_i = \frac{1}{\rho}\sum_{k=1}^{\rho} x_{k,i} \tag{2.4}$$

where $x_{k,i}$ represents the i^{th} object parameter $i \in \{1, 2, \ldots, n\}$ of the k^{th} parent $k \in \{1, 2, \ldots, \rho\}$, from which the recombinant object parameters $\vec{x}' \in \mathbb{R}^n$, positioned at the centre of mass or centroid of the contributing parents, may be derived. Discrete recombination, on the other hand, is given by:

$$x'_i = x_{k,i} \tag{2.5}$$

with k chosen randomly anew for all i.

Both recombination methods may be applied to the mutation step-sizes $\vec{\sigma}$ in addition to the object parameters $\vec{x}$.

Eiben and Bäck (1997) empirically investigated the performance of a multi-membered ES in application to a series of test functions, while varying the parameter ρ. The paper concludes that, in most cases, *multisexual* recombination of the object variables leads to an increase in performance over asexual recombination (no recombination), with optimal results often attained when $\rho = \mu$ (global recombination).

2.3.2.2 Genetic Repair

Beyer (2001) formally confirmed the positive effects of multisexual recombination with a theoretical analysis of the $(\mu/\mu, \lambda)$ ES applied to a simple unimodal sphere function. An ES adopting global intermediate recombination was shown to provide an increase in progress rates when compared with an equivalent (μ, λ) ES without recombination. Beyer attributed this increased rate to the corrective effects of the recombination operator, which gave rise to the *genetic repair hypothesis*:

> *The benefit of (intermediate) recombination lies in genetic repair (GR). The effect of recombination is the extraction of similarities.*
>
> *Beyer (2001, p222)*

When μ/μ intermediate recombination is applied to a population of parent individuals, recombinants are situated at the centroid of the parent population. Mutation then serves to displace offspring from this centroid position normally at random. Beyer demonstrated that mutants deviate from their origin by a mutation vector which may be decomposed into an x component, in the direction of the optimum, and an h component, perpendicular to the direction of the optimum. Deterministic selection of the fittest mutants yields parents endowed with correlated positive x components, with relatively uncorrelated h components. By interpreting the h component as the harmful effects of mutation (as it lies perpendicular to the beneficial x component), the subsequent application of recombination will tend to preserve the useful components of the parents (similarities), while cancelling, or *repairing*, their harmful components (differences). In other words, both the beneficial and harmful effects of mutation are averaged, but selection ensures that the beneficial effects are correlated, while the negative effects are not. Beyer reports that the h component of the calculated centroid is smaller than the mean expected length of a single mutation by a factor of $\sqrt{\mu^{-1}}$. Moreover, as the harmful component of mutation is reduced by the $(\mu/\mu\,,\lambda)$ intermediate strategy, the mutative strength may increase above that which is optimal for a $(\mu\,,\lambda)$ strategy, resulting in a larger improvement step and an overall increase in progress rate.

Interestingly, the genetic repair hypothesis also holds when parents are recombined using the discrete recombination operator. In contrast to the intermediate operator, discrete recombinants are not positioned at the centroid of the parent population. Instead, descendants are constructed from a vector element chosen randomly from the parent population. This procedure is equivalent to randomly sampling the parents' genetic material, which is ultimately distributed around a statistical centroid. Thus, recombination can be viewed as a larger (*surrogate*) mutation from an estimated centroid (Beyer, 1995 and Beyer, 2001).

Although useful in understanding how recombination and mutation play different yet complementary roles in directing a population towards an optimum, it is not possible to extend Beyer's theory to non-spherical optimisation problems, not least real-world problems with unknown characteristics. In rugged problem spaces comprising multiple peaks and flat plateaus, recombination and mutation are still beneficial to evolution; however, their benefits cannot be explained by the genetic repair hypothesis alone.

2.3.2.3 Mutation

In contrast to both the GA (in which recombination is widely regarded to be the primary variation operator) and EP (relying upon mutation alone), the ES takes an intermediate position: mutation and recombination are applied with equal importance (Beyer, 2001). However, the mutation operator does provide the primary source of variation, and thus exploration. Recombination works synergistically with mutation, reducing variation error and accelerating progress rates.

Object Parameter Mutation

Mutation is applied to the object parameters of each recombinant $\vec{x}$ with the addition of the mutation vector $\vec{m}$:

$$\vec{x}' = \vec{x} + \vec{m} \tag{2.6}$$

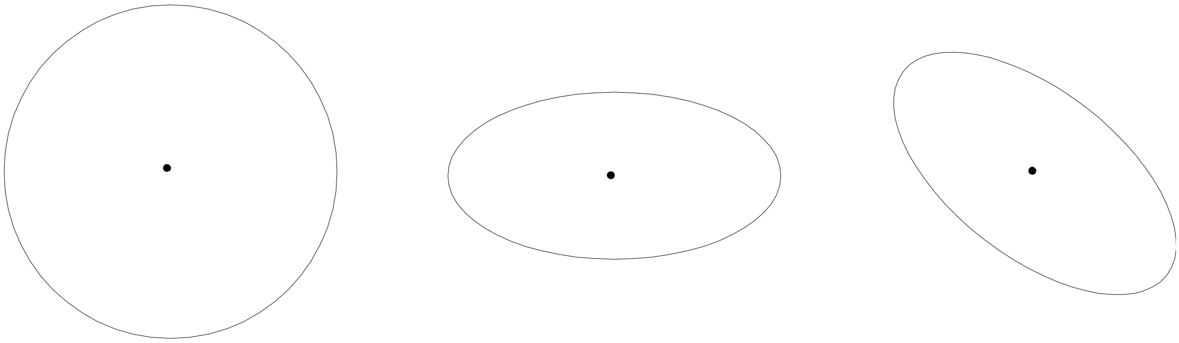
This delivers the mutated object parameters $\vec{x}'$. Each element of the mutation vector is drawn randomly from the standard normal distribution $\mathcal{N}(0\,,1)$ and scaled according to the mutation strength specified by the strategy parameters of the recombinant individual. This mutation scheme ensures that mutative jumps through the search space are:

- *ordinal,* favouring small jumps through the search space over large jumps.
- *scalable,* according to the mutation strength σ, such that any point within the space may be reached.
- *unbiased,* ensuring that, on average, mutants deviate from their point of origin isotropically.

The lack of bias in the mutation operator ensures that that there is no deterministic drift without selection.

Variations

In its most rudimentary form, the mutation normal distribution is isotropic, i.e., only one step-size parameter σ is required for the mutation of all object parameters $\vec{x}$. With an isotropic mutation scheme, the surface of mutation probability isodensity forms a circle, sphere, or hypersphere, dependent upon n, the problem dimensionality. This is depicted in figure 2.4a (with $n = 2$).



(a) (b) (c)

Figure 2.4: Two-dimensional probability isolines of (a) isotropic, (b) ellipsoidal and (c) rotated ellipsoidal mutation

With an isotropic mutation mechanism the mutation vector is given by:

$$\vec{m} = \sigma \cdot [\ \mathcal{N}_1(0\,,1)\ ,\ \mathcal{N}_2(0\,,1)\ , \dots , \mathcal{N}_n(0\,,1)\ ] \qquad (2.7)$$

As such, each individual within the system contains only one strategy parameter σ, which offers global control of the mutation step-size for each object parameter.

In many applications it is beneficial to employ an individual step-size parameter for each object vector element, enabling the mutation density function to form an axis parallel ellipse, ellipsoid, or hyper ellipsoid dependent upon n (see figure 2.4b). This extension to the mutation operator requires each individual to contain a vector of endogenous step-size parameters $\vec{\sigma}$ of length n. The corresponding mutation vector is then given by:

$$\vec{m} = \ [\ \sigma_1 \cdot \mathcal{N}_1(0\,,1)\ , \sigma_2 \cdot \mathcal{N}_2(0\,,1)\ , \dots , \sigma_n \cdot \mathcal{N}_n(0\,,1)\ ] \qquad (2.8)$$

The most elaborate and general mutation scheme was proposed by Schwefel (1981) and incorporates the concept of correlated mutation angles, in which a rotation matrix enables the density (hyper-)ellipse to adaptively align itself to the topology of the objective function (see figure 2.4c). The corresponding mutation vector is given by:

$$\vec{m} = R \cdot [\ \sigma_1 \cdot \mathcal{N}_1(0\,,1)\ , \sigma_2 \cdot \mathcal{N}_2(0\,,1)\ , \dots , \sigma_n \cdot \mathcal{N}_n(0\,,1)\ ] \qquad (2.9)$$

The rotation matrix R consists of $n^2 - n/2$ rotation angles $\vec{\psi}$, which are included within each individual (I) and adapted with the step-size parameters $\vec{\sigma}$. For further reading and implementation details of this generalised self-adaptation mechanism, see (Schwefel, 1981).

When the endogenous strategy parameters ($\vec{\psi}$, and $\vec{\sigma}$) are adapted along with the object-variables, optimisation takes place simultaneously in both the object and strategy parameter spaces. This process ensures that high performing solutions are selected for reproduction along with their corresponding strategy parameters, which may go on to yield even stronger solutions throughout subsequent generations.

Strategy Parameter Adaptation

By selecting optimal values for the strategy parameters controlling the mutation strength, the maximum rate of progress can be maintained. The problem then arises as to how the strategy parameters may be continuously adapted throughout the course of evolution. For the ES there are two standard approaches for step-size adaptation: the $1/5^{th}$ rule and *self-adaptation*.

The $1/5^{th}$*Rule*

By studying the dynamics of the $(1+1)$ ES when applied to two differing objective functions, Rechenberg observed that the maximum rate of progress corresponds to a particular value for the probability of a successful mutation (Rechenberg, 1973, as cited in Beyer and Schwefel, 2002). As the mutation step-size tends to zero, the probability of success becomes very high; conversely, as the step-size tends to infinity, the probability of success becomes very low. In order to maintain an optimal rate of progress, the step-size parameter σ should be adjusted to maintain a probability of success within these two extremes; a range that has become known as the *evolution window*. This observation led to the derivation of a general rule for the probability of success: mutation step-size adaptation by the $1/5^{th}$ rule. Successful mutations are measured over several generations (often equal to the dimensionality of the problem) and if the probability of a successful mutation is found to be below $1/5$, the mutation step-size is decreased. A recommended factor for the multiplicative/multiplicative inverse adaptation of the step-size parameter by the $1/5^{th}$ rule is 0.85 (Schwefel, 1995).

However, there are certain limitations that apply when adapting the mutation step-size using the $1/5^{th}$ rule:

- the rule may only be applied when all object parameters are controlled by a global mutation step-size parameter (isotropic mutation).
- the rule is only accurate for the two-membered $(1+1)$ ES.
- the rule is only accurate for certain landscape characteristics.

Schwefel (1987) subsequently introduced a more flexible adaptation scheme termed *self-adaptative* mutation.

Self-Adaptation

In the self-adaptive mutation scheme, evolutionary search takes place simultaneously in both the object and strategy space. It is assumed implicitly that optimal step-sizes result in fitter descendants and thus will be selected more frequently than non-optimal step-sizes. This adaptation scheme has now become the standard *modus operandi* for the state-of-the-art ES.

In the self-adaptive method, a vector of step-size parameters $\vec{\sigma}$ is included within each population individual, with the object parameters $\vec{x}$. Each element of the step-size vector specifies a unique mutation strength for each object parameter, thus facilitating the axis parallel ellipsoidal mutation scheme illustrated in figure 2.4b. To maintain optimal rates of progress, the mutation step-sizes must themselves be adapted along with the object parameters, by means of recombination and mutation.

Step-Size Recombination

Recombination of the step lengths is considered to be essential for the effective operation of the self-adaptive mechanism (Bäck and Schwefel, 1993). The intermediate and discrete recombination operators, identified above for the variation of object parameters, may be directly applied to vary the step-size parameters.

The progress of the ES is often restricted by large fluctuations in the strategy parameters that occur throughout the course of evolution. This *overadaptation* effect is particularly prominent in cases where small values for μ are assumed in conjunction with discrete recombination. For this reason, intermediate recombination of the strategy parameters is highly recommended as the effects of genetic repair attenuate these fluctuations (Beyer, 2001).

Step-Size Mutation

To ensure that step-sizes remain positive, the individual step lengths of the vector $\vec{\sigma}$ are mutated by a multiplicative, rather than additive process (as is case for mutation of the object parameters). The principles derived for the mutation of object-variables also apply for the mutation of the strategy parameters. For example, mutations to the object parameters should be ordinal, scalable and unbiased. However, as mutations are applied multiplicatively they should be drawn from a random number source with expectation 1.0. For this reason the *log-normal update rule* is applied to the step-size vector $\vec{\sigma}$ as follows:

$$\vec{\sigma}' = \left(\sigma_1 \cdot e^{(z_0 + \tau \cdot z_1)}, \sigma_2 \cdot e^{(z_0 + \tau \cdot z_2)}, \dots, \sigma_{n_\sigma} \cdot e^{(z_0 + \tau \cdot z_{n_\sigma})} \right) \tag{2.10}$$

with $z_0 \in \mathcal{N}(0, \tau_0^2)$ and $z_i \in \mathcal{N}(0, \tau^2)\ \forall\, i = 1, \dots, n_\sigma$. Schwefel and Rudolph (1995) recommend setting the *learning parameters* τ_0 and τ, according to:

$$\tau_0 = \left(\sqrt{2\sqrt{n}} \right)^{-1} \tag{2.11}$$

$$\tau = \left(\sqrt{2n} \right)^{-1} \tag{2.12}$$

The order in which the evolutionary operators are applied to the object and strategy parameters is also an important factor in the successful application of self-adaptation. The step-size parameters should be mutated prior to the object parameters, to ensure that any useful mutative step made in the object space is directly attributed to the accompanying step-size vector. The intention here is that the useful strategy parameters that led to the adaptation of strong object parameters are inherited by descendent individuals to deliver even fitter solutions throughout subsequent generations.

Derandomised Self-Adaptation

Ostermeier *et al* (1994) presented a *derandomised* mutative step-size control procedure designed to improve the performance of the original self-adaptation mechanism. The traditional mutative self-adaptive mechanism (outlined above) has been shown to break down when small population sizes are employed (Schwefel, 1987). While these symptoms can be reduced with the use of intermediate recombination and larger population sizes, Ostermier *et al* (1994) set out to tackle the cause of these shortcomings. Two deficiencies in the traditional self-adaptive process were identified:

- Firstly, there is no guarantee that profitable variations in the object parameters will naturally correlate with an equivalent change in the mutation step-size. In other words, it is possible for a small step-size to yield a large parameter variation; if the resulting individual is subsequently selected, the step-size does not reflect the advantageous mutation.
- Secondly, the amount of variation in the strategy parameters is the same throughout all generations; therefore, the procedure for adapting the strategy parameters is set to facilitate effective mutation irrespective of the distribution of the population throughout the search/strategy space. Consequently, the adaptive process produces a large enough variation in step-size parameters to ensure an appropriate selection difference between individuals. In smaller populations this level of variation can lead to large fluctuations in the strategy parameters that can impede the optimisation process.

To ameliorate these problems, Ostermier *et al* derived a derandomised approach to self-adaptation. In the traditional self-adaptive ES mechanism, object parameters are mutated with the addition of the mutation vector $\vec{m}$:

$$\vec{x}' = \vec{x} + \vec{m} \tag{2.13}$$

The derandomised mutation vector $\vec{m}$ is given by:

$$\vec{m} = (\, \xi \cdot \sigma_1 \cdot z_1 \,, \xi \cdot \sigma_2 \cdot z_2 \,, \dots, \xi \cdot \sigma_n \cdot z_n \,) \tag{2.14}$$

where $\xi = \alpha$ or $1/\alpha$ with equal probability determined for each offspring, $\alpha = 1.4$ and $\vec{z}$ is a normally distributed random vector. Derandomised mutative adaptation is then applied to the step-size vector $\vec{\sigma}$ according to:

$$\vec{\sigma}' = \left(\xi^{\beta} \cdot e^{\beta_{scal} \cdot \left(|z_1| - \sqrt{2/\pi}\right)} \cdot \sigma_1, \xi^{\beta} \cdot e^{\beta_{scal} \cdot \left(|z_2| - \sqrt{2/\pi}\right)} \cdot \sigma_2, \dots, \xi^{\beta} \cdot e^{\beta_{scal} \cdot \left(|z_n| - \sqrt{2/\pi}\right)} \cdot \sigma_n\right) \tag{2.15}$$

in which $\beta = \sqrt{1/n}$ and $\beta_{scal} = 1/n$, and the parameters ξ and $\vec{z}$ are the same parameters used to calculate the mutation vector for the corresponding object parameters. With the derandomised mutation operator, it is ensured that the step-size parameter values are always mutated in proportion to the object parameters, with minimal stochastic fluctuations.

For effective operation of the mutative self-adaptation mechanism, the $(\mu\,,\lambda)$ strategy is widely regarded to offer superior adaptive properties when compared with the $(\mu+\ \lambda)$ alternative (Bäck and Schwefel, 1993). This is due to the possibility that a highly-fit offspring is generated with a step-size parameter that is entirely inappropriate for its new location. This may arise when a recombinant with a very large mutation step-size fortuitously *jumps* to a distant and highly fit region of the search space. If the offspring is able to pass directly into subsequent generations (*elitism*), optimization is likely to stagnate as further progress will be thwarted by the originally useful but now unsuitably large step-size. This situation could not arise in the $(\mu\,,\lambda)$ strategy, as the anomalous offspring would expire after transmitting some of its strong genetic material through recombination.

2.3.2.4 Selection

The selection operator in the ES facilitates the drift of the population towards regions of increasing fitness within the parameter space. Selection works in an opposing yet complementary manner to the variation operators and identifies the direction in which search should proceed. As was discussed earlier in this section, selection in the ES is performed deterministically. In the case of the $(\mu\,,\lambda)$ *comma* (or extinctive strategy), the fittest individuals are chosen from the offspring; whereas in the $(\mu+\ \lambda)$ *plus* (or preservative strategy), selection is made amongst both the parent and offspring populations. Schwefel and Rudolph (1995) established the concept of maximal lifespan with the introduction of the exogenous parameter κ to indicate the number of generations for which each individual is permitted to survive. The resulting $(\mu\,,\kappa\,,\lambda)$ strategy provides a generalisation of the deterministic selection scheme, such that when $\kappa=1$ the ES presents an instance of the extinctive comma strategy; furthermore, when $\kappa=\infty$ the resulting ES is equivalent to the preservative plus strategy. The parameter κ may also be set to any intermediary value in between these two extremes $1\leq\kappa\leq\infty$.

2.4 EA Similarities and Differences

Both the ES and GA derive inspiration from biological evolution; however, the specific implementation of each EA is quite different. For example, in the theory that relates to the GA it is assumed that the genes of the optimal solution are scattered throughout the population; evolution is then the process of recombining these genes to produce the optimum. ES theory, on the other hand, assumes that the optimum solution will be located through a processes of organised, but random, mutative leaps through the object space.

As the research field of evolutionary computation has developed, the boundaries that once existed between distinct classes of EAs have begun to erode. This section aims to summarise some of the more recent developments that bring these algorithms closer together.

The canonical GA employs binary encoding for the representation of real-valued object variables. While this type of representation models the processes of biological evolution more closely than real-valued representation, encoding the search space into discrete intervals for binary representation can introduce harmful side effects, which may in turn increase the complexity of the search space (Bäck *et al*, 1997b). When continuous parameters are represented by bit-strings, there are often large discrepancies (*Hamming cliffs*) between the real and encoded parameter spaces. For example, two points might be separated by only a single bit mutation in the genotype space; however, in phenotype space the same points might be positioned very far apart. This problem may be reduced to some degree by employing a *Grey coding*, such that all adjacent points in the phenotype space are separated by one bit-shift in the genotype space. However, it is still possible that inversion of a single bit can result in a large transition in object space.

EP and ESs, on the other hand, traditionally represent object parameters with real-valued numbers. This representational distinction between EAs has become blurred since Wright's (1991) investigation of real-coded GAs, with phenotypic crossover and (ordinal) mutation operators. This augmentation of the simple GA stimulated a succession of real-coded GA publications that circumvented the inherent precision, range-restriction, and Hamming cliff problems associated with binary-coded representation (Herrera *et al*, 1998) (Deb and Beyer, 1999). Conversely, ESs may also operate on bitstrings (Beyer, 2001, p3), (Beyer and Schwefel, 2002). An ES has even been adapted to model the neighbourhood distribution of the *Grey-code* (Rowe and Hidović, 2004). Consequently, individuals may be represented directly or mapped via binary-coding in either algorithm.

The decision as to whether strategy parameters should be adapted as evolution takes place, or remain unchanged throughout the course of evolution was also once a distinguishing factor between different classes of EA. However, GAs have been developed that permit the variation of mutation rates by a form of self-adaptation (Smith and Fogarty, 1996), (Yang and Kao, 2000). Furthermore, an ES has also been developed that applies the traditional mutation scheme according to a GA-style fixed probability rate (Huband *et al*, 2003). Additional examples of self-adaptive genetic algorithms may also be found in Bäck and

Schutz (1996) and Bäck *et al* (2000).

The selection operator may be distinguished from the mutation and recombination operators as it is entirely independent of the search space structure. As such, any selection operator from one evolutionary algorithm may be easily applied to any other. As was shown earlier, the GA traditionally employs a fitness proportionate probabilistic selection operator. However, *tournament selection* (Goldberg and Deb, 1991) as well as *linear ranking selection* (Baker, 1985) methods are also widely employed. On the other hand, the ES regularly adopts a deterministic scheme. However, selection operators have also been shared between these two classes: A *truncation* selection operator has been designed and implemented for use within the *Breeder* GA (Mtihlenbein and Schlierkamp-Voosen, 1993), which is based upon the deterministic techniques employed by human breeders. Furthermore, deterministic selection-based GA developments have also been developed by Affenzeller *et al* (2005) and Eshelman (1990). The *tournament* selection scheme employed by Goldberg and Deb (1991) for use within the GA has also been adopted by the ES, as described in Schwefel and Rudolph (1995). The ES *plus* selection strategy is also modelled by the *elitist selection* or *generation gap* scheme in GAs (De Jong, 1975).

It is clear that the ideas and concepts that once separated the various implementations of the EA are now shared between them. In his book Beyer even goes so far as to state that the algorithms are only separated by the lack of theory that unites them (2002, p3). Recent evolutionary computation publications are frequently concerned with *hybrid* or *haptic* algorithms with ideas gleaned from the optimisation literature without bias. The relative merits or detriments of one class of EA compared with another is a discussion which will not appear here. The EAs proposed throughout chapters four and five are applied within the framework of the ES; generalisation could easily be made to the GA but such developments are beyond the scope of this thesis. For a side by side comparison of the three main classes of EA (GA, ES and EP) see Bäck and Schwefel (1993).

2.5 Summary of this Chapter

In this chapter the computational model of evolution was reviewed, with details of how the model may be applied to optimise static real-valued problems. The specifics of the GA and ES were introduced with a brief summary of their historic developments and current state. Detailed implementation specifics were provided for ES, as this forms the theoretical framework within which the algorithmic developments documented in chapters four and five of this thesis are based. The following chapter reviews a wide range of EA developments that are designed to improved optimisation performance in complex multimodal search environments.

Chapter 3

Background - Multimodal Optimisation

Since the introduction of EC over 40 years ago, there has been growing interest in the application of EAs to an ever-increasing range of parameter optimisation problems. EAs have been shown to be robust, reliable and straightforward to apply even when there is very little *a priori* knowledge of the underlying problem domain. However, in search space environments containing multiple distinct optima, EAs can often fail. This chapter reviews an EA pathology known as preconvergence and summarises the algorithmic attributes that result in this shortcoming. A range of techniques are then reviewed which have been designed to minimise the likelihood of preconvergence.

3.1 Multimodal Problem Domains and Preconvergence

As described in chapter two, EAs operate through the maintenance of a finite population of solution candidates. Each candidate represents a sample taken from the fitness landscape of the application domain. At the early stages of optimisation, samples are distributed sparsely throughout the search space. On the basis of the topographical information afforded by the landscape samples, regions of the space are identified within which subsequent search will proceed. Through the process of selection (which over time allows fitter solutions to dominate the population), the population will gravitate towards regions of high *cumulative payoff*, enabling fitter regions of the search space to be sampled at increasing resolution.

Consequently, standard evolutionary algorithms are easily *deceived* by multimodal problem domains in which a single optimal peak is located within a subspace of below average *payoff* (Whitley, 1991), (De Jong, 1993). This behaviour is not surprising; rather, it is a consequence of the underlying search process. The fitness landscape may only be observed though the search points of the population, if an optimal peak (or its surrounding gradient) is not sampled, it is effectively invisible. Even if the gradient of the optimal peak is sampled once, there is no guarantee that it will be sampled again throughout subsequent generations. If too much time is invested in finding the whereabouts of hidden peaks (exploration), progress rates are compromised. Conversely, if the population focuses on high performance regions too rapidly (exploitation), suboptimal convergence becomes highly probable, and the robustness of the EA is compromised. A balance must be struck between the interplay of exploration and exploitation. This precipitous focusing of the population is the fundamental cause of what is frequently referred to as *premature convergence*, or *preconvergence*. If the population preconverges at a suboptimal peak, further progress is precluded except as a result of random mutation. This problem is most prevalent when the application domain of the EA is highly multimodal.

The propensity for traditional EAs to converge at a single point is endemic to their architecture and will arise even when search space optima are of equal magnitude. When there is no selective advantage between peaks, the choice will be arbitrary due to the stochastic nature of the variation operators (Schönemann *et al*, 2004). Even when peaks are unequal, optimal convergence is not guaranteed, due to sampling errors which may take place during the exploratory phase of optimisation (Preuss *et al,* 2005).

Since the recognition of these preconvergence issues, a variety of techniques have been developed to enhance the performance of ESs within multimodal problem spaces. The principal aim of these techniques is to preserve a diverse selection of genetic material to facilitate sufficient exploration of the search domain prior to convergence.

3.2 Injecting Diversity

Several variations on the traditional EA model counterbalance the loss of diversity (leading to convergence) with the continual introduction of novel genetic material.

One approach is to ensure that each offspring satisfies a measure of *uniqueness* before being accepted into the population. Offspring that fail to meet the required criterion are systematically mutated until they are sufficiently different from the rest of the population. This technique was adopted by Whitley and Kauth (1988), and Mauldin (1984) to improve the performance of the simple GA. Mauldin's GA applies a variable uniqueness requirement which is decreased throughout the course of evolution; the assumption being that diversity is most important in the early stages of evolution. Gradually reducing the uniqueness level ensures the eventual convergence of the population at a single point. This approach was found to improve the off-line performance of the GA.

Similar results may be achieved by adopting very high rates of mutation (Grefenstette, 1986). Cobb (1990) introduced a *hypermutation* system which comes into effect when it is assumed that diversity is being lost. The traditional GA system is employed whilst fitness is progressing, but when there is a measured decline in progress (population convergence), the GA switches into a *hypermutation* mode (high mutation rate) to restore diversity.

The sudden introduction of new genetic material has a similar effect to the complete re-initialisation of the population, a method examined in Krishnakumar (1989) and Mathias *et al* (1998), termed *cataclysmic mutation* by Eshelman (1990). In other circumstances mutation has been substituted for an entirely stochastic system, in which randomly generated solutions are inserted directly into the population as evolution takes place (Bonham and Parmee, 2004).

These injection approaches to diversity preservation have been criticised as addressing the symptom of the problem rather than the cause. The question then arises: what are causes of diversity loss in traditional EAs? In his PhD thesis Mahfoud (1995) extensively examined the primary causes of diversity loss within the GA. Shir and Bäck (2005) later reconsidered Mahfoud's observations from an ES perspective. There are three major factors that lead traditional EAs towards suboptimal convergence:

- ***Selection Pressure*** - The rate at which weaker individuals are discarded from the population is controlled by the selection pressure. Central to this theme is the concept of *takeover time,* introduced by Goldberg and Deb (1991), which is defined as the number of generations that have elapsed before the population contains only duplicates of the best individual (no population diversity). Raising the selection pressure will produce a corresponding reduction in takeover time, and consequently, an increase in the likelihood of suboptimal preconvergence.
- ***Genetic Drift*** - Genetic drift describes the stochastic process that causes loss of diversity within finite populations, which is introduced by the selection operator. As a limit is placed on genes which propagate to succeeding generations, there is a tendency for the population to approach homogenisation. Subsequent recombination amongst a finite number of offspring can enable certain genes to dominate a population, even when there is no selective advantage (Schönemann *et al*, 2004).
- ***Operator Disruption*** - Operator disruption describes the destructive effects that the variation operators can have on well adapted genes. This can simply occur when mutation has a negative affect on an individual, or when recombination yields offspring of lower fitness than their progenitors.

From the factors summarised above, it is clear that injecting diversity into the population does not directly address the principal causes of diversity loss; indeed, the use of increased mutation rates serve only to aggravate problems associated with operator disruption. Furthermore, from the standpoint of the building block hypothesis (Goldberg, 1989), increasing the mutation rates can only be viewed as counter-productive (see section 2.3.1).

Therefore, introducing diversity purely to prevent convergence is not the solution; as Goldberg and Richardson state:

> *…we need to maintain appropriate diversity--diversity that helps cause (or has helped cause) good strings.*
>
> *Goldberg and Richardson (1987)*

The following sections review many augmentations of the traditional EA model, which seek to improve performance within multimodal problem domains through the maintenance of *appropriate diversity*.

3.3 Appropriate Diversity

While the injection of novel genetic material does succeed in maintaining diversity, it fails to maintain appropriate diversity as described by Goldberg. As the selection and variation operators work to navigate the population towards ever-increasing regions of fitness, the application of high mutation rates serves only to hamper the optimisation procedure. The purpose of maintaining diversity is not just to delay convergence, but to ensure that the search space is sufficiently characterised prior to convergence.

Central to the maintenance of appropriate diversity are the concepts of *niche* and *species*. Although loosely defined, the term species is used to refer to solutions that share similar characteristics, and niche to refer to the region within the search space that a species occupies.

Numerous interpretations of these concepts have been implemented in the EA literature, all of which incorporate some notion of population division. Partitioning the population into groups of individuals encourages the parallel investigation of multiple distinct search space sub-domains, which often serves to delay convergence; however, it is important to note that this is not the aim. Restriction of local competition between population subdivisions (species) can result in an overall reduction in selection pressure, while local variation amongst subdivision members reduces operator disruption.

3.4 Speciation

If the intention is to maintain appropriate diversity, by encouraging the formation of population species, the traditional EA architecture must be adapted in some way. When applied to bimodal problems with equally sized peaks, traditional EAs are unable to maintain both optima, regardless of the population size (Mahfoud, 1995). As appropriate diversity could be maintained by enabling population members to form into species, a second question is raised: what are the properties of traditional EAs that prevent the formation of species? In essence, this question has been addressed implicitly in the literature that relates to niching and speciation EAs. The two major factors are as follows:

- ***Optimistic Selection*** - Traditional EA selection operators consider only fitness when identifying those members of the population to partake in recombination. As such, it is possible for a single adaptation with high relative payoff to dominate the population before other regions of the space have been sufficiently explored. To enable the formation of species, it is required that the selection/replacement operators consider not only fitness, but also the location of each individual with respect to the rest of the population; whether this spatial consideration be in the search domain or internal to the EA.

- ***Recombination Disruption*** - Recombination has the power to destroy, as well as unite, the beneficial traits of individuals. When used to optimise a multimodal problem space, traditional recombination, acting globally on the population, will attempt to blend genetic material from individuals that represent independent search space peaks without bias. The corresponding recombinant will thus characterise some midpoint between contributing individuals, and is not guaranteed to occupy any of the peaks represented by the parental set. These disruptive effects of global recombination may be reduced by modifying the EA to ensure that mating only takes place locally between population members within the same niche, thus creating species.

The remainder of this chapter considers a variety of speciation techniques that provide improved performance within multimodal problem domains by addressing one or both of the factors identified above. Frequently this results in more reliable location of the optimum, and/or the simultaneous maintenance of multiple solutions positioned at distinct peaks within the search space.

3.4.1 Non Partition-Based Speciation Methods

In the first group of speciation methods reviewed here, the notion of species is not imposed upon the evolutionary architecture in any way; rather, species are induced either by carefully selecting and replacing population members, or by *warping* the internal perception of the fitness landscape.

3.4.1.1 Similarity-Based Selection/Replacement

Species may be encouraged to form within the population by implementing a similarity-based selection procedure to ensure that a diversity of genetic material is maintained within the population. While similarity measures were adopted in the mutation-based approaches reviewed in section 3.2, this approach differs as diversity is maintained by a controlled selection mechanism, rather than a stochastic mutation procedure.

Crowding

In his doctoral thesis, De Jong (1975) developed a series of GA-based algorithms, to improve performance when facing a variety of problem characteristics. When experimenting with an elitist model, in which the fittest individual of one generation automatically passes into the next, De Jong noted a performance improvement over the baseline GA when optimising unimodal test functions. However, a performance decrease was observed when optimising a multimodal test function. Preservation of the fittest individuals was found to deliver improved local search at the expense of global search, and thus the likelihood of suboptimal preconvergence was increased. To generate improved results within a multimodal test environment, De Jong proposed an augmentation to the conventional GA mechanism called *crowding,* which formed one of the earliest attempts to preserve population diversity and thus promote increased search space exploration.

The crowding scheme implements a *steady-state* GA, in which new offspring are accepted into the population immediately following their creation. To maintain a fixed population size, new offspring replace existing population members on the basis of a bit by bit (genotypic) comparison or *Hamming distance* measure. By replacing like with like, a diversity of genetic material is encouraged to persist within the population, facilitating the formation of multiple species.

Deterministic Crowding

To improve efficiency of the crowding technique, offspring may be compared for likeness with a set of randomly chosen individuals rather than the entire population. The size of the set is controlled by a *crowding factor* parameter. A low crowding factor leads to a degree of stochastic error in the replacement process: offspring frequently replace parents from a different niche or subspace of the application domain (*replacement error*). While crowding is, to some extent, successful at maintaining genetic diversity, this replacement error leads to the movement of the population towards fixed bit positions through genetic drift. To circumvent these problems, Maulford (1992) presented a *deterministic crowding* scheme in which similarity is measured at the phenotypic level (*Euclidian distance*), and offspring may only replace their parents when there is an improvement in fitness. Deterministic crowding was found to provide a significant improvement over the original crowding method, resulting in the near extirpation of replacement errors, and the maintenance of solutions around all of the peaks in the tested multimodal functions.

3.4.1.2 Restricted Tournament selection

As described in section 2.4, tournament selection provides a stable mechanism for identifying population members to take part in recombination. In stochastic binary tournament selection, pairs of randomly chosen population members compete for places in the mating pool. On average, the fittest individual is accepted twice, while the weakest individual is discarded. This procedure enables the selection pressure to be controlled (Miller and Goldberg, 1995), preventing the sudden proliferation (*takeover*) of a relatively strong individual; a problem to which *roulette wheel* selection is susceptible.

Harik (1995) later developed a *restricted tournament selection* mechanism to enhance GA performance within multimodal problem domains. The model encourages the preservation of appropriate diversity by taking measures to restrict competition between individuals that do not belong to the same niche. As in the crowding approach, a steady-state GA is adopted in which each new offspring competes with selected population members for a place in the population. However, in the restricted tournament scheme, the competing individual is selected from w (*windowsize*) randomly chosen population members as the most (phenotypically) similar to the new offspring.

3.4.2 Fitness Sharing

In natural evolution, species develop as a result of conflict for finite resources. A species will grow until the environmental resources on which it relies for subsistence become scarce. When a niche becomes saturated, new species will emerge that exploit environmental resources which are relatively uninteresting to other species.

Goldberg and Richardson (1987) modelled the processes of natural speciation within the GA by introducing a fitness sharing procedure that encourages population members to explore multiple search space optima simultaneously. Species are induced by devaluing the performance of each individual according to the quantity of neighbouring population members that fall within a defined sharing radius σ_{share}. Once a group of individuals are positioned within a high performance search space peak, the local resources are shared, encouraging the remaining population members to pursue alternative optima. This sharing procedure has the effect of adaptively warping the internal perception of the fitness landscape to promote increased exploration.

It was later demonstrated by Deb and Goldberg (1989) that the fitness sharing GA was able to maintain multiple distinct search space optima more reliably than De Jong's original crowding GA. Furthermore, to minimise the production of so-called *lethal* offspring, through disruptive recombination, a novel mating procedure was also introduced. *Restricted mating* operates by selecting an individual from the population, followed by a potential mate. If the chosen individual and mate occupy the same niche (i.e. within the predetermined sharing radius), recombination is performed, otherwise an alternative mate is chosen. Goldberg and Deb's restricted mating scheme was shown to provide an on-line performance improvement over the traditional GA recombination operator on the tested problem domains.

Since its introduction, several drawbacks to the fitness sharing approach have been noted:

- *a priori* knowledge of the application domain is required in order to select an appropriate value for σ_{share} (Goldberg and Deb, 1989).
- for the most effective results, optima should be distributed evenly throughout the search space (Hocaoglu and Sanderson, 1997) and of equal size (Gan and Warwick, 1998).
- sharing can often result in individuals forming around search space optima (Bäck, 1996), due to the warping of the fitness landscape.

Numerous efforts have since been developed which attempt to ameliorate one or more of the drawbacks identified above, notable authors include Yin and Germay (1993), Lin and Yang (1998), Lin *et al* (1998), Lin and Wu (2002), Miller and Shaw (1996), Goldberg and Wang (1997), Gan and Warwick (1998) and Gan (2001), some of which will be reviewed in section 3.6.1.

3.5 Static Partition Speciation Methods

Rather than promoting the formation of species through an inductive procedure, an alternative approach is to include a notion of parallelism within the EA model. Division of the population into isolated subpopulations which evolve concurrently directly addresses problems associated with optimistic selection and recombination disruption.

As noted by Harik (1994), many of the methods for conserving diversity by dividing an EA population into spatially separated subpopulations (*demes*) derive inspiration from the *Shifting Balance Theory* posed by Sewall Wright (1969). Wright conjectured that, by separating the population into smaller subpopulations, individuals will range more widely over the landscape, and the *diluting* effects of blending highly fit genes with the population at large are reduced. In other words, strong adaptations are more likely to persist within a deme than a global *(panmictic)* population. There are two major classes of static partition parallel population EAs:

- Coarse-grained *island* models.
- Fine-grained *diffusion* models.

3.5.1 Coarse-Grained Parallel Population Methods: The Island Model

Within a coarse-grained EA, each deme evolves in relative isolation except for the occasional exchange of individuals at a fixed rate of *migration*. Once a strong individual has dominated its respective deme, its spread is restricted until the subsequent migration phase, whereupon the genes of this well-adapted individual may permeate into neighbouring demes. This enables alternative evolutionary pathways to progress naturally, a process which may otherwise have been disrupted had all individuals been forced to compete globally. A visual representation of the island model is depicted in figure 3.1.

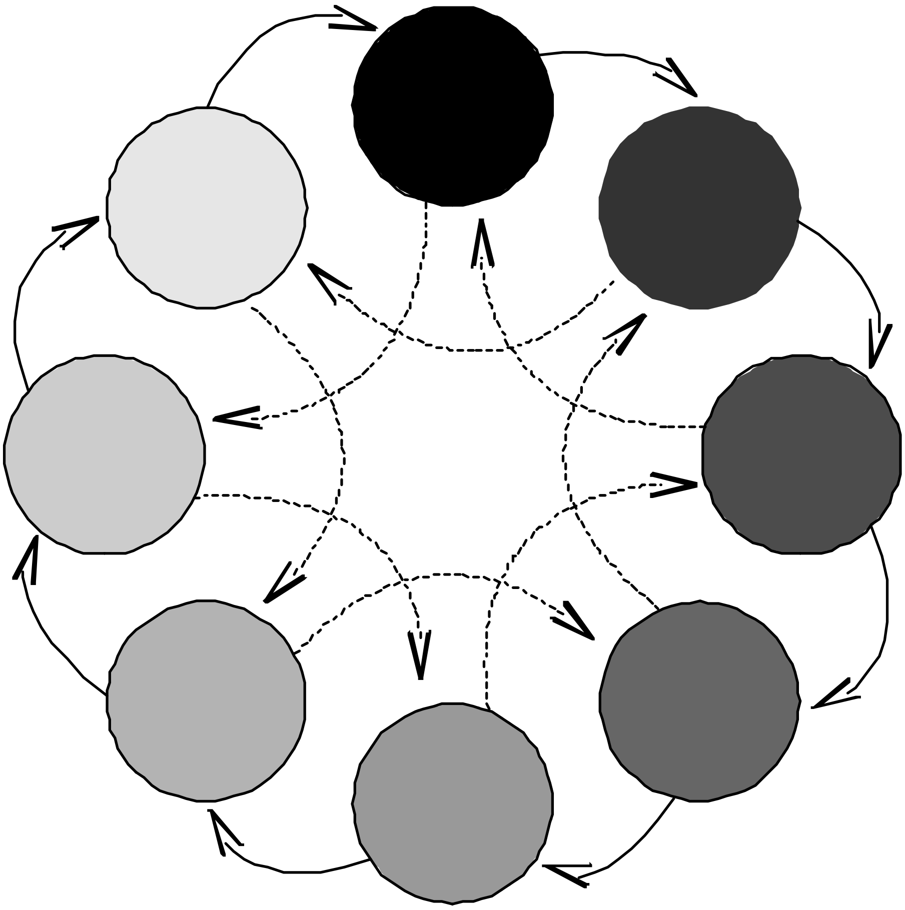


Figure 3.1: Island model

Tanese (1989) compared a distributed GA with both a simple and partitioned GA (without migration), concluding that both the distributed and partitioned GAs outperform the simple GA to an equal extent. However, the distributed GA was found to position a higher proportion of individuals at the global optimum. In a more recent study, Whitley *et al* (1999) demonstrated that when optimising certain application domains, a partitioned system with migration possesses a significant advantage over a partitioned system without migration.

A comparable GA-based model was presented by Whitley and Starkweather (1990), named GENITOR II. The migration procedure differed from that of the distributed GA as, upon migration, the *k* best individuals from the source deme are selected to replace the *k* weakest individuals of the destination deme. This optimistic migration method results in an increased selection pressure over the purely stochastic migration procedure of the distributed system.

Island model EAs are not restricted exclusively to the GA paradigm; distributed versions of the ES have also been implemented, see for example Rudolph (1991) and Lohmann (1991). The major distinguishing factors between the various implementations of the island model are the quantity of demes and the choice of migration interval. Whitley *et al* (1999) investigated the effect that these factors have on the performance of an island model GA, concluding that the optimal settings are highly coupled with the underlying structure of the application domain.

Herdy (1992) experimented with an adaptive migration interval *(isolation period)* within a hierarchically organised ES, which later became known as the Meta-ES (Beyer and Schwefel, 2002). At the lowest level of the Meta-ES hierarchy, a conventional ES maintains a population of candidate individuals. At the next level of abstraction, the species level, a number of competing populations are evolved, again using a conventional ES. At this level the individual is no longer the selective unit, selection acts at the population level, where a single species comprises multiple populations. At the uppermost level, the genus level, a population of species is maintained, and again optimised by ES, with selection favouring those species that contain populations performing the best. This rather complex arrangement is illustrated in figure 3.2.

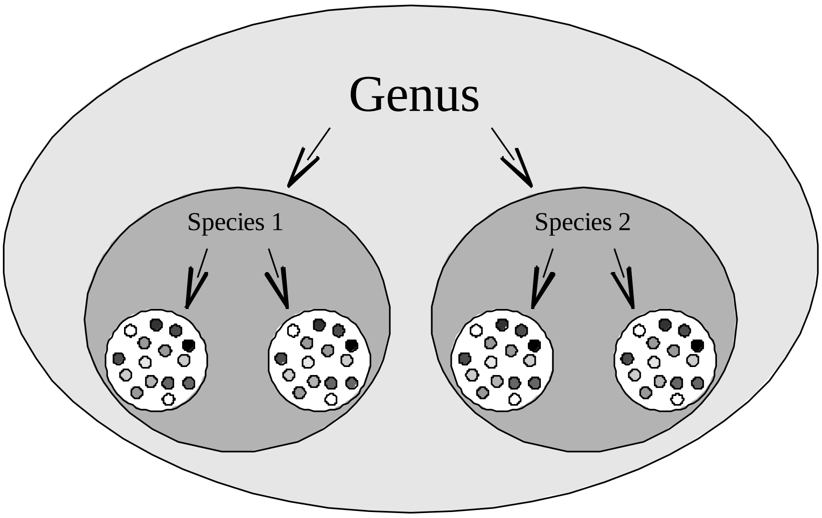


Figure 3.2: Meta-ES

A species may be considered analogous to an island; however, within the Meta-ES framework, species contain a pair (sometimes more) of competing subpopulations. Pairs of subpopulations within each species are arranged such that one subpopulation emphasises exploration, while the other emphasises exploitation. Upon migration, at the lowest level, members of the best subpopulation are duplicated to replace those of the worst subpopulation. Individual mutation step-sizes are then adjusted such that members of one subpopulation are assigned values above the average mutation step-size, while members of the other subpopulation are set below the average (exploration and exploitation respectively). At the species level migration interval, all species are set to the best, and the migration intervals are adapted such that the isolation period in one species is increased while the other is decreased. This complete process facilitates the self-adaptation of both the step-sizes and migration intervals. For fuller treatment see Herdy (1992).

3.5.2 Fine-Grained Parallel Population Methods: The Diffusion Model

Fine-grained EAs, like their coarse-grained counterparts, also divide the population to ensure that competition and mating take place only between individuals belonging to the same deme, or *neighbourhood*. Each individual occupies a unique neighbourhood, which includes population members positioned in the immediate locality, where the population is often arranged in a linear, planar or higher-dimensional spatial topology. Unlike the island model, demes are not entirely disjoint; rather, they overlap adjacent demes, and thus incorporate an implicit notion of migration. Highly fit individuals are able to pervade the population through a steady *diffusion* process. This architecture delays total population convergence enabling genetic pathways to progress in relative isolation prior to global competition. A fine-grained EA population is illustrated in figure 3.3; the depicted model adopts a planar Moore neighbourhood.

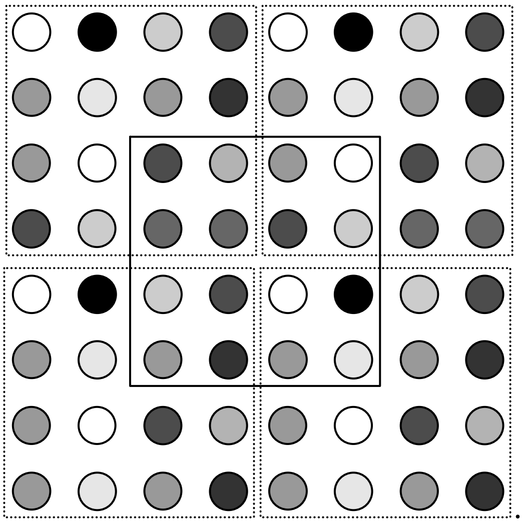


Figure 3.3: Fine-grained architecture

Gorges-Schleuter (1989) developed a fine-grained EA named ASPARAGOS, which was applied to optimise the Travelling Salesman problem. In the ASPARAGOS system the population is arranged on a toroidal ladder, and each deme is defined as the reproductive community that surrounds each population member. In turn, each individual is chosen from the population to act as a parent and paired with a mate from within its neighbourhood. Resulting offspring then replace the parent if they are of higher fitness, otherwise an alternative mate is selected and the process repeats.

Several other derivatives of the fine-grained EA may be found in the literature; for example, GA- and ES-based diffusion models have been explored by Mühlenbein (1992), and Sprave (1994) respectively. Furthermore, a comparison between traditional and fine-grained EAs may be found in Manderick and Spiessens (1989).

3.5.3 Discussion

The benefits of these *diffusion* and *island* models are twofold. Not only do they facilitate the preservation of diversity for a period that enables increased exploration of the search space, but these models may also be deployed easily on parallel computing hardware. The island model, for example, may be easily implemented on a *coarse-grained* computational architecture, in which each processor is allocated an entire deme for processing. The diffusion model, on the other hand, is well suited for implementation on a *fine-grained* computational architecture, in which each individual is assigned to a processor within a massively parallel arrangement. Deployment of EAs on parallel computational hardware is not of interest here, so this subject is not addressed any further within this thesis. For more detailed information and a comparative analysis of these partitioned EAs, see Gordon and Whitley (1993).

While these static partition models enable species to emerge throughout the course of evolution, migration (coarse-grained) and diffusion (fine-grained) will ultimately lead the population to a state of homogeneity. When migration is frequent, global convergence is accelerated. Conversely, when migration is omitted subpopulations are able to locally converge at independent optima (*niching*). However, without migration, the system is less robust and often highly redundant, as subpopulations may converge to the same niche independently. An alternative approach to population division is to derive partitions dynamically, based upon the distribution of the population within the search space. These methods can be grouped together and referred to as dynamic-partition speciation methods.

3.6 Dynamic-Partition Speciation Methods

Rather than educing species by warping the fitness landscape or by imposing a static geographic scenario upon the population, a third approach is to identify structure which may be emerging within the population and arrange partitions dynamically as evolution takes place.

3.6.1 Cluster-Based Partition Methods

By interleaving the evolutionary operators with a cluster analysis procedure, partitions may be defined that group individuals occupying the same niche.

Clustering was first introduced to EC by Yin and Germay (1993) to ease the application of Goldberg and Richardson’s (1987) fitness sharing procedure (see section 3.4.2) when there is little or no *a priori* landscape information. Two disadvantages of the original fitness method were identified:

- the requirement for the parameter σ_{share} to be set relative to the number of peaks.
- the assumption that all peaks are equally distributed throughout the search space (Deb and Goldberg, 1989).

To alleviate these shortcomings, Yin and Germay incorporated an adaptive k-means clustering procedure within the traditional GA to identify individuals that belong to the same species. The fitness of each offspring is then abated according to the formula:

$$f'_i = \frac{f_i}{m'_i} \tag{3.1}$$

with

$$m'_i = n_c - n_c \left(\frac{d_{ic}}{2d_{max}}\right)^{\alpha} \tag{3.2}$$

Where n_c is the number of individuals belonging to cluster c, d_{ic} is the distance between the individual i and the centroid of the cluster, and α controls the linearity of the sharing function, which is set to 1 (linear) in Yin and Germay's experiments. The distance parameters d_{max} and d_{min}, define the maximum and minimum cluster radii. Judicious choice of these parameters is necessary, as the algorithm is highly sensitive to error. For example, if d_{max} is too large, only a small number of clusters will form and fine peaks may be lost. Conversely, if the distance parameter d_{min} is too small, a superfluity of clusters may deteriorate the overall performance. A simple mating restriction scheme based upon the system employed in Deb and Goldberg (1989) was also applied to prevent inter-species breeding which was found to produce improved results.

Although Yin and Germay's clustering approach to sharing was repeatedly shown to outperform the original scheme presented by Goldberg and Richardson (1987), Lin and Wu (2002) noted shortcomings to the approach and proposed improvements. While clustering is appropriate for identifying distinct groups of individuals in the search space, it is unable to determine whether all members of a given cluster occupy the same niche (or peak). To circumvent this problem a *niche identification technique* was developed which groups individuals based upon their relative fitness[2] as well as their search space proximity; for example, an individual with surrounding neighbours of lower fitness defines a niche centre, while an individual with surrounding neighbours of higher fitness defines a niche boundary (see figure 3.4). Subsequent to the niche identification procedure, the fitness of each individual is suppressed (shared) according to the number of population members belonging to the same niche.

[2] Several examples of algorithms in which diversity is preserved by observing the distribution of fitness throughout the population may be found in Ursem (1999) (reviewed in section 3.6.2), Hutter (2002) and Legg *et al* (2004).

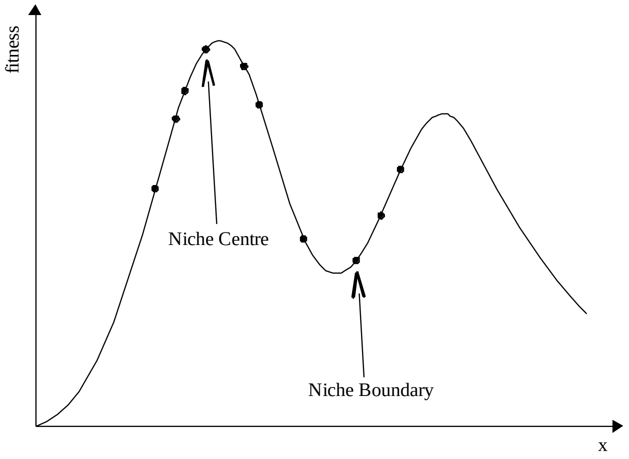


Figure 3.4: Niche identification technique

Prior to the publication of the niche identification technique, Lin presented two alternative fitness sharing approaches. In the first approach (Lin and Yang, 1998), individuals are clustered according to their search space density with the application of a *crowdedness function*. In the second approach (Lin *et al*, 1998), the *crowdedness* function is reapplied in combination with a local search operator. At a time when either *N* generations have elapsed or a cluster has stabilised, a local search algorithm is seeded with the cluster centre. The located peak is then removed from the search space, and all individuals within the corresponding cluster are relocated at random.

Hocaoglu and Sanderson (1997) incorporated minimal representation size cluster analysis (MRSC) into the GA for evolving unknown multimodal optimisation problems. This algorithm follows a comparable procedure to the island model, maintaining multiple subpopulations that evolve separately with the occasional *Cross-Species-Interaction* (crossover). However, unlike the island model, the number of subpopulations is not fixed, it is varied adaptively with the application of MRSC analysis to the merged population. This approach reduces redundancy in the parallel population model as subpopulations found to be converging upon the same optimum will be merged. Each subpopulation is assigned a fixed number of individuals, resulting in a general population size which varies according to the number of identified clusters. The periodic application of cluster analysis facilitates the formation of species at the cost of increased computational load. The MRSC GA was found to maintain distinct solutions for two multimodal test problems and to provide multiple distinct solutions to a mobile robot path-planning problem.

An ES-based clustering optimiser was presented by Sullivan (2001) in which the local search properties of the ES were combined with the strengths of fuzzy cluster analysis. By partitioning the search population into clusters of individuals that locally recombine and progress, it was demonstrated that the fuzzy clustering ES (FCES) was able to evolve optimal solutions when applied to a variety of multimodal test problems. Fuzzy clustering was applied previously for objective function optimisation to deliver multiple start points for a local (hill-climbing) optimiser (Törn, 1986). The FCES adopts the same approach, replacing the local optimisation algorithm with an ES. Optimisation then proceeds with alternate application of cluster analysis and the evolutionary operators. The fuzzy-recombination operator enables species to form within the population by blending the genes of parents in proportion to their membership of a given cluster.

The selection mechanism within the FCES conforms to the global procedure of the traditional ES; the consequent global selection pressure drives the population towards the niche that offers the highest payoff. FCES is therefore unable to optimise multiple distinct search space optima simultaneously, as only one of the factors precluding species formation is addressed (recombination disruption). Later, in section 4.2.1, the FCES model is extended with a local cluster-based selection procedure, to facilitate the long-term maintenance of multiple distinct search space solutions.

A second of Törn's (1977) cluster-based global optimisers has also been adapted for evolutionary search. The original algorithm adopted a density-based clustering algorithm to extract seed points for the subsequent phase of local search. Cluster analysis identifies those search points approaching the same peak (niche), and reseeds all but the fittest search point, which is optimised locally thereafter. This technique was adapted by Hanagandi and Nikolaou (1998), to replace the local search operator with a GA, an approach which is also employed by Pétrowski (1996), and Damavandi and Safavi-Naeini (2003).

A notable extension to the density-based clustering models identified above has been presented by Streichert *et al* (2003). This algorithm maintains a variable number of sub-populations which are determined dynamically by cluster analysis. This is a technique which is comparable with the MRSC GA reviewed earlier (Hocaoglu and Sanderson, 1997). Streichert's *Cluster-Based Niching* method begins with a single, undifferentiated population. The algorithm then enters a *species differentiation phase* in which subpopulations are derived by a *density-based* clustering procedure. Any population members not assigned to a cluster *(loners)* are aggregated and evolved within their own subpopulation. Evolution then proceeds with competition and recombination restricted between subpopulations. At the species differentiation phase, the clustering algorithm is also applied within each subpopulation to identify new species which may have emerged. To counterbalance the division of subpopulations, species may also be merged if they are found to be converging towards the same niche. The density-based cluster analysis associates individuals within an Euclidean distance of less than σ_{dist}, which was found to be quite sensitive to error. With an appropriate setting for the parameter σ_{dist}, the cluster-based niching ES was found to perform well compared to a multi-start hill climber and a fitness sharing ES.

3.6.2 Alternative Dynamic-partition Methods

There are alternative approaches for adapting the location of subpopulation boundaries which do not rely on cluster analysis to derive subpopulations.

The Multinational GA (Ursem, 1999) maintains multiple subpopulations, the positioning and quantity of which are determined dynamically as evolution progresses. Mating and selection are restricted between individuals belonging to different subpopulations, known as *nations*. To ensure that each nation only inhabits a single peak, the landscape topology between respective nations is examined by *hill-valley detection*. If multiple nations are approaching the same peak they are merged. To counterbalance the merging of nations, new subpopulations are created when multiple peaks are detected within the same nation, again using the hill-valley detection procedure.

The forking GA (Tsutsui and Fujimoto, 1993) monitors a global population of individuals for the emergence of dominating schemata. When dominance is observed, a subpopulation is spawned composed from individuals exhibiting the identified schema. Search continues in the contracted search space of the non-fixed loci of the schema until a local optimum is found. The global population then pursues search within the remaining schemata.

3.7 Multimodal Optimisation with Cooperative Coevolution

Cooperative coevolution (Husbands and Mill, 1991) forms a multiple population optimisation method which differs significantly from those reviewed already within this chapter. In earlier examples, the definition of a species, in the context of parameter optimisation, is fairly consistent: a collection of potential solutions, which are in some way classified to be similar. Within the Cooperative coevolutionary paradigm, a species represents a subcomponent of a complete solution to a given problem (Potter and De Jong, 1994). A i decomposition for many optimisation problems is to define each parameter, or dimension of the problem, as a separate species, although alternative decompositions may be equally as valid. Each species is then assigned a subpopulation of competing solutions to the problem component. Each subpopulation is evolved concurrently, but in isolation from the rest; the assumption being that it is easier to find good components and assemble them, than it would be to solve the complete problem directly. The cooperative coevolution architecture is depicted in figure 3.5.

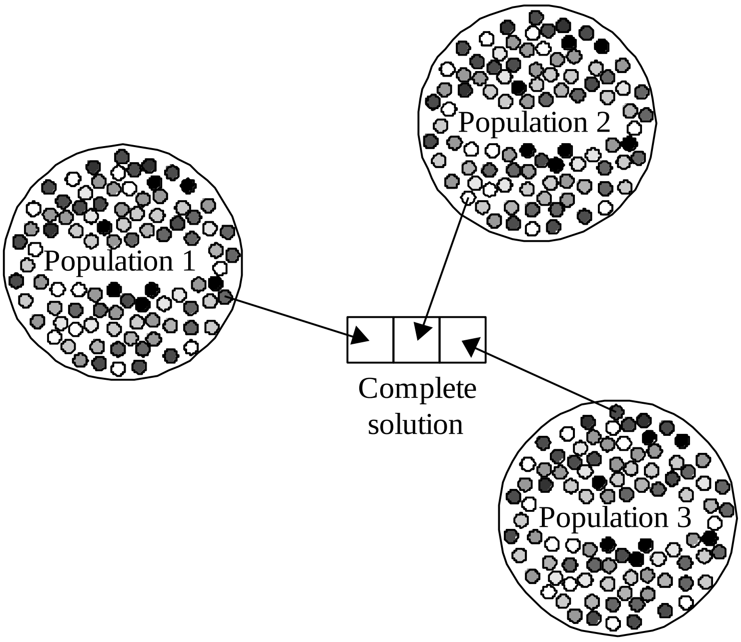


Figure 3.5: Cooperative coevolution architecture

To evaluate each individual for fitness, subpopulation members are adjoined with collaborators chosen from each of the other subpopulations to construct a complete solution to the greater problem. In Potter and De Jong’s (1994) cooperative coevolutionary model, initial fitness values are obtained by combining each subpopulation member with a randomly selected individual from each of the other subpopulations. The resulting solution is then applied to the target function to yield a corresponding fitness rating.

In the first of two collaboration schemes, identified by Potter and De Jong (1994) (CCEA-1), fitness evaluations are performed by amalgamating candidate offspring with the single-best collaborators from neighbouring subpopulations. Each subpopulation is then sequentially coevolved by a traditional EA. In the same paper, Potter and De Jong describe an experimentally verified weakness in this credit assignment procedure, which appeared only to work well on problems with entirely independent function variables. In fact, for one test problem, exhibiting high inter-subpopulation dependencies, the CCEA-1 was outperformed by a simple GA. A second scheme was thus proposed (CCEA-2), equipped with an enhanced credit assignment procedure. Candidate individuals are evaluated twice: once in combination with the best collaborators from each of the other subpopulations, and a second time in combination with randomly-selected collaborators from each subpopulation. The fitness of the better performing combined solution is then assigned as the individual's fitness.

Experimental results were presented on a set of multimodal problems in which the cooperative coevolutionary algorithm was shown to outperform a traditional EA, in terms of both convergence speed and the quality of final solutions. By evolving each species in isolated subpopulations, selection pressure does not result in the convergence of all species toward the same niche; rather, they converge within their own individual niche, based upon their collaborations with other species. This evolutionary pressure encourages individuals to make a unique contribution to the larger problem by interacting cooperatively with neighbouring sub-populations, a process Potter and De Jong refer to as *mutualism.*

When compared with traditional evolutionary algorithms, CCEAs have shown potential for tackling a variety of parameter optimisation problems; consequently, there is growing interest in the application of cooperative coevolution within the domain of function optimisation (Potter, 1994), (Jansen and Wiegand, 2004), (Bucci and Pollock, 2005) and (Iorio, 2002). However, only recently are researchers beginning to be understand how the model works, and when it may prove useful. The majority of this progress been presented by Wiegand *et al*. (2001), Jansen and Wiegand (2004), Wiegand (2004), and Jansen (2004). This work will be reviewed in chapter five, where an enhanced cooperative coevolutionary algorithm is presented and tested.

3.8 Summary of this Chapter

This chapter summarises the difficulties that EAs often face when optimising multimodal objective functions. The properties of EAs which often lead to suboptimal preconvergence were outlined and the importance of species and niche in the maintenance of appropriate population diversity were recognised. A review of different speciation and niching techniques was then followed by a brief discussion of the cooperative coevolutionary architecture. In chapters four and five, the concepts and ideas from this chapter are synthesised and developed to form two novel algorithms, designed to provide robust optimisation of multiple distinct solutions, in rugged, multimodal search environments.

Chapter 4

A Clustering-Based Niching Evolution Strategy

This chapter describes the first algorithmic contribution of this work: an ES-based *niching* optimiser. The algorithm develops Sullivan's Fuzzy Clustering Evolution Strategy (FCES) (reviewed in section 3.6.1) extending the model in such a way as to preserve diversity between clusters, facilitating the concurrent maintenance and optimisation of multiple distinct high-performance solutions.

An extensive review of the FCES is provided, followed by a description of the new modifications and operators that offer improved performance over Sullivan's original. Thereafter, empirical analysis is provided documenting the performance of the new algorithm in application to selected benchmark test functions. In Section 4.3.1 the algorithm's ability to locate the global optimum within environments composed of many local optima is tested. In Section 4.3.2 the ability of the algorithm to identify multiple high performance optima is assessed. Finally, in section 4.3.3 the scalability of the algorithm to higher-dimensional search spaces is evaluated.

4.1 The Fuzzy Clustering Evolution Strategy (FCES)

FCES is an evolutionary optimisation algorithm designed to reduce the likelihood of local preconvergence by incorporating the notion of species into the canonical ES model by means of cluster analysis, which takes place prior to recombination. The algorithm pseudocode is provided in figure 4.1:

```
t = 0
initialise(μ(t));
loop begin
       cluster(μ(t))
       λ(t) = recombine(μ(t));
       λ(t) = mutate(λ(t));
       evaluate(λ(t));
       μ(t+1) = select(λ(t) (+ μ(t)));
       t = t + 1;
loop end;
```

Figure 4.1: FCES pseudocode

4.1.1 Cluster Analysis

Cluster analysis is a generic term for a collection of unsupervised algorithms designed to identify structure within data (Bezdek and Pal, 1992). Because of its suitability for a variety of pattern recognition problems, cluster analysis has found extensive use in fields of image processing, data compression, data mining, statistics and natural sciences. More recently, as was seen in chapter three, clustering has also been incorporated into EAs to assist with the optimisation of multimodal search spaces. For a fuller discussion of clustering and a variety of cluster analysis methods, the reader is referred to the work of Kaufman and Rousseeuw (1990); the review in this chapter only considers those cluster analysis methods which are relevant to this work.

Partitioning Data Objects

Given a finite dataset of objects X, the objective for a cluster analysis procedure is to identify k natural subgroups within the set. In objective function-based clustering, this process is itself an optimisation procedure, which seeks to minimise the function:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} \left\| x_i - v_j \right\|^2 \tag{4.1}$$

Where u_{ij} is an element of the partition matrix U, and $\|x_i - v_j\|$ is any norm which acts as distance metric between x_i, the i^{th} data object, and v_j, the centroid of the j^{th} cluster. Frequently, the Euclidian distance metric is adopted as defined by:

$$|\vec{x} - \vec{y}| = (\vec{x} - \vec{y})^T(\vec{x} - \vec{y}) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{4.2}$$

for n-dimensional Euclidean space. The Euclidian norm is employed in all experiments recorded here. When cluster centroids are located at positions that most accurately partition the data, a minimum value for this function is delivered. In other words, when J is small, data objects are positioned close, in Euclidean geometry, to their respective cluster centroids. The partition matrix $U = |u_{ij}|$ is populated with membership values that specify the cluster to which each data object belongs. When hard clustering is employed, the partition matrix contains binary values such that u_{ij} is assigned a value of 1 when x_i is a member of the j^{th} cluster, and 0 otherwise; more formally:

$$u_{ij} = \begin{cases} 1 & when\ x_i \in C_j \\ 0 & when\ x_i \notin C_j \end{cases}, \quad C_j \subset X \tag{4.3}$$

4.1.2 Clustering for Niche Identification

In the context of evolutionary computation, the population constitutes the dataset, and cluster analysis provides a procedure for identifying species. With fixed geographic evolutionary methods (reviewed in section 3.5), a fixed notion of species is imposed upon the population, irrespective of the underlying problem structure. However, clustering enables species to be derived from the distribution of population members within the search space.

4.1.2.1 Fuzzy Clustering

The clustering method employed by Sullivan (2001) within the FCES is the fuzzy c-means technique, which was originally introduced by Bezdek (1973, 1981). Fuzzy clustering incorporates fuzzy set theory (Zadeh, 1965) to the cluster analysis model which, unlike alternative hard-clustering methods, ensures that all data objects have some degree of membership to each cluster. In practice, this is achieved by enabling the partition matrix U to contain real-values in the interval $[0\ ,1]$. The advantage of this approach is that *outliers*, or data points positioned on a cluster boundary, are correctly identified as such.

The fuzzy clustering algorithm operates by iteratively minimising the following function.

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{k} \left(u_{ij}\right)^m \left\| x_i - v_j \right\|^2 \ , \quad 1 \leq m < \infty \tag{4.4}$$

Where u_{ij} is now an element of the fuzzy partition matrix U that specifies the degree to which data object x_i belongs to cluster set C_j. The additional parameter m acts as a weighting exponent, controlling the degree of fuzziness between clusters. For the experimentation presented in this thesis, m is always set to a value of 2, as recommended by Sullivan (2001), based upon an empirical study presented in Xie and Beni (1991).

The process for optimising equation 4.4 can be summarised by the following pseudocode:

```
randomly initialise fuzzy partition matrix
do
{
      calculate cluster centroid locations
      recompute fuzzy partition matrix
}
while(change in partition matrix > tolerance)
```

Figure 4.2: Pseudocode for the fuzzy centroid optimisation procedure

Initially, the fuzzy partition matrix is randomly instantiated according to the condition:

$$\sum_{j=1}^{k} u_{ij} = 1 \ , \quad i = 1 \ \ldots n \tag{4.5}$$

From which, cluster centroids V may be calculated by the equation:

$$v_j = \frac{\sum_{i=1}^{n} \left(u_{ij}\right)^m x_i}{\sum_{i=1}^{n} \left(u_{ij}\right)^m} \quad , \quad j = 1 \ \ldots n \tag{4.6}$$

Elements of the fuzzy partition matrix U may then be recomputed by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left[\frac{\|x_i - v_j\|}{\|x_i - v_k\|}\right]^{\frac{2}{m-1}}} \quad , \quad \begin{matrix} \forall j = 1, \dots, k \\ \forall i = 1, \dots, n \end{matrix} \tag{4.7}$$

These functions are alternately computed until the cluster centroid positions do not significantly change between consecutive iterations. This convergence check is performed using the infinity norm:

$$\|U^t - U^{t-1}\|_\infty = max_{ij}\left|u_{ij}^t - u_{ij}^{t-1}\right| \tag{4.8}$$

In which t represents the current iteration of the clustering procedure. When the infinity norm produces a result less than a threshold set in the range $[0\ ,1]$, the iteration cycle terminates and clustering is complete.

4.1.2.2 Partitioning Population Members Using *C*-Means Fuzzy Clustering

The FCES clusters the parent population of an extinctive $(\mu\ ,\lambda)$ ES to identify any organisation or structure which may be emerging amongst selected population members. The partition information is then accessed during recombination to ensure that offspring are generated by blending genetic material according to the information stored in the partition matrix. This process encourages recombination amongst individuals from within the same niche, and conversely, limits interspecies recombination between members of different niches. This approach is reminiscent of Deb and Goldberg’s (1989) *restricted mating scheme*, which reinforces the formation and preservation of species when fitness sharing is employed (see section 3.4.2).

Two fuzzy-recombination techniques are presented in Sullivan’s thesis:

- ***Fuzzy Discrete Recombination*** - In which offspring are constructed from alleles copied directly from randomly selected parents. This recombination technique is comparable with the probabilistic selection scheme often employed by genetic algorithms; however, each parent’s selection probability is defined by partition information rather than fitness information. Thus, the likelihood of any parent individual being selected to participate in recombination is weighted in proportion to their fuzzy membership.

- ***Fuzzy Intermediate Recombination*** – In which the value at each offspring locus is calculated by the arithmetic average of all corresponding parent alleles, and the contribution of each parent is individually weighted in proportion to their fuzzy membership.

The introduction of fuzzy clustering to the ES addresses one of the main factors identified in section 3.4, which precludes the formation of species within conventional evolutionary algorithms: *Recombination Disruption.* Strong genetic adaptations may be preserved by biasing the recombination operator such that the offspring of a selected cluster are comprised primarily from the genes of parents positioned closest to the cluster centroid. Consequently, the disruptive effects of cross-species recombination are reduced, and the likelihood of global optimisation in multimodal environments is increased (Sullivan, 2001), (Mitchell and Pipe, 2006).

However, the FCES is not a complete niching method as defined by Mahfoud (1995). Upon locating a global optimum within a given problem space, the population will begin to converge towards this single point through the pressure of global selection. However, if the problem space is composed from multiple peaks of equal magnitude it is possible that FCES will locate multiple optima. This is a special case in which FCES can function as a genuine niching algorithm. Sullivan thoroughly investigated the FCES when applied to Himelblau’s function (Sullivan, 2001), a test example which exhibits these search space characteristics.

In order to concurrently preserve and maintain multiple disparate solutions of varying fitness, the model must be augmented in some way. One approach is to observe the remaining factor that prevents the formation of species - *optimistic selection* - and modify the algorithm accordingly. Indeed, this refinement constitutes one of the contributions of this thesis: an extension of Sullivan’s clustering algorithm to enable niching.

4.2 Multiple Solution Clustering Evolution Strategy (CES)

The principle cause of complete population convergence with FCES is the use of global selection when offspring are chosen to act as parents of the subsequent generation. Should a cluster of individuals be positioned at a region of relative high fitness, it is likely that global selection, acting on the entire population, will favour descendants of this cluster over descendants of less well-positioned individuals of other clusters. Subsequent homogenisation of the population will render the clustering phase redundant, as all individuals will occupy the same subspace of the search environment, which may hinder subsequent progress rather than help.

One remedy to this global convergence problem is the use of a resources-sharing mechanism such as *fitness sharing* (Deb and Goldberg, 1989), which *warps* the fitness landscape, forcing individuals within the same niche to share the local resources. An alternative approach is to employ a selective replacement strategy to ensure that offspring only replace individuals from within the same niche, a concept originating from the *crowding* technique (De Jong, 1975).

Both of these approaches have previously been applied to the ES to good effect by Shir and Bäck (2005). However, there is an alternative approach which neatly fits the existing clustering architecture. Inspired by Harik's (1995) *restricted tournament selection* scheme, diversity can be preserved by preventing members of one cluster competing with those of another. This is achieved using a novel cluster-based selection scheme in which the best μ/k offspring are selected as parents from each cluster. Individuals belonging to distinct clusters may then pervade the course of evolution.

4.2.1 Restricted Cluster Selection

Before parental genetic information is recombined to produce offspring, it must first be decided from which cluster each offspring will originate. In the model presented in this thesis, offspring cluster membership is decided by random uniform selection amongst all clusters; however, alternative methods may also be applied. Offspring are then created by fuzzy recombination and traditional ES random mutation. This selection/recombination/ mutation cycle then iterates until a complete population of λ offspring have been successfully bred.

Selection then draws the fittest μ/k offspring from the descendants of each cluster, to act as parents of the subsequent generation. Should the number of offspring produced by any cluster happen to be less than μ/k, additional cluster members are generated by mutating the cluster's fittest individual. This novel selection process is termed *restricted cluster selection*.

Parents are subsequently merged and cluster membership is recomputed. The reapplication of cluster analysis is paramount to the success of this niching algorithm, as it ensures that clusters converging upon the same niche merge to form a single cluster. Consequently, the remaining clusters will then be assigned elsewhere, promoting increased exploration and preventing the entire population from gravitating towards the same peak.

In this respect, all evolutionary operators now consider the cluster membership, and the architecture may be interpreted as a fuzzy-multiple population algorithm in which subpopulations are re-established at each generation to reduce redundancy.

4.2.2 Hard Clustering

The application of fuzzy clustering is actually detrimental to the process of niching within the FCES. If each member of the population has some degree of membership to all clusters, highly fit adaptations are able to diffuse throughout the population, conflicting with the notion of multiple solution niching.

For this reason, the improved niching ES presented in this chapter adopts a hard cluster analysis technique, known as *k*-means, which partitions the parent population such that individuals belong only to one cluster. As the *fuzziness* is removed from the FCES this new architecture is referred to as the clustering ES or CES.

K-Means Clustering

Bezdek's *c*-means fuzzy clustering technique is an extension to a precursory cluster analysis method known as *k*-means (Lloyd, 1982 and MacQueen, 1967). *K*-means analysis seeks to position k cluster centroids such that the value for J (equation 4.1, page 52) is minimised. The process is represented by the following pseudocode:

```
Select data objects as initial cluster centroids
do
{
       assign each data object to the closest cluster centroid
       recompute new cluster centroids
}
while(cluster membership is not stabilised)
```

Figure 4.3: Hard cluster centroid optimisation procedure

Cluster centroids are usually initialised by random selection of objects from the dataset. Each remaining data object is then assigned to the cluster with the nearest (Euclidian distance) centroid. Cluster centroids are then recalculated according to:

$$v_j = \frac{1}{n_{c_j}} \sum_{x_i \epsilon\, c_j} x_i \quad , \quad \forall j = 1, \dots, k \tag{4.9}$$

Data objects are subsequently reassigned to the nearest cluster and centroids are again recomputed. The process iterates until the centroid positions stabilise.

Cluster Initialisation

Despite its wide application, k-means analysis often fails to cluster data objects such that J is optimally minimised. Repeated application of the assignment and centre calculations serve to navigate cluster centroids towards the nearest local minima (Peña *et al*, 1999). K-means analysis is thus very sensitive to the positioning of the initial cluster centroids. This shortcoming has motivated a range of different initialisation techniques, some of which are reviewed and compared by Peña *et al* (1999) and He *et al* (2004)[3].

The furthest point algorithm (Gonzalez, 1985) guarantees a constant factor approximation of two, by compiling a maximally diverse subset of V objects from the data set X. The initialisation begins by selecting a random object as the first centre. The second centre is then chosen as the point that maximises its distance from the first centroid. Subsequent centres are then chosen from the data set that maximise their distance from the nearest centroid already in the set, the so-called *MaxMin* criterion:

$$v_{j+1} = max_i \lfloor min_j |x_i - v_j| \rfloor \tag{4.10}$$

This initialisation approach is also outlined by Katsavounidis *et al* (1994), and found to result in ‘significantly better cluster separation’ (He *et al*, 2004), corroborating results attained by Snarey *et al* (1997). To ensure that population members are accurately grouped into diverse clusters, the k-means cluster centroids in the CES are initialised according to the furthest point algorithm.

[3] Interestingly, EAs have also been employed to compute optimal cluster partitions, see for example (Krishna *et al*, 1999).

4.2.3 New Recombination Operators

The application of a hard cluster analysis renders Sullivan's fuzzy recombination operators incompatible with the CES. Thus, two new recombination operators are proposed that prohibit mating between parents belonging to different clusters:

- ***Hard Discrete Recombination*** - In which offspring are engendered by copying alleles directly from randomly selected parents drawn from within the same cluster. Unlike the fuzzy variant, each cluster member has equal selection probability, and parents belonging to other clusters cannot be selected.
- ***Hard Centroid Recombination*** - This recombination operator takes into account the work on *genetic repair* presented by Beyer (2001) reviewed in section 2.3.2. Beyer has demonstrated that progress rates can be significantly improved by setting the number of parents that partake in recombination as high as possible $(\rho = \mu)$. Intermediate recombination is then the process of assigning each offspring individual to the centroid of the parent population. Within the CES, this procedure is already performed for each niche at the cluster interval. Therefore, hard centroid recombination automatically assigns the offspring of each cluster directly to the position of its parents' cluster centroid, discarding the need for recombination of object parameters entirely. The process of cluster analysis is therefore intimately linked with the recombination operator.

4.2.4 Cluster Quantity – Selecting a Value for *K*

As with other exogenous population parameters μ and λ, the optimal value for k is problem dependant; as such, no single value can be quoted that is appropriate for all classes of problem. When the composition of the problem space is known, it is useful to set k equal to the number of peaks. However, in most real-world applications, *a priori* domain specific knowledge is not available, and practitioners should derive an appropriate value for k by some other means. In his doctoral thesis, Sullivan recommends a minimum cluster cardinality of three for the FCES algorithm (Sullivan, 2001). That is, the parent population size μ should be set to at least $3k$. Alternatively, the cluster quantity k should be set to at most $\mu/3$. One approach is to set k equal to the number of desired independent solutions. However, a more general approach is to adjust k within the range $2 \leq k \leq \mu/3$ and select an appropriate value based upon performance. In this respect, the number of clusters may be determined directly from the parent population size: a value which must be chosen by the practitioner when optimising with conventional ESs.

A future development of this clustering approach would be to use an automatic means to set k, for example the *x*-means cluster analysis method developed by Pelleg and Moore (2000). However, in the synthesis application domain presented within this thesis, a specific number of solutions is desired and thus automatic methods are not required here, but will be developed outside of this thesis

In the following sections, the performance of the CES algorithm is compared with a selection of conventional ES-based algorithms in application to selected benchmark test functions.

4.3 An Analysis of Performance in Selected Test Environments

In the lore of evolutionary computation it has become common practice to assess EA performance by comparative analysis with alternative optimisers, in application to a wide range of benchmark test functions, see for example (Streichert *et al*, 2003). Many theoretical test functions have been devised that exhibit problem characteristics encountered in real engineering problem spaces. However, this approach must be used with caution. If it were possible to compare algorithmic performance within every possible test environment, it would be increasingly difficult to determine which algorithm performs the best; as proposed by the *no free lunch theorem* (Wolpert and Macready, 1997).

The CES is a specialised optimiser which is designed to perform well on problems with certain search space characteristics which are shared by many different real-world problems including the parameter estimation problem analysed later in this thesis. Three performance attributes are thus identified, and relevant test functions are chosen for which algorithmic performance is quantified using standard test metrics. Algorithms are examined in application to a minimum of two test problems for each performance attribute to verify the consistency of the results.

4.3.1 Experimental Introduction

In the first experiments, the robustness of the CES algorithm is explored in application to a variety of test problems. There are three sections, each focusing on different optimisation attributes:

- ***Global*** **-** the ability of the algorithm to locate the global optimum within environments composed from many local sub-optima. This attribute is entitled *Multimodal proficiency*.
- ***Multiple*** **-** the ability of the algorithm to locate and maintain multiple distinct solutions within multimodal environments. This attribute is entitled *Multiple solution proficiency.*
- ***Multidimensional*** - the ability of the algorithm to find and maintain multiple high fitness solutions in problems of high-dimensionality. This attribute is entitled *Multidimensional proficiency.*

While these attributes do not encapsulate every problem characteristic, they represent characteristics which are pertinent to many real-world problems including the sound synthesis application examined later in this thesis. The FM matching application presents a problem space that is highly complex and multimodal (Horner, 1997), the chosen optimiser is required to be able to locate a highly optimal solution; hence the identification of test attribute 1. However, the delivery of a variety of problem solutions is also of interest here; hence the inclusion of attribute 2. The final attribute is important as synthesisers present high-dimensional search spaces, within which the EA must scale well.

4.3.1.1 Experimental Set-Up

For the empirical results that follow, the performance of four ES-based algorithms are compared with the novel CES niching algorithm, proposed in the previous sections of this chapter. A brief summary of each algorithm is outlined below.

Algorithm Selection

$(\boldsymbol{\mu} \overset{+}{,} \boldsymbol{\lambda})$ ***Evolution Strategy (ES)***

The traditional ES as defined by Schwefel (1995), Beyer (2001) and Bäck (1996), see section 2.3.2 of this thesis.

Multi Start $(\mathbf{1+1})$ ***Evolution Strategy (MSES)***

A variant of the basic two-membered (1+1) ES as defined originally by Rechenberg (1973). Multiple instances of the algorithm are evolved concurrently; this algorithm is also referred to as a multi-start hill-climber (Streichert *et al*, 2003). Each (1+1) ES mutates object parameters isotropically according to a single mutation step-size, which is adapted by the 1/5th rule (Schwefel, 1995, see section 2.3.2.3).

Cooperative Coevolution Strategy (CCES)

An implementation of Potter and De Jong's (1994) cooperative coevolutionary algorithm, in which a conventional ES is employed as the underlying EA. This algorithm is consistent with the model described in section 3.7.

Fuzzy Clustering Evolution Strategy (FCES)

Sullivan's original EA employing fuzzy cluster analysis to partition the dataset, as described earlier in this chapter. Numerous variants of this algorithm are included in the subsequent experimentation, details of which are outlined in the next section.

Clustering Evolution Strategy (CES).

An implementation of the novel niching algorithm based upon FCES, proposed throughout the preceding sections of this chapter. Several variants of the algorithm are implemented, details of which will also be provided in the following section.

4.3.1.2 Algorithm Structure and Parameters

To ease accurate analysis of, and comparison between each algorithm, consistent parametric constraints are imposed across all experiments. Indicated results are produced from the mean average of 30 runs for each algorithm when applied to each problem. Performance differences are then discussed with claims backed by a statistical comparison of population means. The population is randomly initialised anew for each run and, where possible, the same random data set is used to initialise each algorithm, enabling observed performance differences to be attributed to each algorithm's interaction with the same initial perspective of the problem space.

Many of the test functions employed in the subsequent analyses are maximisation problems; however, all of the ESs included here are object function minimisers. To ensure appropriate optimisation, each maximisation problem is simply adapted by negating the result produced by each test function accordingly.

Runtime

Experiments are executed for 50 generational cycles, except under circumstances in which comparison by generations does not provide a meaningful indication of optimisation time. For example, in cooperative coevolutionary algorithms, the term *generation* corresponds to the advancement of a single sub-population by one generation. Of more interest here is the period in which all sub-populations have advanced by one generation, referred to as a *round*, according to (Jansen and Wiegand, 2004). In these instances a fitness evaluation counter is used to provide a meaningful comparison.

Evolutionary Operators

Mutation

To ensure a fair comparative study, the same evolutionary operators are adopted for all algorithms. Mutation is provided by the derandomised mutation operator defined by Ostermeier *et al* (1994), with the exception of the MSES as described previously.

Recombination

Where relevant, EAs are tested using both discrete and intermediate recombination operators with all parents participating in the production of offspring, i.e., $\rho = \mu$. The obvious exception to this rule is the CES, in which parents are chosen from within each cluster according to the selection and recombination operators described earlier.

In applying the cooperative coevolutionary architecture, each test function is naturally decomposed into n subcomponents (Potter and De Jong, 1994), where n represents the dimensionality of the problem space. A separate subpopulation is then assigned to each component such that each individual represents a single object parameter of the greater problem. Discrete recombination of a single parameter is thus equivalent to no recombination, as genes are passed directly to descendants without variation.

In the experiments presented throughout the remainder of this thesis, the selection pressure is maintained at a fixed ratio of $\mu/\lambda = 1/7$ where applicable, as indicated to be optimal by Schwefel (1987). Population sizes vary in this proportion with exact figures indicated for each test case. For algorithms that employ cluster analysis, the cluster *cardinality* is set to five, such that $c = \mu/5$; Sullivan recommends a minimum cardinality of three (Sullivan, 2001). Results are provided for both '$+$' and '$,$' (*plus* and *comma*) strategies for each algorithm, where Sullivan's original FCES only employed extinctive '$,$' selection, which is widely accepted to be the superior selection strategy for the self-adaptive mutation mechanism (Schwefel, 1995), (Bäck *et al*, 1993).

Algorithmic Variation

Summaries of the algorithmic variations for each algorithm are provided below.

CES and FCES – Results are presented with both recombination types for each algorithm: discrete and centroid recombination; and discrete and intermediate recombination respectively. Sullivan’s global selection operator is also compared, in collaboration with the *k*-means clustering, as well as the presented *restricted cluster selection* to provide fair comparison between the cluster analysis techniques.

CCES – Results with both the *single-best,* and *single-best plus one random* credit assignment procedures are presented, as described in section 3.7, which are referred to as CCES1 and CCES2 respectively. Both intermediate and discrete recombination operators are tested.

4.3.2 Attribute 1: *Global* Multimodal Proficiency

In this section the ability of each evolutionary algorithm to locate the global optimum within three selected multi-modal environments is assessed. The benchmark test problems have been selected as they each exhibit landscape characteristics which are of relevance to the final FM application domain. The selected test functions are the *multimodal* problem, *Langermann’s* function and the *Maximum of Two Quadratics* problem.

Performance Criteria

In the following experiments, each algorithm is applied to two-dimensional implementations of each problem, recording the proportion of 30 runs in which the optimum peak is successfully located.

Each algorithm maintains an offspring population size of 140. For the MSES this corresponds to the concurrent execution of 140 $(1+1)$ ESs. For the multi-membered ESs, 20 parents are selected at each generation; $(20 \overset{+}{,} 140)$ in ES notation. In the clustering EAs, the parent population is partitioned into four clusters, such that five individuals are allocated to each cluster.

4.3.2.1 Experiments on the Multimodal Test Function

The multimodal function is defined in *n* dimensions by:

$$f(\vec{x}) = \prod_{i=1}^{n} sin^6(5.1\pi x_i + 0.5) \cdot e^{\left(-4\left(\frac{x_i-0.0667}{0.8}\right)^2\right)} \quad , \quad x_i \in [0\,,1]\forall i \qquad (4.11)$$

This test environment was originally introduced by Goldberg and Richardson (1987) for testing the effectiveness of their fitness sharing operator, and has since become a popular choice for the investigation of niching evolutionary algorithms, see for example the work of Shir and Bäck (2005) and Hocaoglu and Sanderson (1997). The problem is comprised of a sinusoid, shaped within the envelope of a decaying exponential. Parameter ranges are restricted to the interval $\vec{x} \in [0\,,1]^n$ in which there are 5^n equally spaced peaks, with one optimum located at $x_i = 0.0683\ \forall i.$ A landscape and contour plot of equation 4.3 is provided in figure 4.4.

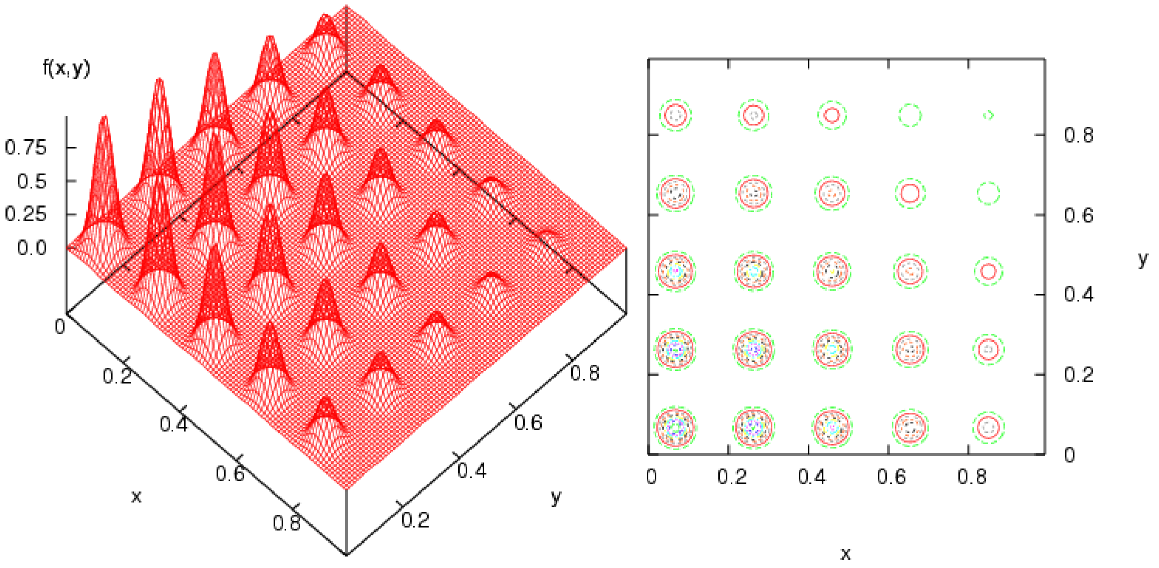


Figure 4.4: Multimodal landscape and contour plot

The results from each experiment are provided in figure 4.5**,** where the y axis of each chart provides the algorithm type, in the format:

(strategy type) / algorithm type / recombination type / selection scope

If the selection scope is not indicated, global selection is assumed. The x-axis provides the number of successful runs in which the global optimum is located.

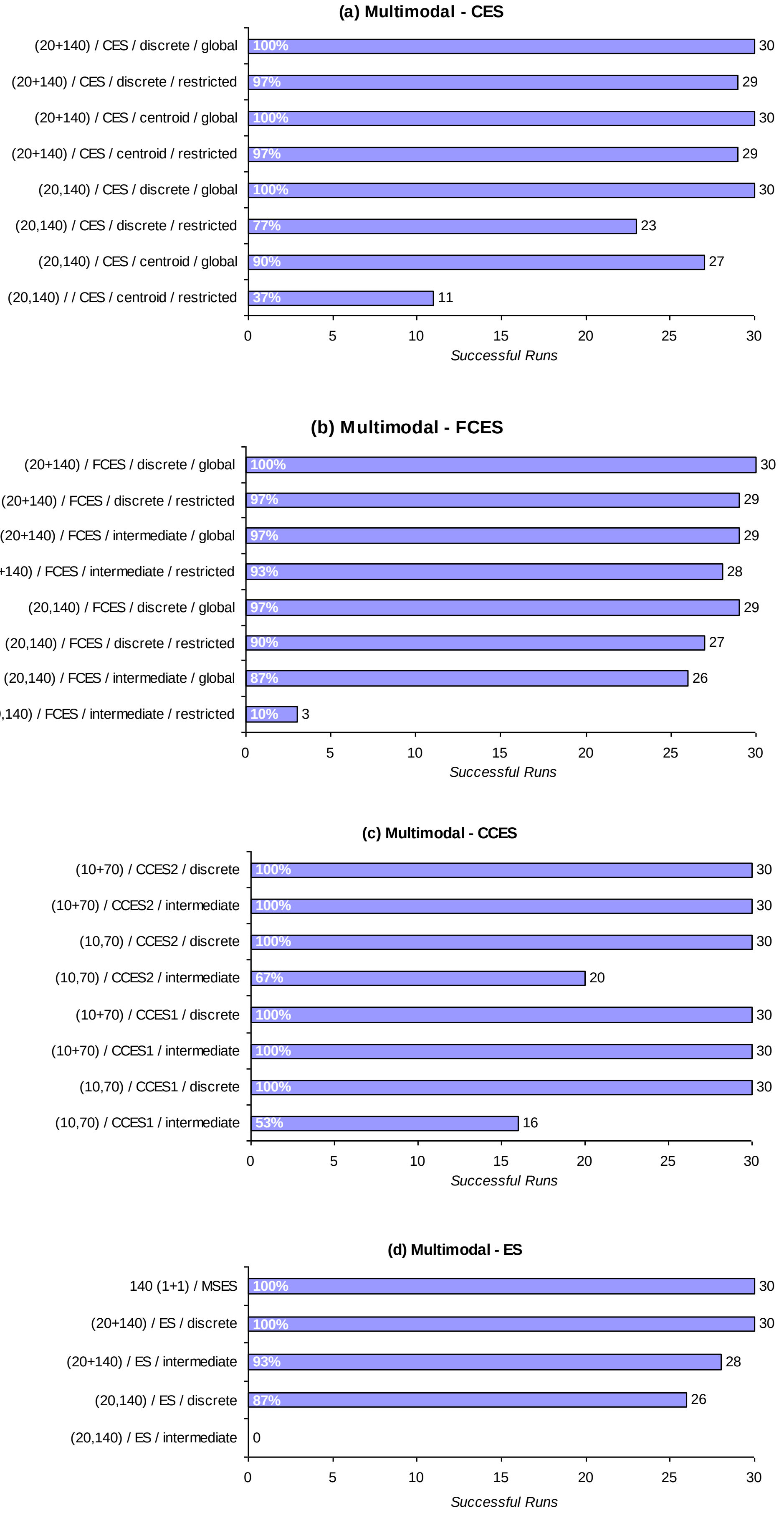


Figure 4.5: Results from experiments with the multimodal function

Discussion

The canonical ES variants tested here have been found to perform well within the two-dimensional multimodal problem; with the exception of the intermediate recombination variant with extinctive selection. By arithmetically averaging the entire parent population, offspring produced by intermediate recombination are biased towards the centre of the search space. Within the multimodal problem, this characteristic tends to guide the population away from the global optimum. The notable improvement in results when using intermediate recombination for the FCES and, better still, centroid recombination for the CES, confirm that this pathology is significantly ameliorated when clustering is incorporated into the ES architecture.

It is clear that, in terms of optimal performance, there is a general improvement in the CES over that of the FCES for the multimodal function. This conclusion is evidenced by the five out of the eight test cases in which the hard clustering strategies outperform the fuzzy clustering strategies. The only test case in which the CES does not outperform the FCES is when discrete recombination with extinctive selection is employed.

When the proposed *restricted cluster selection* (labelled ‘restricted’) is used by the cluster-based ESs, the likelihood of finding the optimum is also reduced. It may be that there is a trade-off between the quality of the best solution and the number of final solutions located. This may be because cluster-based selection intentionally preserves less well adapted offspring to retain diversity, which may restrict performance in terms of global optimisation.

The CCES algorithm also performs well with discrete recombination, and also when the intermediate recombination operator is used in conjunction with preservative selection. The multimodal problem shares many characteristics with Schwefel’s function in terms of modality, separability and optimal positioning, a test function to which Potter has previously applied the cooperative coevolutionary model with great success (Potter and De Jong, 1994).

4.3.2.2 Experiments on Langermann’s Function

Langermann’s function is defined by the following equation:

$$f(\vec{x}) = -\sum_{i=1}^{n} c_i \left[e^{\left(\frac{-1}{\pi}\|\bar{x}-A(i)\|^2\right)} \cdot cos(\pi\|\bar{x}-A(i)\|^2) \right] \quad , \quad x_i \epsilon [0\,,10] \forall i \qquad (4.12)$$

where the matrices c and A can be found in the work of Bersini et al (1996). This test function provides a multimodal search space which, unlike the multimodal function, is irregular, non-separable and highly rugged. There are numerous local sub-optima and only one optimum. Langermann’s function presents a search space that better characterises the search space of real-world problems: unevenly distributed local optima, a large flat noisy plain, and an optimum positioned towards the extremity of the fitness landscape – $(x_1\,, x_2) = (8.0655\,, 9.0382)$ for two dimensions. A landscape and contour plot is provided for the Langermann function in figure 4.6.

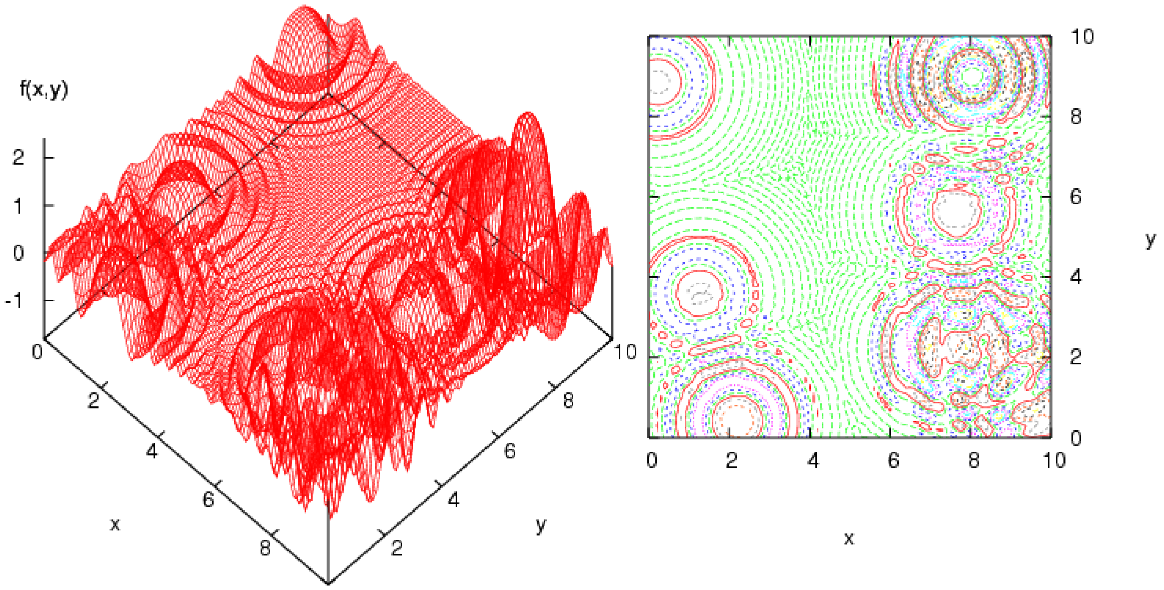


Figure 4.6: Langerman’s function with contour plot

In the following experiments, the experimental procedure adopted for the preceding multimodal problem is reapplied with identical parameter settings and population sizes.

Results

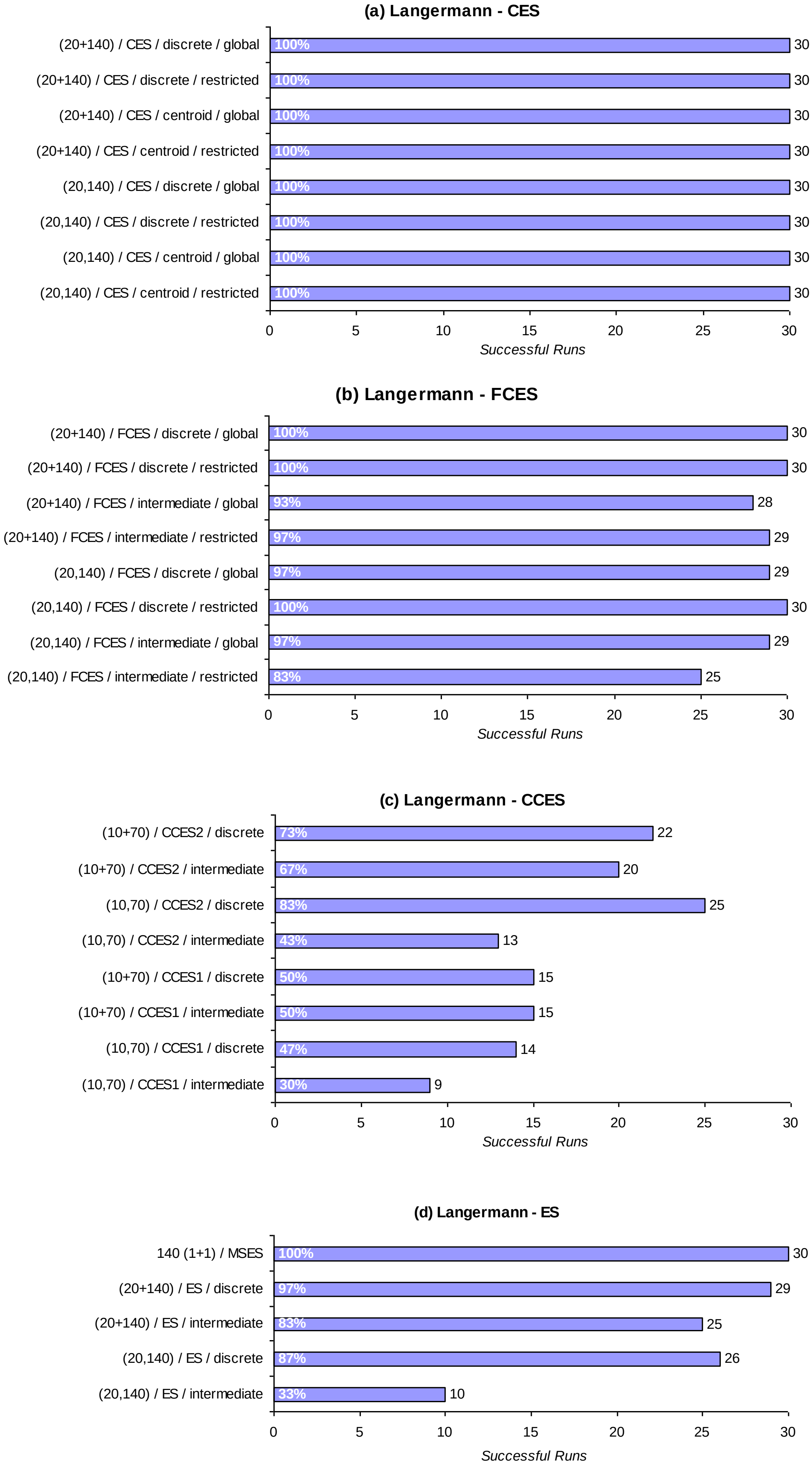


Figure 4.7: Results from experiments with Langermann's function

Discussion

In all test cases in which the CES is applied to Langerman's function the optimum is successfully located. The performance of the FCES is also good; however, there is a small degradation in performance when compared with the CES. This differential is most notable between the intermediate recombination for FCES, and centroid recombination for the CES, with the latter shown to be consistently superior.

A small yet consistent advantage may also be observed when using the *restricted cluster selection* method for the FCES, rather than the more standard global selection operator. As cluster-based selection facilitates the preservation of species, the population is not driven to converge towards a single optimum. In turn, this leads to increased exploration, which may account for the improvement in performance in this instance.

Compared with the experiments on the multimodal function in the previous section, the CCES is shown to be significantly less successful at locating the global optimum within this environment. While cooperative coevolutionary algorithms are particularly efficient at identifying the global optimum in certain problems, they are quite poor in others. This behaviour has been thoroughly analysed by Wiegand *et al* (2001), and further details of his work will be provided later in this thesis. In fact, next test function to be explored was designed by Wiegand to illustrate a tendency for cooperative coevolutionary algorithms to optimise parameters that are relatively robust to change in other parameters, rather than parameters that are globally optimal.

4.3.2.3 Experiments on the Maximum of Two Quadratics Function

This section considers the Maximum of Two Quadratics function (MTQ) which was defined originally by Wiegand *et al* (2002) to test the performance of cooperative coevolutionary algorithms. The MTQ function presents a two-dimensional bimodal deceptive problem constructed from two quadratic functions defined as follows:

$$f(x\,,y) = max\begin{cases} H_1\left[1 - \dfrac{16(x-X_1)^2}{S_1} - \dfrac{16(y-Y_1)^2}{S_1}\right] \\ H_2\left[1 - \dfrac{16(x-X_2)^2}{S_2} - \dfrac{16(y-Y_2)^2}{S_2}\right] \end{cases} \quad , \;\; x\,,y \;\in\; [0\,,1] \qquad (4.12)$$

where the parameters H_i, X_i, S_i, and Y_i control the height, width and vertex of each function. The parameters for the first (suboptimal) and second (optimal) peak are set according to the values provided in table 4.1; the same values are employed in the work of Panait *et al* (2004) and Bucci and Pollack (2005). With these settings, the two peaks are positioned at near opposite corners of the search space. The narrow optimal peak is positioned at a low-performing region of the broader suboptimal peak. Consequently, search algorithms may be deceived by the large basin of attraction, which may draw search points away from the global optimum.

Parameter	***Value***
H_1	50
H_2	150
S_1	1.6
S_2	1/32
(X_1, Y_1)	(3/4 ,3/4)
(X_2, Y_2)	(1/4 ,1/4)

Table 4.1: MTQ function parameters

These values produce the landscape and contours shown in figure 4.8.

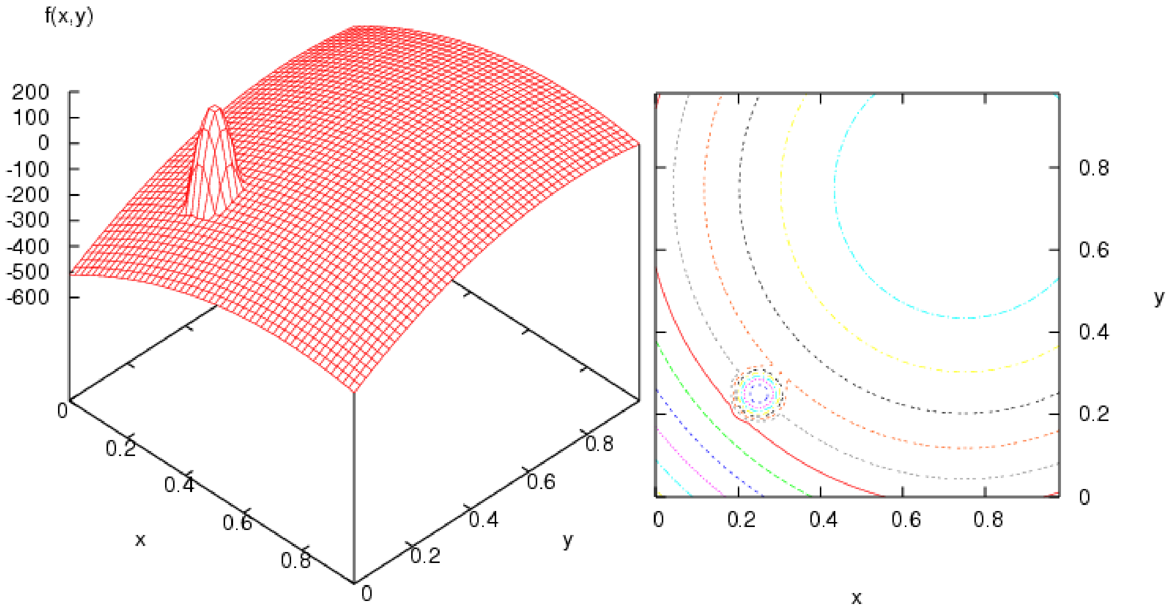


Figure 4.8: Maximum of two quadratics function with contour plot

In the experiments that follow, each algorithm is applied with the experimental procedure employed in the previous two problems, and the results are provided in figure 4.9.

Results

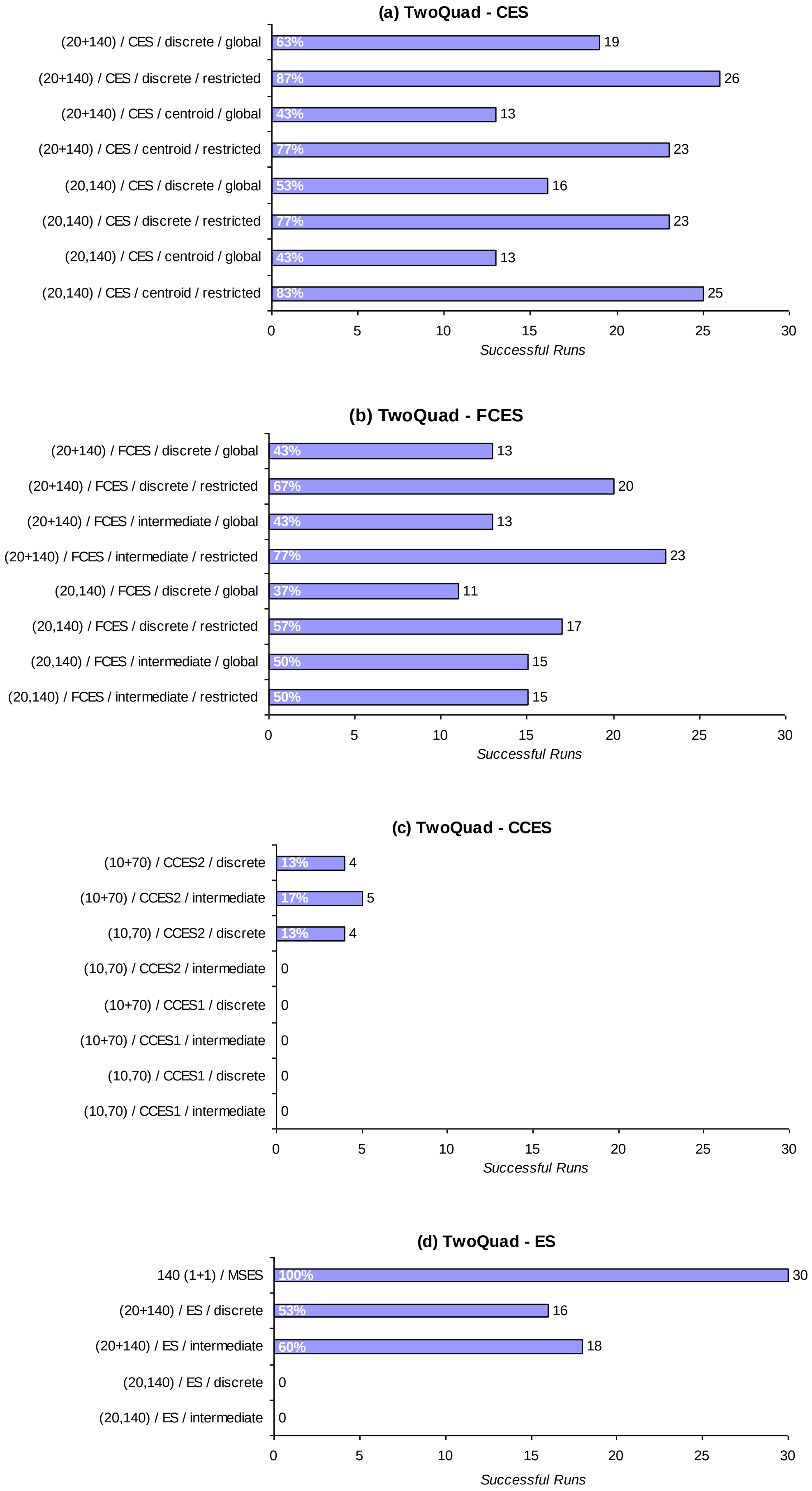


Figure 4.9: Results from experiments with Maximum of Two Quadratics function

Discussion

The results acquired from experiments performed when all algorithms are applied to optimise the MTQ test environment provide further empirical evidence to confirm that, within certain multimodal environments, the CES with hard clustering and *restricted cluster selection* exhibits superior performance when compared with the FCES with global selection. As fuzzy cluster analysis enables all individuals to participate in the recombination process, genes of individuals positioned at large regions of high fitness (the suboptimal peak in this instance) are able to diffuse into neighbouring subpopulations. With such a large basin of attraction drawing the population toward the sub-optimal peak, individuals located around optimal peak may be lost.

Restricted cluster selection also provides enhanced performance, as solutions positioned near to the optimum compete only with their surrounding cluster members, and not with the entire population. Moreover, in all successful runs in which the optimum is located, the restricted cluster selection operator ensures that individuals located at the sub-optimum are also maintained. The location of multiple optima is a performance attribute which is examined in the next section of this chapter.

The Maximum of Two Quadratics function was introduced by Wiegand *et al* (2002) to demonstrate one of the main pathologies that restrict the performance of CCEAs when applied to parameter optimisation problems. CCEAs do not optimise components that are optimal in terms of fitness; rather, they optimise components that are most resilient to change in other parameters, a behaviour which has been termed *relative overgeneralisation*. Wiegand’s work is reviewed in more detail later in this thesis, when cluster analysis is included within the cooperative coevolutionary framework to reduce the effects of relative overgeneralisation, and therefore improve the parameter optimisation performance of the algorithm.

In this test function, and also the multimodal and Langermann functions before, flawless global optimisation by the MSES is observed. This result is unsurprising, as the MSES included in these experiments is equivalent to running 140 hill climbers simultaneously. In each test run it is inevitable that one of the initial random seeds will be sown within the basin of the optimal peak. For the tested problems, it is clear that the MSES is the highest-performing algorithm, however, this performance advantage may not extrapolate well to other test attributes or even measures. As will be shown in the following section, the convergence plots for the MSES indicate slower progress rates than the other algorithms.

4.3.2.4 Convergence Dynamics

In addition to recording the number of successful runs for each algorithm, it is also useful to identify the rate at which optimal solutions are delivered. In the synthesis-matching problem application domain that motivates this work, finding multiple solutions to the problem in the least number of fitness evaluations is imperative. To facilitate comparison between the progress rates of each algorithm, the fitness of the best solution at each generation is plotted for each of the test functions explored in this section. For the purposes of brevity, only one example of each algorithm is plotted. Details of the algorithms and their operators are shown in table 4.2.

Algorithm Variant	***Recombination Type***	***Selection Type***	***Selection Scope***
CCES1	*discrete*	*elitist ('+')*	*global*
CCES2	*discrete*	*elitist ('+')*	*global*
CES	*discrete*	*extinctive (',')*	*restricted*
ES	*discrete*	*extinctive (',')*	*global*
FCES	*discrete*	*extinctive (',')*	*global*
MSES	*none*	*elitist ('+')*	-

Table 4.2: Algorithmic variations for convergence comparison

Each figure displays the convergence velocities for the algorithms listed in table 4.2 when applied to the multimodal (figure 4.10), Langermann (figure 4.11), and MTQ (figure 4.12) functions. Each curve is generated from the average trajectories of five successful runs, except when five successful runs were not achieved, in which case the convergence of the five best runs are averaged.

The results from multimodal and Langermann experimentation demonstrate the efficiency of CCES1, as it converges to the function minimum in the least number of fitness evaluations. The convergence velocity of the CES is relatively low when compared with other algorithms. This may be attributed to the reduction in selection pressure due to the maintenance of diversity through the *restricted cluster selection* operator which allocates population members uniformly between identified clusters. For all test cases in which a global selection scheme is adopted, the population naturally gravitates towards the region in which the highest-performing individuals are located. As an increasing number of trials are allocated to the niche of the global optimum, the progress rate is increased. Of all the optimisers, the MSES is found to be the most robust; however, in these examples it is also the least efficient, converging more slowly than all other algorithms. This is due to the absence of recombination. The MSES is consistently able to locate the global optimum in these experiments as the quantity of parallel $(1 + 1)$ ESs is more than adequate to ensure that at least one ES is seeded within the gradient of each peak. It is worth noting that, in section 4.3.4, the EAs are tested in an environment in which the search space is so vast that this initialisation condition cannot be guaranteed.

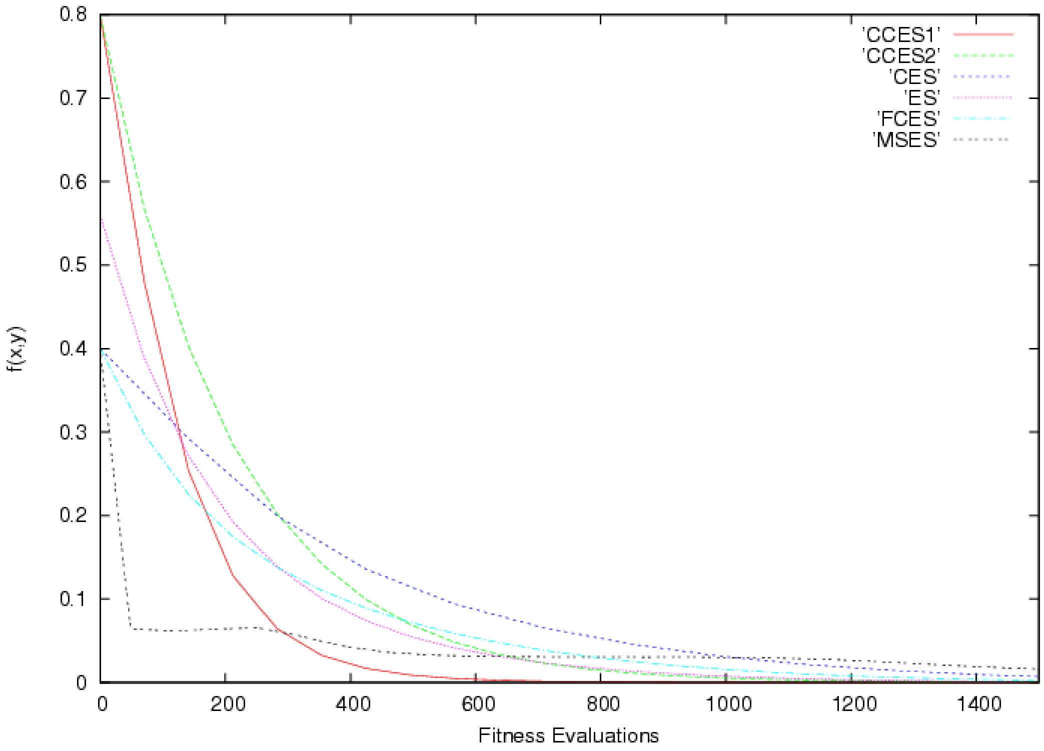


Figure 4.10: Multimodal function convergence dynamics

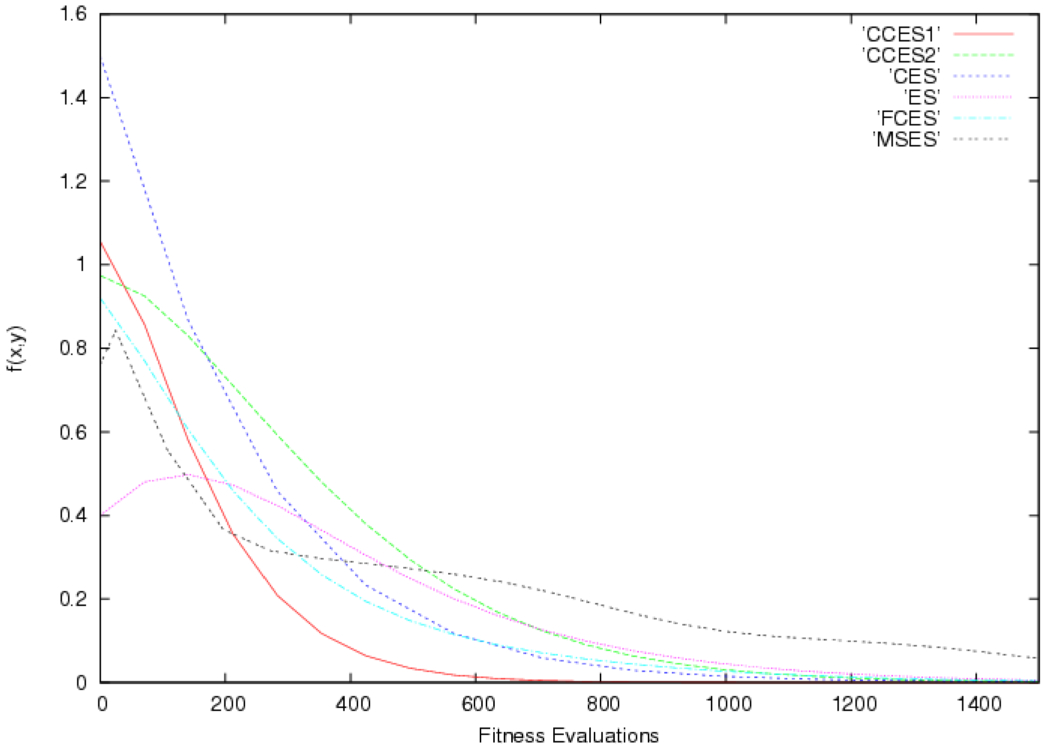


Figure 4.11: Langermann's function convergence dynamics

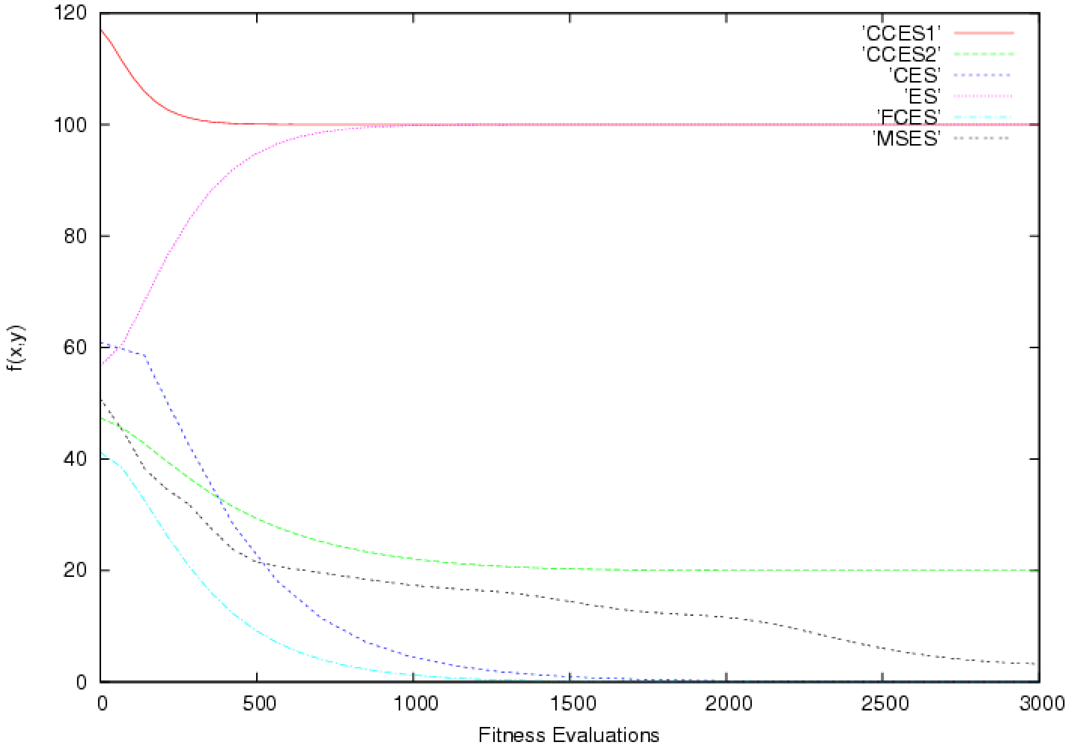


Figure 4.12: Maximum of Two Quadratics function convergence dynamics

4.3.3 Attribute 2: *Multiple* Solution Proficiency

In this section the ability of the proposed CES algorithm to locate and maintain multiple distinct solutions is assessed in application to three multimodal test environments: *Himmelblau's* function, the *Multimodal* function and the *Waves* function. As the ES and CCES are not capable of maintaining multiple distinct search space solutions, they are excluded from these experiments, and results are compared with only the FCES. Experimentation with the MSES is not included here as, in terms of multiple solution maintenance, the multimodal test functions employed in this section present a trivial problem for such an algorithm. The $(1+1)$ ES exhibits behaviour which is similar to that of a hill climber: it will converge to optimum of the nearest peak. If parity were maintained between tested algorithms, such that the MSES executes an equivalent number of concurrent $(1+1)$ ESs as there are offspring in the multimembered ESs, the number of hill climbers would considerably outnumber the number of optima within each test environment. Consequently, the likelihood of a strategy not being seeded within the basin of each peak is very small. In this respect the MSES will elicit flawless multiple solution performance in all test problems investigated here. It is worth noting that comparison with the MSES is resumed in section 4.3.3, where performance is examined within search space environments in which the number of peaks significantly outnumbers the individuals in the ES population.

Performance Criteria

In assessing the performance of EA-based niching algorithms, a great deal of research focuses on drawing a comparison between the distribution of the final population set, and the ideal fitness-proportionate distribution, which may be derived per objective function prior to experimentation. The metric frequently adopted for this purpose is the *chi-square-like performance statistic*, as originally defined by Deb and Goldberg (1989). This metric is ideal for measuring fitness sharing-based EAs, as it reflects the procedure by which the development of niche is promoted by fitness sharing: by allocating population members to each subspace in proportion to the available resources (fitness) (see section 3.4.2). Focus is placed on the ability of each algorithm to successfully identify multiple search space optima, with little interest in the actual distribution of solutions in the final population set. Moreover, the *restricted cluster selection* operators, proposed here, serve to distribute solutions uniformly amongst clusters, irrespective of the topology of the search space. Therefore, an alternative performance metric is employed here known as the *maximum peak ratio,* previously applied to assess the performance of niching algorithms, originally for the GA (Miller and Shaw, 1996), and later for the ES (Shir and Bäck, 2005). This performance measure quantifies both the number and quality of optima in the final population set, and is defined by:

$$Maximum\ Peak\ Ratio\ (MPR) = \frac{\sum_{i=1}^{q_f} f_i}{\sum_{i=1}^{q_F} F_i} \tag{4.13}$$

where $\vec{f}$ is a vector containing the fitness of the q_f optima, represented by the final population set, and $\vec{F}$ is the fitness of the q_F actual optima within the search space. In the results provided below the MPR is averaged over 30 runs.

Furthermore, the performance for each test case is evaluated with three additional measures adopted previously by Shir and Bäck (2005) and Ursem (1999):

- the *Global* optimum location performance: the number of runs, in which the optimum is successfully located.
- the *Total* optima location performance: the number of runs in which all q_F optima are successfully identified and maintained.
- the *Optima* location average: the actual number of separate optima located, averaged over all separate runs, with respect to the total number of peaks q_F.

4.3.3.1 Experiments on Himmelblau's Function

In the first set of experiments, performance is assessed when the CES and FCES are applied to optimise the modified Himmelblau's function, originally presented by Deb and Goldberg (1989) as an augmentation of Himmelblau's original function (Himmelblau, 1972). The two-dimensional problem space is expressed by the equation:

$$f(x,y) = 200 - (x^2 - y - 11)^2 - (x + y^2 - 7)^2 \quad , \quad x, y \in [-5, 5] \qquad (4.14)$$

The search space contains four peaks of equal magnitude as depicted in figure 4.13. While being a relatively simple problem to optimise, concurrent maintenance of all four optima can pose a challenge to evolutionary niching algorithms.

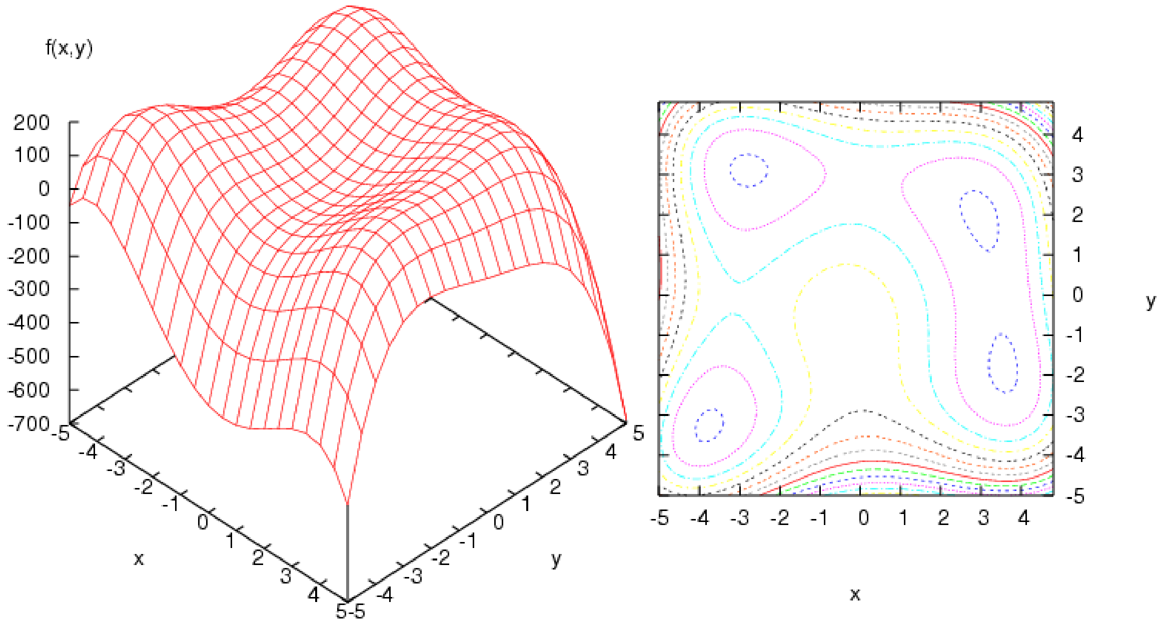


Figure 4.13: Himmelblau's function landscape and contour plot

The experimental procedure presented in the previous section is repeated, with offspring population sizes set to 140, parent sizes to 20 and cluster quantity set to five. Each algorithm runs for 50 generations, and results are drawn from statistical analysis of the final population set averaged over 30 runs. Results of the simulation are provided in table 4.3 and plotted in figure 4.14. Details of each algorithm are presented in the format:

(strat)	*recomb*	*sel*

in which *strat* refers to the strategy type indicating the population sizes and selection mechanism in traditional $(\mu \dagger \lambda)$ ES notation; *recomb* indicates whether intermediate or discrete recombination is employed; and *sel* signifies the scope of the selection operator (global or restricted, where restricted is only applicable to the clustering-based algorithms.

Results

					Himmelblau's Function					
		Algorithmic Parameters					Optima out of 4		MPR	
Algorithm		*(strat)*	*recomb*	*sel*	Global out of 30	Total out of 30	mean	σ	mean	σ
FCES	1	*(200,1400)*	*Discrete*	*Restricted*	30	25	3.83	0.38	0.96	0.09
CES	2	*(200,1400)*	*Discrete*	*Restricted*	30	30	4.00	0.00	1.00	0.00
FCES	3	*(200,1400)*	*Discrete*	*Global*	30	2	2.63	0.72	0.66	0.18
CES	4	*(200,1400)*	*Discrete*	*Global*	30	1	2.50	0.68	0.63	0.17
FCES	5	*(200,1400)*	*Intermediate*	*Restricted*	30	25	3.83	0.38	0.96	0.09
CES	6	*(200,1400)*	*Centroid*	*Restricted*	30	27	3.90	0.31	0.98	0.08
FCES	7	*(200,1400)*	*Intermediate*	*Global*	30	0	2.57	0.50	0.64	0.13
CES	8	*(200,1400)*	*Centroid*	*Global*	30	4	2.93	0.58	0.73	0.15
FCES	9	*(200+1400)*	*Discrete*	*Restricted*	30	28	3.93	0.25	0.98	0.06
CES	10	*(200+1400)*	*Discrete*	*Restricted*	30	27	3.90	0.31	0.98	0.08
FCES	11	*(200+1400)*	*Discrete*	*Global*	30	0	1.53	0.57	0.38	0.14
CES	12	*(200+1400)*	*Discrete*	*Global*	30	0	1.33	0.48	0.33	0.12
FCES	13	*(200+1400)*	*Intermediate*	*Restricted*	30	25	3.83	0.38	0.96	0.09
CES	14	*(200+1400)*	*Centroid*	*Restricted*	30	28	3.93	0.25	0.98	0.06
FCES	15	*(200+1400)*	*Intermediate*	*Global*	30	0	1.93	0.69	0.48	0.17
CES	16	*(200+1400)*	*Centroid*	*Global*	30	0	1.73	0.52	0.43	0.13

Table 4.3: Multiple solution results of experiments on Himmelblau's function

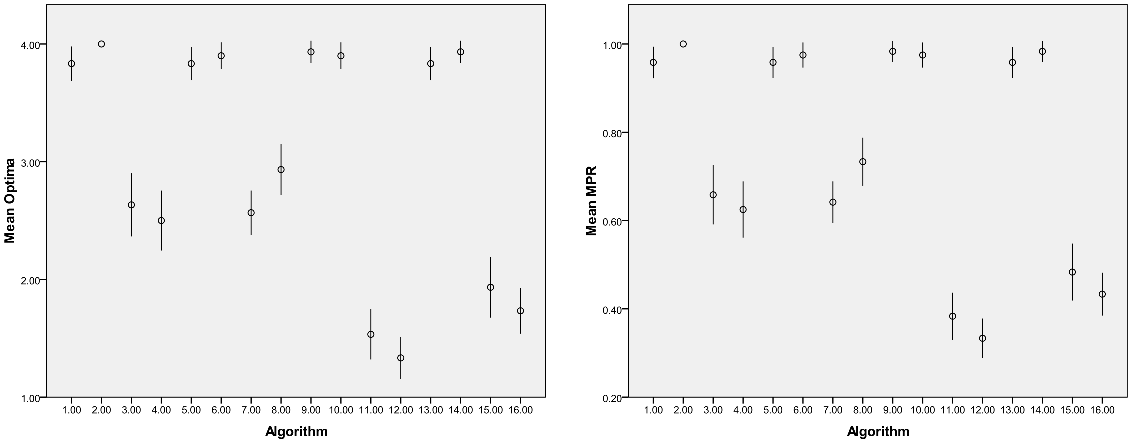


Figure 4.14: Mean and 95% confidence intervals for Optima and MPR results on Himmelblau's function

Discussion

From the results provided above, the superiority in terms of niche maintenance of the *restricted cluster selection* operator is clear. The t-tests for equality of means revealed that the restricted selection operator offered a statistically significant (two-tailed) improvement over the global selection operator in both the mean Optima and mean MPR results. The best results overall were obtained by the CES when restricted selection was employed. Furthermore, in most test cases the CES was shown to outperform the FCES when the restricted selection operator was used, and the reverse was evident when global selection was employed. However, this trend was not statistically significant across all test cases. These results suggest that the CES is better equipped to maintain multiple optima in conjunction with the restricted selection operator than the FCES.

4.3.3.2 Experiments on the Multimodal Function

In this round of experimentation, the niching algorithms are reapplied to optimise the multimodal function, details of which have already been provided in section 4.3.1. The testing method outlined above is repeated here; however, population sizes are scaled in accordance with the landscape topology. In the interval $x\,, y \ \in [0\,, 1]$ the search space contains 25 peaks; thus, the parent population is partitioned into 25 clusters. With five parents per cluster and a selection pressure of $\mu/\lambda = 1/7$, this leads to the exogenous strategy parameters $(125 \overset{+}{,} 875)$. All algorithm derivatives that employ global selection are excluded from this experimentation as total population convergence would take place as a result of the single search space optimum.

As the optimisers minimise, the fitness value produced by the algorithms is negated, as described for the earlier experiments. However, for calculating the MPR there is an additional factor which should be considered. In the multimodal function, fitness results can vary in the range $[0\,, 1]$. As the MPR ratio is weighted to represent the quantity of located peaks in terms of their respective magnitudes, the fitness is modified such that the true fitness of a peak is subtracted from one, to ensure that the fitness of the highest peak contributes the most to the MPR, as should be the case. Results of the multimodal simulation are provided in table 4.4 and plotted in figure 4.15.

				Multimodal Function					
		Algorithmic Parameters				Optima out of 25		MPR	
Algorithm		*(strat)*	*recombination type*	Global out of 30	Total out of 30	mean	σ	mean	σ
FCES	1	*(125, 875)*	*Discrete*	30	0	15.33	1.27	0.80	0.03
CES	2	*(125, 875)*	*Discrete*	30	27	24.87	0.43	1.00	0.01
FCES	3	*(125, 875)*	*Intermediate*	30	0	15.87	1.83	0.79	0.06
CES	4	*(125, 875)*	*Centroid*	30	28	24.93	0.25	1.00	0.00
FCES	5	*(125+ 875)*	*Discrete*	30	0	16.83	2.10	0.84	0.05
CES	6	*(125+ 875)*	*Discrete*	30	24	24.77	0.50	1.00	0.01
FCES	7	*(125+ 875)*	*Intermediate*	30	0	18.53	1.46	0.88	0.03
CES	8	*(125+ 875)*	*Centroid*	30	22	24.70	0.53	1.00	0.01

Table 4.4: Multiple solution results of experiments on the multimodal function

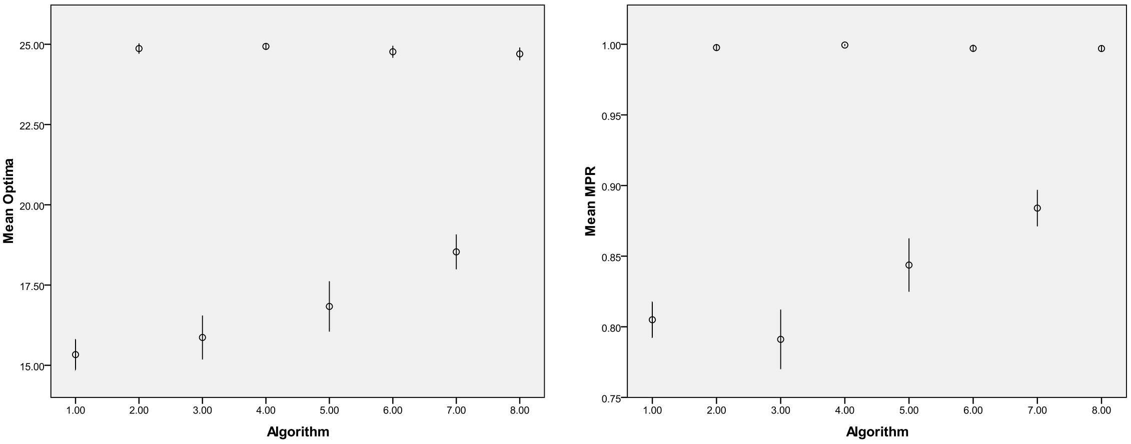


Figure 4.15: Mean and 95% confidence intervals for Optima and MPR results on the multimodal function

Discussion

An equality of means t-test confirms a statistically significant (two-tail t-test) advantage of the CES with *k*-means clustering initialised with *MaxMin* selection, over the Fuzzy *c*-means clustering, for the identification and maintenance of species. With the CES maintaining on average 8.2 optima and scoring 1.6 more than the FCES for the Optima and MPR measures respectively. The high MPR results achieved by the CES indicate that those few peaks that were not located are of very low magnitude.

4.3.3.3 Experiments on the Waves Function

The final test function, on which the cluster-based niching algorithms are assessed for multiple solution proficiency, is known as the Waves function, previously employed by Ursem (1999) and Streichert *et al* (2003). This is defined in two dimensions as:

$$f(x\,,y) = 0.3x^2 - (y^2 - 4.5y^2)xy - 4.7cos\big(3x - y^2(2+x)\big)sin(2.5\pi x)\ , \quad x \in [-0.9\,,1.2]\ ,\quad y\ \in [-1.2\,,1.2] \qquad (4.15)$$

This asymmetric test function features 10 unevenly distributed peaks, many of which are positioned at the extremities of the search space. The surface and contour plot are provided in figure 4.16.

The experimental procedure expounded above is reapplied with the population sizes $(50 \overset{+}{,} 350)$, in which the parent population is partitioned into 10 clusters. Again to ensure accurate MPR results, the fitness values are modified as described for the multimodal function. Results for each test case are provided in table 4.5 and plotted in figure 4.17.

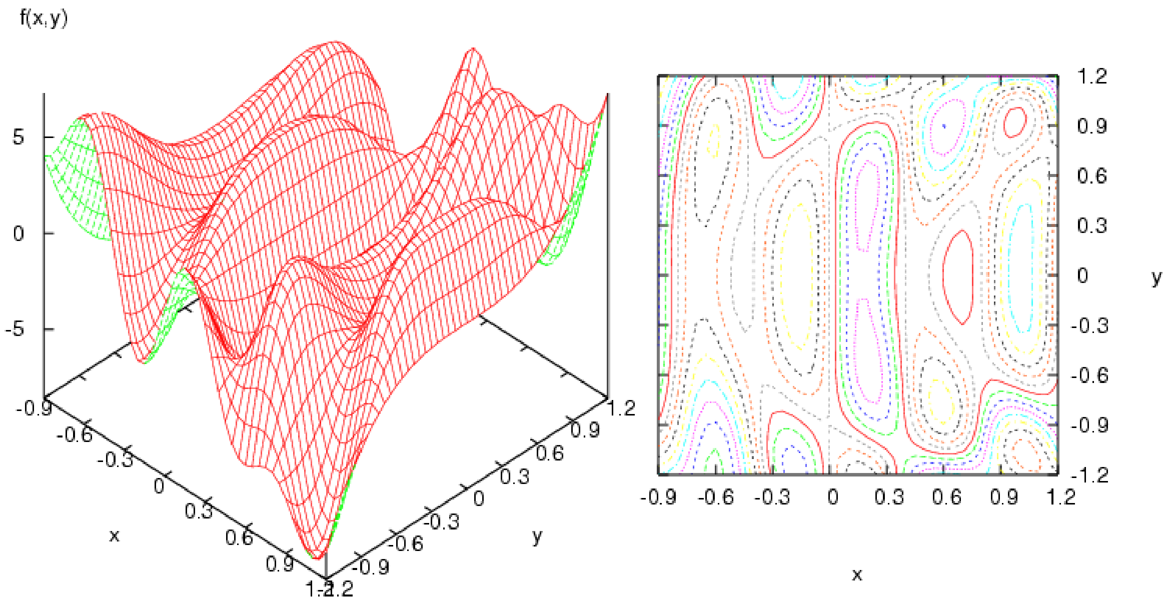


Figure 4.16: Waves function landscape and contour plot

Results

				Waves Function					
		Algorithmic Parameters				Optima out of 10		MPR	
Algorithm		*(strat)*	*recomb*	Global out of 30	Total out of 30	mean	σ	mean	σ
FCES	1	*(50, 350)*	*Discrete*	30	0	6.10	1.24	0.69	0.16
CES	2	*(50, 350)*	*Discrete*	30	2	7.77	1.01	0.80	0.13
FCES	3	*(50, 350)*	*Intermediate*	30	0	6.57	0.90	0.56	0.15
CES	4	*(50, 350)*	*Centroid*	30	2	7.97	0.96	0.81	0.15
FCES	5	*(50+350)*	*Discrete*	30	0	6.53	1.14	0.74	0.17
CES	6	*(50+350)*	*Discrete*	30	1	8.03	1.03	0.83	0.14
FCES	7	*(50+350)*	*Intermediate*	30	0	6.90	1.30	0.66	0.20
CES	8	*(50+350)*	*Centroid*	30	2	8.17	1.05	0.84	0.15

Table 4.5: Multiple solution results of experiments on the waves function

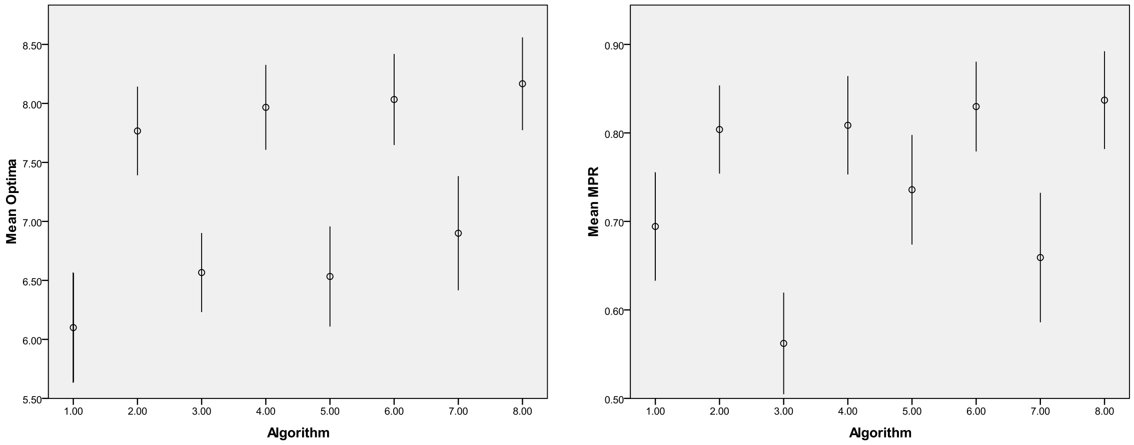


Figure 4.17: Mean and 95% confidence intervals for Optima and MPR results on the waves function

Discussion

These results corroborate the findings in earlier experiments: the CES algorithm repeatedly outperforms the FCES; again shown to be statistically significant by independent t-test for both the Optima and MPR measures. For this reason, results from the FCES are omitted from subsequent experiments. However, experimentation with the MSES is resumed for comparison with an alternative multiple solution algorithm.

4.3.4 Attribute 3: *Multidimensional* Proficiency

In this section, focus is placed on the performance of the CES as the dimensionality of the search domain is increased. Experiments are presented in which the algorithms are used to optimise solutions within two multi-dimensional test domains: the *n-dimensional sine* and *multimodal* functions. In the former, performance is assessed in terms of the algorithms' ability to maintain all search space optima as the dimensionality is raised. In the latter, performance is assessed when the number of optima significantly outnumber the number of search points in the population. It is worth noting that one drawback to Sullivan's clustering method is the need to specify the quantity of clusters *a priori*. If there are fewer clusters than peaks, global optimisation cannot be guaranteed (Sullivan, 2001). In many real-world applications the number of peaks is unknown, and may be too numerous to feasibly allocate a cluster to each.

Performance Criteria

For each variant of the CES algorithm tested here, the performance measures introduced throughout earlier sections of this thesis are employed. Namely, the *Global* optimum location percentage, the *Total* optima location performance, the *Optima* location average and the *MPR* (maximum peak ratio) as defined previously in equation 4.13. For a fuller description of these measures see section 4.3.3.

4.3.4.1 Experiments on the *n*-Dimensional Sine Function

The *n*-dimensional sine function has been employed previously for the purposes of testing *n*-dimensional niching algorithms (Streichert *et al*, 2003). The surface is defined by the equation:

$$f(\vec{x}) = 1 - \frac{1}{n}\sum_{i=1}^{n}\left(1 - sin^{6}(5\pi x_i)\right) \quad , \quad x_i \in [0\,, 0.4]\forall i \qquad (4.16)$$

Within this multimodal hyperspace there are $q_F = 2^n$ evenly distributed optima of equal fitness. The corresponding landscape and contour plots are shown in figure 4.18.

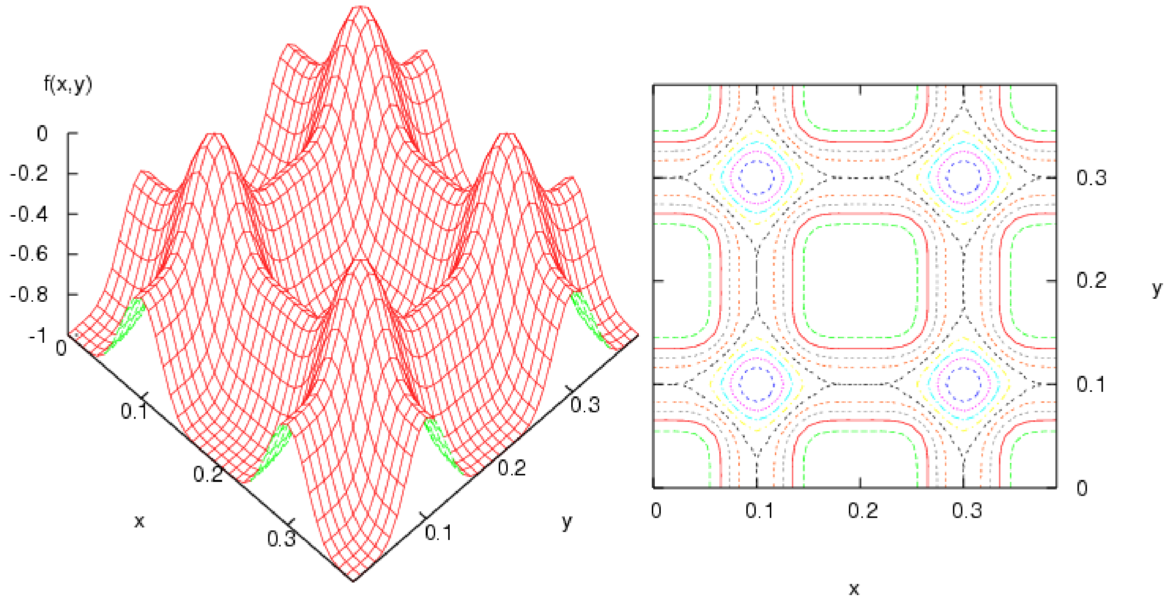


Figure 4.18: Two-dimensional sine function landscape and contour plot

The CES and MSES are next applied to the above test function, with the problem dimensionality varied in the range $[2\,,6]$. In each test case, the cluster quantity of the CES is adjusted to match the number of search space optima. The parent and offspring populations are scaled as before with five solutions per cluster and seven offspring per parent. To provide a fair comparison, the number of concurrent $(1+1)$ ESs within the MSES is set to the offspring size of the equivalent CES. Results are provided in table 4.6 and plotted in figure 4.19.

Results

						N-dimensional Sine Function					
Search Space			***Algorithmic Parameters***					Optima out of q_F		MPR	
n / q	***Algorithm***		*(strat)*	*recomb*	clusts	Global out of 30	Total out of 30	mean	σ	mean	σ
2 / 4	MSES	1	140 x (1+1)	-	-	30	30	4.00	0.00	1.00	0.00
2 / 4	CES	2	(20, 140)	discrete	*4*	30	30	4.00	0.00	1.00	0.00
3 / 8	MSES	3	280 x (1+1)	-	-	30	30	8.00	0.00	1.00	0.00
3 / 8	CES	4	(40, 280)	discrete	*8*	30	27	7.90	0.31	0.99	0.04
4 / 16	MSES	5	560 x (1+1)	-	-	30	30	16.00	0.00	1.00	0.00
4 / 16	CES	6	(80,560)	discrete	*16*	30	19	15.63	0.49	0.98	0.03
5 / 32	MSES	7	1120 x (1+1)	-	-	30	30	32.00	0.00	1.00	0.00
5 / 32	CES	8	(160,1120)	discrete	*32*	30	9	30.90	0.96	0.97	0.03
6 / 64	MSES	9	2240 x (1+1)	-	-	30	30	64.00	0.00	1.00	0.00
6 / 64	CES	10	(320,2240)	discrete	*64*	30	2	61.43	1.52	0.96	0.02

Table 4.6: Multiple Solution results of experiments on the n-dimensional sine function

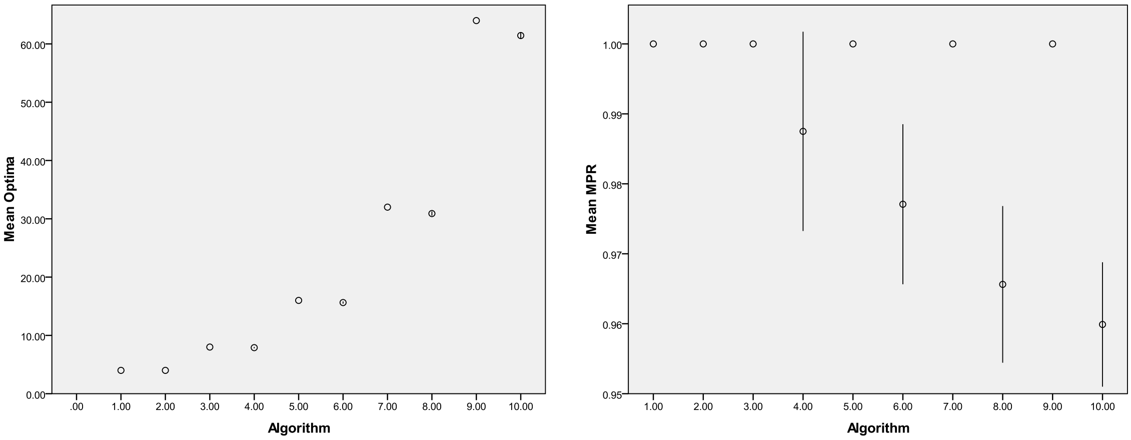


Figure 4.19: Mean and 95% confidence intervals for Optima and MPR results on the n-dimensional sine function

Discussion

From the results, a slight deterioration in performance of the CES is observed as the dimensionality of the problem space is increased. However, the location of, on average, 96% of all optima in a six-dimensional space indicates good niching performance. The MSES, by contrast, exhibits flawless performance within this environment. This result is due to the quantity of concurrent strategies outnumbers the search space optima by a ratio of 35:1. With such a large number of strategies all peaks are easily found, as random initialisation ensures that at least one of the $(1+1)$ strategies is seeded within the gradient of each optimum. In the next experiment, performance is assessed in the reverse situation: when the number of strategies/offspring are significantly outnumbered by the quantity of search space optima.

4.3.4.2 Experiments on the Multimodal Function

This experiment is intended to simulate the scenario in which little is known of the landscape topology, yet multiple high-fitness optima are still required. For all experiments the exogenous parameters are fixed at $(50\,,750)$ with the parent population being partitioned into five clusters at each generation. The algorithm is then applied to the problem while the search space dimensionality is varied from three to 10. For comparison, the MSES, running 750 $(1+1)$ strategies is applied within the same domain. The results are provided in table 4.7, and plotted in figure 4.20.

Results

						Multimodal Function				
Search Space			*Algorithmic Parameters*				Optima out of q_F		Best solution (min) avr	
n / q_F	*Algorithm*		*(strat)*	*recomb*	clusts	Global out of 30	mean	σ	mean	σ
3 / 125	MSES	1	750 x (1+1)	-	-	30	123.27	1.28	0.00	0.00
3 / 125	CES	2	(150, 750)	discrete	*5*	29	5.00	0.00	0.00	0.01
4 / 625	MSES	3	750 x (1+1)	-	-	21	419.47	7.63	0.02	0.03
4 / 625	CES	4	(150, 750)	discrete	*5*	24	5.00	0.00	0.01	0.03
5 / 3125	MSES	5	750 x (1+1)	-	-	5	660.27	7.57	0.08	0.04
5 / 3125	CES	6	(150, 750)	discrete	*5*	17	5.00	0.00	0.03	0.03
6 / 15625	MSES	7	750 x (1+1)	-	-	1	730.60	3.65	0.13	0.07
6 / 15625	CES	8	(150, 750)	discrete	*5*	23	5.00	0.00	0.02	0.04
7 / 78125	MSES	9	750 x (1+1)	-	-	0	745.83	2.23	0.26	0.08
7 / 78125	CES	10	(150, 750)	discrete	*5*	20	5.00	0.00	0.03	0.04
8 / 78125	MSES	11	750 x (1+1)	-	-	0	749.43	0.82	0.36	0.12
8 / 78125	CES	12	(150, 750)	discrete	*5*	14	5.00	0.00	0.06	0.06
9 / 1953125	MSES	13	750 x (1+1)	-	-	0	749.87	0.43	0.43	0.11
9 / 1953125	CES	14	(150, 750)	discrete	*5*	11	5.00	0.00	0.06	0.06
10 / 9765625	MSES	15	750 x (1+1)	-	-	0	749.97	0.18	0.53	0.10
10 / 9765625	CES	16	(150, 750)	discrete	*5*	14	5.00	0.00	0.06	0.07
10 / 9765625	CCES1	17	(30,210)	intermediate	-	30	1	0.00	0	0.00

Table 4.7: Multiple Solution results of experiments on the multimodal function

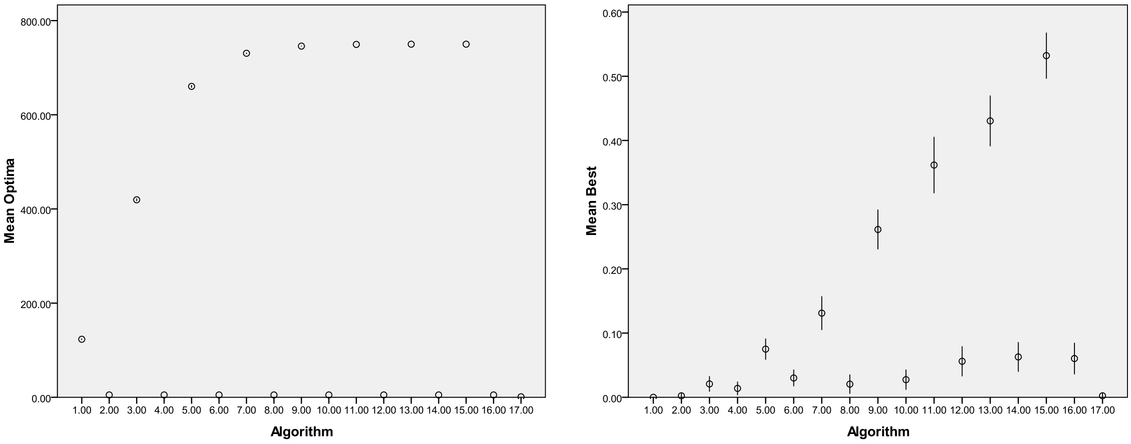


Figure 4.20: Mean and 95% confidence intervals for Optima and Best solution results on the multimodal function

Discussion

The weakness of the MSES is apparent in these results. In earlier experiments, the MSES performed well, however within such a vast multimodal space, such as this, the likelihood of an individual being seeded within the basin of the global optimum is negligible, and thus location of the global optimum becomes improbable.

The CES performs consistently well in all test cases; even with a relatively small number of clusters the global search capabilities are good. In the 10-dimensional space, the optimum is located in almost 50% of the test cases. Even when the optimum is not found, the value of the average final best solution is near-optimal. Note that with the population partitioned into five clusters, the CES is only able to maintain five distinct optima.

For comparison, and to lead into the next chapter, a similar experiment is run for the CCES1 cooperative coevolution strategy. Within this particular problem domain, the CCES performs exceptionally well, locating the global optimum in every test case. The CCES is clearly scalable and fast when applied to certain separable problems (Potter and De Jong, 1994). However, under the CCES architecture, the population will ultimately converge to a single optimum. In this work, locating the global optimum is not the only interest, it important that multiple high fitness solutions are also evolved. Indeed, this is the subject matter for the following chapter, in which a CES/CCES hybrid is synthesised, providing the first cooperative coevolutionary optimiser designed to maintain multiple search space optima, for fast and robust multimodal optimisation.

4.4 Summary of this Chapter

In this chapter, a novel niching algorithm called CES was developed and assessed. The specifics of the FCES, on which the novel algorithm is based, was described and reviewed. The factors precluding the maintenance of multiple distinct solutions in the FCES were indicated and discussed, subsequent to which a niching algorithm was developed which addressed each of these factors. An ES-based algorithm was developed which incorporates *k*-means cluster analysis with furthest point initialisation into the evolutionary cycle, with hard cluster recombination and a new *restricted cluster selection* operator to ensure that population species are preserved throughout the course of evolution. The resulting algorithm was termed clustering evolution strategy (CES).

The CES was then empirically examined in comparison with four other ES-based algorithms, including the FCES. Performance of each optimiser was tested in terms of their global optimisation, multiple distinct solution and multidimensional capabilities. The CES was found to be the most robust EA when all attributes were considered, although the reduction in selection pressure due to the properties of the niching operators resulted in a minor increase in convergence time. In terms of multiple solution performance, the novel recombination and selection operators, as well as the use of *k*-means cluster analysis, were confirmed statistically to improve the niching capabilities of the ES. In the last section the CES was demonstrated to scale well in multidimensional environments.

Chapter 5

Clustering Cooperative Coevolution Strategies – a New Synthesis

Coevolution is the biological term ascribed to a natural phenomenon in which there is reciprocal evolutionary adaptation between interacting species. Examples of specialisation due to coevolution are evident in the myriad *predator-prey*, *host-parasite* and *mutual* relationships exemplified in nature.

In the field of evolutionary computation, coevolution is used as a general term to refer to a particular type of evolutionary algorithm in which fitness is determined through the interaction of individuals within the evaluation environment (Ficici, 2004), (Bull, 2001). There are two categories of coevolutionary algorithms: Cooperative and Competitive Coevolution. The distinction between these two classes is a subject of much debate; however, explicitly defined examples of both are available in the literature, see for example the work of Potter and De Jong (2000) and Rosin and Belew (1997) respectively. Such discussion will not appear here, as the ultimate application domain presents a static, single objective parameter optimisation problem, an application domain for which the cooperative coevolutionary framework is most applicable.

However, the question arises as to what relevance cooperative coevolution has here? After all, traditional cooperative coevolutionary algorithms tend to converge towards a state of balance rather than optimality (Wiegand, 2004). The answer lies in the structure of the sound synthesis application domain considered here. Synthesisers, including the FM model adopted within this work, are often constructed from modular components, the parameters of which are easily decomposed into separate subcomponents suitable for optimisation by cooperative coevolution. In other words, the representational structure of the coevolutionary algorithm may be easily matched with the architecture of the underlying synthesis model, and thus parameter optimisation with this architecture may prove advantageous.

This chapter, therefore, begins with a brief introduction to the cooperative coevolutionary framework, followed by a short review of recent findings and refinements that bias the model towards more robust global optimisation. The motivation for creating a new multiple distinct solution cooperative coevolutionary model is then provided, followed by the introduction of a novel niching-based cooperative coevolutionary architecture designed to optimise multiple distinct solutions to complex function optimisation problems and improve the optimisation capabilities of the cooperative model. Thereafter, some empirical analysis is provided for the proposed algorithm when applied to a variety of test functions to demonstrate the general applicability of the architecture.

5.1 Cooperative Coevolutionary Algorithms

Cooperative coevolution is the name given to a particular type of EA in which individuals are assessed based upon their interaction with other individuals within the evolutionary system. In this chapter, focus is placed on the cooperative coevolutionary algorithm (CCEA) as defined by Potter and De Jong (1994) and Potter (1997)[4]. A brief review of the CCEA has already been provided in chapter three, in the context of multimodal static optimisation. While this is the application domain of interest here, the scope of CCEAs is much broader than this: CCEAs have been applied to many different types of learning problem; for example see the work of Potter and De Jong (1998), Potter and De Jong (2000) and Roberts and Claridge (2004).

When applying CCEAs for the purposes of parameter optimisation, the search space must be decomposed into separate subcomponents. A standard approach is to decompose the

[4] Although some researchers prefer the name *compositional coevolution* (Wiegand, 2005)

problem space such that each object parameter represents a separate subcomponent of the problem. This is referred to as a *natural* decomposition. Each subcomponent is then assigned and explored concurrently by an independent subpopulation of potential (component) solutions, which are varied in isolation by a conventional EA. In order to determine the fitness of a subpopulation individual, it must be adjoined with a member drawn from each of the other subpopulations to form a complete solution. The assembled solution is then evaluated as normal, in application to the problem, to yield a fitness value. It is through this collaboration mechanism that subpopulations are encouraged to cooperate with each other, to coordinate towards a common goal.

5.1.1 Collaboration

The process by which candidates are chosen and fitness is allocated is referred to as *collaboration*. Through the interaction space of the collaborative process, subpopulations are able to retrieve information about the underlying structure of the problem domain. There exists a selection of alternative collaboration techniques, from which the correct choice is an important factor in successful application of the CCEA. The subjective fitness rating that each individual receives is indicative of its performance in combination with collaborators from neighbouring subpopulations. In light of this, collaboration may be viewed as the process of *sampling the interaction space* between subpopulations (Wiegand, 2004). In some applications, the interaction space may be sufficiently characterised by a single collaboration between subpopulations; in others, more sophisticated schemes may be required. Consequently, there is a variety of collaboration techniques, and selection of an appropriate method should be chosen according to the topology of the problem domain. Comprehensive studies of many different collaboration, or *partnering,* strategies have been conducted by Bull (1997) and Wiegand (2001). In Weigand's study, three collaboration attributes were examined for the CCEA:

- collaboration pool size.
- collaborator selection pressure.
- collaborator credit assignment.

The collaboration pool size indicates the number of collaborations an individual undergoes before a final fitness value is assigned. The collaborator selection pressure controls the amount of bias towards collaboration amongst the best individuals of each subpopulation. The collaborator credit assignment determines the method by which resulting fitness values are aggregated to yield the final value.

The most consistent finding of Wiegand's study was in the credit assignment procedure. When agglomerating fitness, the most effective strategy is to discard all but the result of the most profitable collaboration. This method was first introduced by Potter (CCEA-2) (Potter, 1994), and shown to produce good results in Bull (1997). Wiegand refers to this assignment method as *optimistic*.

The selection pressure is controlled by selecting collaborators according to their rank within their respective subpopulations. Selection of the best collaborator (as in CCEA-1) constitutes the most extreme selection pressure, and conversely, selecting the worst constitutes the weakest pressure. Potter's *single-best plus one random* collaboration method (CCEA-2) therefore represents a slightly weakened version of the single-best collaboration method, as the second collaborator is chosen randomly. Interestingly, Wigand's method for varying the selection pressure had little effect on the outcome of his experiments in both separable and inseparable problems. The main factor that was found to affect the performance of the CCEA was the *collaboration pool size*.

In most instances, it was shown that increasing the collaboration pool size results directly in an observed performance improvement. This result is perhaps unsurprising, as multiple collaborations sample more information from the interaction space at the cost of additional objective function evaluations. However, when the problem is linearly separable, no improvement in performance over the *single-best* collaboration strategy (CCEA-1) can be attained; an observation again corroborated by Bull (1997).

While Wiegand's study does provide useful information regarding the role of collaboration within CCEAs, it does not directly address which collaboration methods are most suitable for different types of problem. Popovici and De Jong (2005) have made recent progress to this end, by analysing the runtime dynamics of CCEAs on simple two-dimensional optimisation problems. Through visualisation of the best-of-generation dynamics on top of a problem's *best-response curve* (Popovici and De Jong, 2004), it is possible to predict how different collaboration methods may affect algorithm behaviour in other search domains. While this work is beginning to explain the runtime behaviour of CCEAs, significant analysis of the search space is required to produce best-response curves, and application to more complex, high-dimensional real-world problems remains as future work.

5.2 CCEAs For Parameter Optimisation

As was confirmed in chapter four, CCEAs are capable of performing both better and worse than traditional EAs. The question then arises: under which circumstances is there a CCEA advantage? It is still unclear exactly which search properties are most suited for optimisation with CCEAs. As such, this question remains open; however some progress has recently been made.

Wiegand identifies that any performance disparity between traditional EAs and CCEAs can only be attributed to the two real differences between them: the division of the problem into multiple subspaces, and the subsequent increased effects of the search operators within these contracted spaces. Wiegand refers to these properties as *partitioning and focusing* (Wiegand, 2004). By dividing the problem into subcomponents there is an exploratory advantage, as each component may be varied with less risk of disrupting an entire solution. In other words, the notion of *divide and conquer* is embedded within the architecture.

5.2.1 Separability, Decomposition and Cross-Subpopulation Epistasis

It was originally contended that CCEAs are most suited for application domains that are divisible into independent subcomponents, between which there are no epistatic interactions (i.e. separable problems) (Potter and De Jong, 1994) (Bull (1997). While separability is certainly a relevant factor, Wiegand *et al* have demonstrated that search space attributes which directly affect the performance of CCEAs are more complex than merely the absence, or presence, of separability alone (Jansen and Wiegand, 2003), (Jansen and Wiegand 2004) and (Wiegand, 2004).

When a CCEA is tuned such that the problem decomposition *matches* the problem's separability, the representation ensures that there is no cross-subpopulation epistasis, and the single-best collaboration strategy is all that is required to yield a good solution. However, if there are non-linear interactions between parameters, represented by differing subpopulations, there may be a certain degree of cross-subpopulation epistasis, which may require more complicated collaboration strategies in order to infer any useful gradient from the interaction space. To further complicate matters, the mere presence of cross-subpopulation epistasis does not alone justify the use of more complicated collaboration strategies; it is the type of epistasis that must be considered (Wiegand, 2004).

Wiegand demonstrated that a pseudo-boolean *TRAP* function was particularly damaging to cooperative coevolution, as it exhibits a characteristic known as *contradictory cross-*[sub]*population epistasis*. In application to this particular problem, Wiegand proved that, if component solutions are not present within the subpopulations at initialisation, the problem cannot be solved; the mechanism of the algorithm will itself preclude global optimisation. This is because the coevolutionary algorithm conducts its search by locking all but the currently evolving subcomponent. In certain environments this approach is advantageous, in others it can be disastrous. As Wiegand describes:

> *It is not hard to see that the same partitioning process that assists the CCEA in gaining advantage against the EA in certain situations can become its Achilles heel in the presence of this* [contradictory cross-subpopulation epistasis] *deception*
>
> Wiegand (2004, p.94)

However, Popovici and De Jong (2005) have since established that contradictory cross-subpopulation epistasis is not always a good indicator of problem difficulty. This finding was demonstrated on two simple real-valued functions, both of which exhibit contradictory cross-subpopulation epistasis. While it is possible to solve both functions with a CCEA, the more interesting outcome of this work was the observation that the collaboration method that most effectively solved one problem was the least effective at solving the other. Moreover, when solving the first (*oneRidge*) test problem, it was found that the greedier the collaboration scheme and the larger the subpopulation sizes, the worse the results (Popovici and De Jong, 2005). This work indicated that tactics frequently employed to enhance the performance of conventional EAs are not always transferable to CCEAs.

5.2.2 Relative Overgeneralisation

Relative overgeneralisation is a phenomenon which occurs as a result of the subjective nature in which fitness is assessed within the CCEA model (Wiegand, 2004). Each individual is assessed in terms of its performance when collaborating with only a subset of the potential interaction space. Search is then conducted through projections along the dimensions of the search space, which may preclude optimal collaboration by rendering regions of the space inaccessible. Within each subpopulation, selection will tend to favour those individuals that project search across distributions of high average, and not necessarily optimal fitness. In other words, deceptive landscapes with broad suboptimal peaks (of high average fitness) and a narrow optimal peak (of lower average fitness) are problematic for cooperative coevolutionary algorithms: CCEAs have a propensity for consensus over optimality. These observations lead to the conclusion that CCEAs are not static optimisers of optimal collaboration, but rather optimisers of robust collaboration. They optimise collaborators that offer the most resilience to variation in other parts: the so-called *robust resting balance* (Wiegand, 2004). However, for certain applications, practitioners may actually be interested in optimising for robustness rather than maximum payoff. This may go some way to explain why CCEAs have been found to produce good results in multi-agent team environments (Potter *et al*, 2001), (Wiegand, 2006).

To demonstrate the tendency of CCEAs to converge to a robust resting balance, Wiegand devised the Maximum of Two Quadratics (MTQ) function, which exhibits exactly the properties described above: a broad suboptimal peak and a narrow optimal peak (see figure 4.8, page 72). When applied to this problem the CCEA was found to converge to the suboptimal peak; results which were repeated in chapter four.

5.2.3 Modified CCEAs for Single Objective Static Optimisation

In the final chapter of his doctoral thesis, Wiegand (2004) offers two methods by which the static optimisation performance of CCEAs might be improved: biasing for optimal collaboration and balancing evolutionary change.

Biasing for Optimal Collaboration

The first of Wiegand's suggestions modifies the credit assignment procedure to bias the model toward the discovery of optimal collaborators. This is achieved by computing the fitness of an individual by the weighted sum of two terms: the first, ascertained via the usual collaboration procedure; and the second, by estimating an individual's maximum possible fitness, had it interacted with its optimal collaborators (Wiegand, 2004), (Panait *et al*, 2004). While this method has sound theoretical grounding, it is not immediately obvious how a solution's best possible fitness can be approximated. In fact, if it were possible to generate an accurate estimate, there would be no need for any interaction between subpopulations at all (reducing each subpopulation to a conventional EA). One approach is to produce estimates based on historical data gathered during the run (Panait *et al*, 2003). This is achieved by dividing a solution's search space into discrete intervals; each interval may then be assigned a *MaxReward* value, which is equal to the highest fitness it has been rewarded so far. Obviously, the *MaxReward* values give an increasingly accurate approximation of an interval's maximum possible reward as the search proceeds, and, to account for this, the contribution of the *MaxReward* value is increased accordingly throughout the run. This relatively straightforward mechanism was found to offer improved performance over the vanilla CCEA on a simple two-dimensional problem by Panait *et al* (2003).

Balancing Evolutionary Change

Wiegand's second approach augments the original model by imposing a spatial embedding on each coevolving subpopulation, see also Wiegand and Sarma (2004). This embedding is synonymous with the *diffusion* model, reviewed in section 3.5.2, whereby each subpopulation individual is distributed on a toroidal grid. Reproduction takes place within a pre-defined neighbourhood (deme) on the grid, and collaboration is similarly confined between members of the corresponding demes of the other subpopulations. This provides some notion of linkage between the neighbourhoods of each population, which are able to explore the space quasi-independently. Strong adaptations are able to pervade the populations by diffusion in a controlled fashion, leading to a steady and symmetrical rate of convergence between coevolving subpopulations. The maintenance of symmetry in the convergence dynamics is one of the key motivations behind this approach, as it retains comparable levels of diversity between cooperating populations. Should one population totally converge before the rest, there may be a *loss of gradient* in the interaction space, which may ultimately lead to poor optimisation performance (Wiegand, 2004). Wiegand implemented a simple example, demonstrating improved performance when applied to the MTQ function over the baseline CCEA.

Optimal Collaboration by *Finer-Grained* Comparison

A further adaptation of the standard cooperative coevolutionary model, also designed to improve global optimisation performance, has been developed by Bucci and Pollack (2005). This approach employs Pareto dominance comparison, borrowed from Multiobjective optimisation (Deb, 2001). Individuals are selected based upon which solutions (Pareto) dominate the remainder of the population. By selecting with Pareto dominance, the CCEA becomes more sensitive to informational differences within the population (due to multiple peaks in the search environment), where conventional CCEAs are only sensitive to relative fitness ratings. It was demonstrated that the Pareto CCEA (pCCEA) was able to locate the global optimum of the MTQ function more reliably than a conventional CCEA-2 and a *complete mixing* CCEA. Although not stated in their paper, Bucci and Pollack’s pCCEA was also capable of maintaining multiple solutions at distinct optima as a result of the underlying selection techniques (Bucci, 2007). However, there is a considerable computational impact as a result of the Pareto dominance procedure. To facilitate Pareto dominance selection, it is required for subpopulation individuals to participate in every possible combination of collaborations with members of the other subpopulations (*complete mixing*) within the system, a process which requires a significant increase in fitness evaluations at each generation, which is combinatorial with the number of subpopulations.

5.2.4 A Practical Alternative

The theoretically sound methods for optimal collaboration identified above, have been shown to provide excellent results when applied to simple two-dimensional optimisation problems that otherwise pose a problem for conventional CCEAs. However, their practical application to larger, more complicated problem domains is not entirely clear. For example, Panait *et al*’s (2004) paper, documenting the application of Wiegand’s biasing method, is closed with the recognition that application to larger continuous problems will require further exploration. Additionally, the use of complete mixing in Bucci and Pollack’s (2005) Pareto CCEA, requires $\mu^{(p-1)}$ function evaluations per individual, where μ is the number of parent individuals in each population and p is the number of subpopulations. A computational overhead which becomes impractically large as the problem dimensionality (and thus p) is increased.

To provide an alternative, practical and scalable solution, it is possible to glean ideas from these methods and take a utilitarian approach. The desirable attributes of the system required here are as follows:

- to deliver multiple solutions at distinct optima in multimodal problem domains.
- to provide improved optimal collaboration over the baseline CCEA.
- to retain the fast convergence properties of cooperative coevolution.

To enable the CCEA to maintain multiple independent search optima, inspiration is drawn primarily from the spatial embedding model proposed originally by Wiegand (2004). Wiegand’s adapted model is designed to improve the likelihood of optimal collaboration by introducing, to the CCEA model, a diversity preservation technique often applied to improve the performance of conventional EAs. A similar approach is adopted here by utilising the *k*-means clustering-based niching method of chapter four to partition each cooperating subpopulation into multiple clusters. This synthesis forms the second algorithmic contribution of this work: the clustering cooperative coevolution strategy or CCCES.

In the following sections a new cooperative coevolutionary model is introduced, and experimental evidence is provided to demonstrate its niching and improved optimal collaboration capabilities over the traditional CCEA. It will also be demonstrated that the proposed model maintains the fast convergence characteristics of the conventional CCES.

5.3 Niching in Coevolutionary Algorithms

The distributed population structure, employed by Wiegand (2004), is not the first instance in which coevolution researchers have adopted diversity preservation techniques to improve the performance of their algorithms. There are many examples in which competitive coevolutionary algorithms have benefited from similar techniques to evolve strong solutions to test-based problems, for example, evolving game players and sorting networks. For example, an early coevolutionary application for evolving minimal sorting networks by coevolution is described by Hillis (1989). Near optimal results are attained when the genetic diversity is maintained with the use of a *diffusion* style population structure, an approach adopted also by Husbands (1994) in application to a generalised version of the job shop scheduling problem. Fitness sharing methods have also been applied for this purpose in order to evolve well rounded game-players by Smith and Gray (1994), Darwen (1996), Rosin and Belew (1995) and Cartlidge (2003).

Interest in the maintenance of diversity in this thesis stems from the desire to maintain multiple distinct solutions to the problem, and by doing so, it is hoped that sufficient adaptive balance is maintained between populations to induce improved optimal collaboration performance.

5.3.1 The Niching Cooperative Coevolutionary Algorithm (NCCEA)

In this section, a novel coevolutionary algorithm called the niching cooperative coevolutionary algorithm (NCCEA) is proposed. The NCCEA follows largely the same procedure as a conventional CCEA with one important difference: the underlying algorithm is a niching EA. To facilitate the concurrent maintenance of solutions in distinct regions of the component space, the individuals belonging to each coevolving subpopulation must be partitioned in to subgroups. To avoid confusion, these (within subpopulation) subgroups are referred to as species, henceforth. A diagram of the model is provided in figure 5.1.

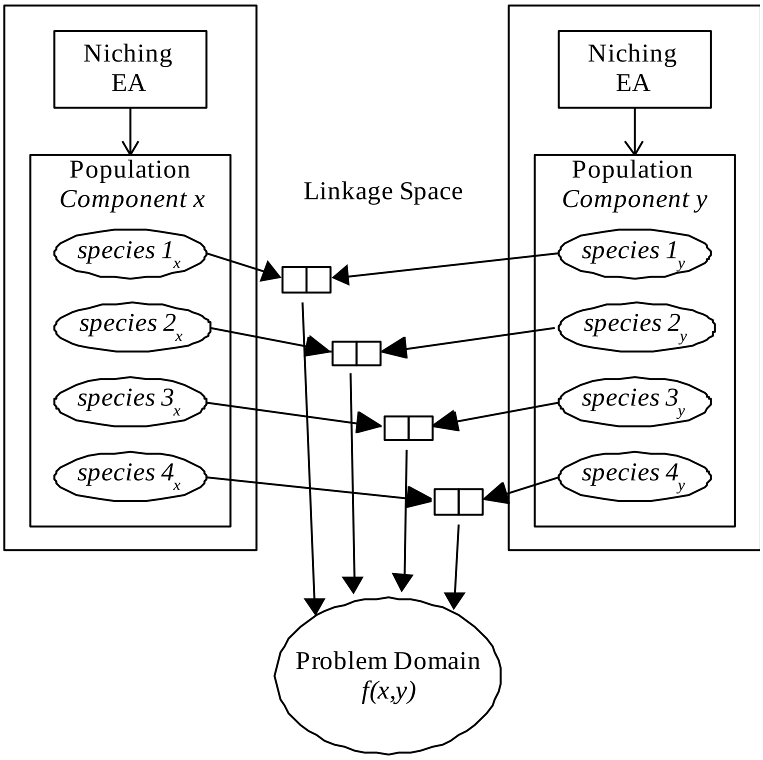


Figure 5.1: Two population NCCEA

For the purposes of evaluating fitness, there are several alternatives. One possible approach is to associate, or *link*, each subpopulation species (as shown) with the corresponding species of the other coevolving subpopulations. In this configuration each individual, belonging to a given species collaborates with individuals drawn from the linked species of the neighbouring populations. Any existing collaboration strategy may be adopted for this purpose; e.g. the *single-best* or *single-best plus one random* methods may be easily implemented.

In this respect, Wiegand and Sarma’s (2004) spatially embedded CCEA may be considered a special case of this model, in which the geographic structure of the subpopulations and the linkage between each species is fixed. However, Wiegand’s algorithm is unable to deliver multiple solutions at distinct optima as the underlying algorithm is not a niching EA.

5.3.1.1 Collaboration

One notable drawback of a fixed linkage model is the situation in which optimal, or at least near-optimal, component solutions exist within the system but are unable to collaborate as their species are not linked. For optimal collaboration to subsequently take place, these adaptations must either survive within their current linkage configuration, propagating to surrounding species by some mode of immigration (diffusion in Wiegand's model); or emerge independently within the neighbouring species.

However, an alternative technique is available that reuses the available niching information to derive the linkage configuration that achieves maximum payoff. The procedure is as follows:

- every possible linkage combination between subpopulation species is considered a valid link (*complete linkage*).
- fitness for each offspring is then assigned as the maximum fitness achieved when all possible linkage collaborations have taken place.

In the model depicted in figure 5.1, this method corresponds to the pairing of each offspring individual for *Component x* with the best[5] individual from each species of subpopulation for *Component y*; four collaborations in total.

5.3.1.2 Diverse Collaboration

When applied to a problem in which a more complicated collaboration procedure is necessary to ensure sufficient characterisation of the interaction space, the NCCEA provides a more methodical technique for selecting multiple collaborators than simply selecting collaborators arbitrarily (*random mixing*) (Wiegand et al, 2001), and a more efficient method for guaranteeing collaboration between diverse subpopulation members than exhaustive selection methods (*complete mixing*) (Bucci and Pollack, 2005). Random selection of individuals does, to some extent, increase the probability of selecting a diversity of collaborators. However, given that the underlying niching algorithm is itself designed to identify and preserve diversity, each species represents clusters of population members that are similar to each other and different from the rest. Therefore, collaboration between representatives from each species ensures diverse collaboration. However, as noted earlier, additional collaborations are costly and should be minimised. Pareto cooperative coevolution was previously discarded due to the computational impracticalities

[5] if a *single-best* collaboration procedure is employed

of complete mixing in high-dimensional problem domains. This factor is considerably reduced in the proposed niching method in which s^{p-1} function evaluations are required per offspring (where s represents the number of species) rather than μ^{p-1}. However, as the maximal linkage is established anew for each generation, the number of required fitness evaluations can be reduced further still by enabling the linkage configuration to adapt as the system evolves, a procedure termed *dynamic linking*.

5.3.1.3 Dynamic Linking

Dynamic linking is a process by which representatives drawn from each species are evaluated to establish a single linkage configuration that produces the highest payoff. The configuration then defines how each offspring within the algorithm should collaborate with members of adjacent subpopulations.

The maximal-reward inter-subpopulation linkage configuration for each species can be established by assessing every possible collaborative combination of individual representatives chosen from each species. The linkage configuration may then be used when assessing the fitness of new individuals. The offspring of a particular species are each adjoined with a combination of representatives from neighbouring species that were identified to be most profitable during the linkage configuration. For example, the maximal-reward inter-species linkage at a given generation may be consistent with the configuration depicted in figure 5.1. However, for the subsequent generation, the linkage depicted in figure 5.2 may be found to be maximal. By recalculating the linkage configuration at each generation, the strongest collaborative links between species are identified and maintained.

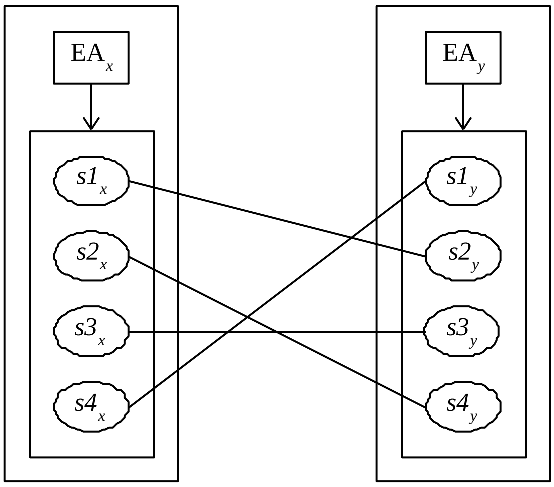


Figure 5.2: NCCEA showing different linkage arrangement

If one representative is selected from each species, this procedure reduces the number of generational objective function evaluations to one per offspring (the same as the *single-best* strategy of the conventional CCES), in addition to s^p evaluations required to compute the linkage table.

5.3.1.4 Maintaining Diversity with Exclusive Linkage

If all linkage permutations are considered to be valid, it is possible that the dynamic linkage procedure may deliver a configuration similar to that depicted in figure 5.3.

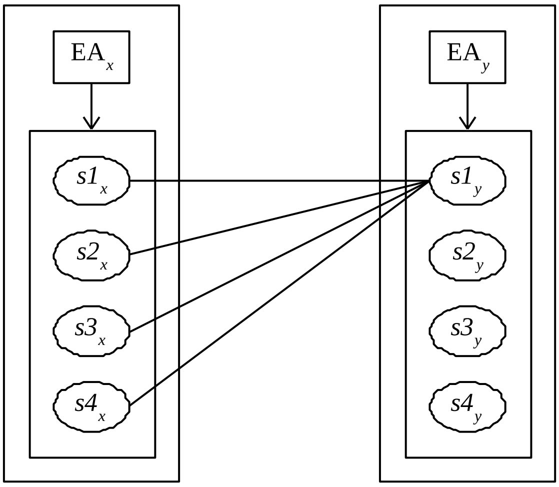


Figure 5.3: NCCEA with common linkage

In this situation all species in the x component subpopulation are found to produce maximal results in collaboration with the representative of the first species in the y component subpopulation. In this linkage configuration the likelihood of every species converging toward the same point is significantly increased, as all subpopulation species are evaluated along the same projection. By prohibiting *many-to-one* linkage arrangements, such as this, all subpopulation species are forced to link exclusively with only one species from each subpopulation. To ensure that the maximal link is kept intact, the first link is set between the highest-scoring representatives and subsequent links between remaining species are set in order of fitness.

Within the framework of the proposed NCCEA there is potential for a variety of implementations. For example, any niching EA may be adopted as the underlying algorithm, and alternative techniques for deriving the linkage configuration may also be considered. Analysis of the multifarious incarnations of this model is beyond the scope of this work; only one example of the model is implemented here, based upon the principles presented earlier in this chapter. In the next section, an instance of the NCCEA is introduced, which implements the CES to evolve separate subpopulation species. Details of the algorithm are provided, followed by an empirical analysis in application to a variety of test problems.

5.3.2 A Niching Cooperative Coevolutionary Algorithm – The CCCES

To test the NCCEA concepts which were set out in previous sections, an ES-based instance of this model will be examined that employs the clustering-based niching method introduced in chapter four. Recall the CES model, in which the parent population of the ES is partitioned into separate species by *k*-means cluster analysis. In the proposed algorithm each coevolving subpopulation is evolved independently by a separate CES; this implementation of the model is referred to as the Clustering Cooperative Coevolution Strategy or CCCES.

The algorithm is defined by the following pseudocode:

```
initialise ($\mu$);
for each subpopulation $p$
    evaluate_initial($\mu_p$);
    cluster ($\mu_p$);
    $R_p$ = select_reps($\mu_p$);

L = calc_linkage(R);

round loop begin
    for each subpopulation $p$
        $\lambda_p$ = recombine($\mu_p$);
        $\lambda_p$ = mutate($\lambda_p$);
        evaluate($\lambda_p$ , L);
        $\mu_p$ = select($\lambda_p$ ( + $\mu_p$));
        cluster($\mu_p$);
        $R_p$ = select_reps($\mu_p$);
        L = calculate_linkage(R);
round loop end;
```

Figure 5.4: NCCEA pseudocode

The CCCES algorithm begins by initialising all parent subpopulation members with a number drawn randomly from within its object range. Individuals of each subpopulation are then evaluated for fitness in collaboration with individuals selected at random from each of the other populations within the system. All populations are subsequently partitioned into species by *k*-means cluster analysis and species representatives are selected as the fittest individual belonging to each cluster. Thereafter, representatives are used to compute the maximal linkage configuration according to the *exclusive-linkage* method outlined earlier. The generational cycle of the CCCES may now commence.

Individuals of the currently evolving subpopulation are first recombined, according to the cluster-based recombination operators presented in chapter four, and then mutated using Ostermeier's (1994) derandomised self-adaptive mutation operator. Offspring are subsequently evaluated in collaboration with species representatives according to the linkage table `L` and then selected as parents by *restricted cluster selection.* Finally, the subpopulation cluster membership is recomputed, new representatives are selected and the linkage configuration is updated.

In the system proposed here, the exclusive-linkage is calculated by evaluating every collaboration permutation between the representatives of each species to produce a table of fitness results. The first link is then defined as the fittest collaboration in the table. To ensure exclusivity, any collaboration in the fitness-table that specifies species connected by the first link are removed, and the fittest remaining collaboration is assigned as the second. The procedure then continues until each representative is exclusively linked.

5.3.2.1 CCCES Parameters: Cluster Quantity and Problem Decomposition

In the CES, the value of the clustering parameter k specifies the number of species into which the population is partitioned, and thus the number independent solutions the algorithm is able to maintain. The same applies to the CCCES, where k specifies the number of species into which each coevolving subpopulation is divided, and again the number of solutions the algorithm is able to optimise concurrently at distinct optima. As such, the guidelines set out for the CES when choosing an appropriate value for this parameter also apply here to the CCCES. However, there are additional, computational factors that should be considered when optimising with the CCCES.

As discussed earlier, the number of fitness evaluations required to calculate the linkage configuration at each generation is s^p, where s is the number of species and p, the number of coevolving subpopulations. Increasing either parameter will raise the number of evaluations required to compute the linkage configuration.

The optimum value for k is problem dependant, so should be set either equal to the number of desired independent solutions or adjusted within the range $2 \leq k \leq \mu/3$ (see section 4.2.4). As the number of fitness evaluations required to compute the linkage configuration is combinatorial with the number of coevolving subpopulations, increasing the parameter p should be done with caution. The number of subpopulations within a cooperative coevolutionary system relates directly to the decomposition of the problem space. As with the number of subpopulation species, the optimum decomposition is closely coupled with the characteristics of the problem domain. Consequently, there are no guidelines appropriate for all classes of problem. The value for this parameter must again be chosen by the practitioner as with conventional CCEAs, and kept as small as possible.

5.4 An Analysis of Performance in Selected Test Environments

In the following sections the performance of CCCES is tested in application to a variety of test problems, comparing results with the baseline CCEA, in terms of multiple distinct solution maintenance, optimal collaboration and convergence velocities.

Experimental Set-up

In the first set of experiments focus is placed on the ability of the CCCES to locate the global optimum while simultaneously maintaining multiple distinct suboptima. Two multimodal problem domains are chosen that were introduced in chapter four of this thesis: Himmelblau’s function and the multimodal function. In the second set of experiments, performance of the CCCES is examined in comparison with several variants of the conventional CCES model in which a variety of collaboration strategies are adopted. The test environment of this comparison is the Maximum of Two Quadratics function, as it has previously been shown to exacerbate the relative overgeneralisation pathology of the traditional CCES. The results of these experiments should indicate whether the niching model provides improved optimal-collaboration performance.

For the subsequent experiments, the test procedure employed in chapter four is reapplied. Results are averaged over 30 runs, in which each algorithm is executed for 50 rounds[6]. As in previous experiments, a selection pressure is maintained at a ratio of $\mu/\lambda = 1/7$. The cluster cardinality, or number of individuals per species, is given by the formula μ/c, and exact subpopulation sizes are indicated for each experiment.

5.4.1 Experiments on Himmelblau's Function

The modified Himmelblau function was previously introduced in section 4.3.2, in which the ability of the CES algorithm to maintain multiple solutions at distinct optima was examined. The function is defined by equation 4.14 (page 79) which produces a landscape that features four equally sized peaks, as depicted in figure 4.13. In the experiments that follow, the CCCES is repeatedly applied to Himmelblau's function with increasing subpopulation sizes.

As with the experimentation with CCES in chapter four, the problem space of each test function is decomposed naturally, such that each parameter is represented by a separate subpopulation (two for all experiments here). Each individual encapsulates only one parameter. Discrete recombination, as in chapter four, is equivalent to no recombination, where parameters are varied only by mutation between successive generations.

Previous performance metrics are employed to indicate the number of runs in which all optima are successfully identified and maintained (*Optima*), and the actual number of separate optima located, averaged over all test runs (*Total*).

The results from each test case are provided in table 5.1. Details of each algorithm are presented in the format:

(strat)	*recomb*	*clusts*

In which *strat* indicates the subpopulation size and selection mechanism in traditional $(\mu \overset{+}{,} \lambda)$ ES notation; *recomb* indicates the recombination type (centroid or discrete); *clusts* indicates the number of clusters, or species, within each subpopulation.

Results

The results are shown in Table 5.1 and plotted in figure 5.5.

[6] i.e. in one round all subpopulations have advanced by one generation.

					Himmelblau's Function		
		Algorithmic Parameters				Optima out of 4	
Algorithm		*(strat)*	*recomb*	*clusts*	Total out of 30	mean	σ
CCCES	1	(8,56)	*centroid*	*4*	19	3.63	0.49
	2	(8,56)	*discrete*	*4*	18	3.60	0.50
	3	(8+56)	*centroid*	*4*	23	3.77	0.43
	4	(8+56)	*discrete*	*4*	21	3.70	0.47
	5	(12,84)	*centroid*	*4*	16	3.53	0.51
	6	(12,84)	*discrete*	*4*	22	3.73	0.45
	7	(12+84)	*centroid*	*4*	21	3.70	0.47
	8	(12+84)	*discrete*	*4*	20	3.67	0.48
	9	(16,112)	*centroid*	*4*	18	3.60	0.50
	10	(16,112)	*discrete*	*4*	21	3.70	0.47
	11	(16+112)	*centroid*	*4*	24	3.80	0.41
	12	(16+112)	*discrete*	*4*	25	3.83	0.38
	13	(20,140)	*centroid*	*4*	18	3.60	0.50
	14	(20,140)	*discrete*	*4*	19	3.63	0.49
	15	(20+140)	*centroid*	*4*	24	3.80	0.41
	16	(20+140)	*discrete*	*4*	24	3.80	0.41
	17	(40,280)	*centroid*	*4*	26	3.87	0.35
	18	(40,280)	*discrete*	*4*	24	3.80	0.41
	19	(40+280)	*centroid*	*4*	29	3.97	0.18
	20	(40+280)	*discrete*	*4*	26	3.87	0.35
	21	(80,560)	*centroid*	*4*	30	4.00	0.00
	22	(80,560)	*discrete*	*4*	30	4.00	0.00
	23	(80+560)	*centroid*	*4*	29	3.97	0.18
	24	(80+560)	*discrete*	*4*	29	3.97	0.18

Table 5.1: Results of CCCES on Himmelblau's function

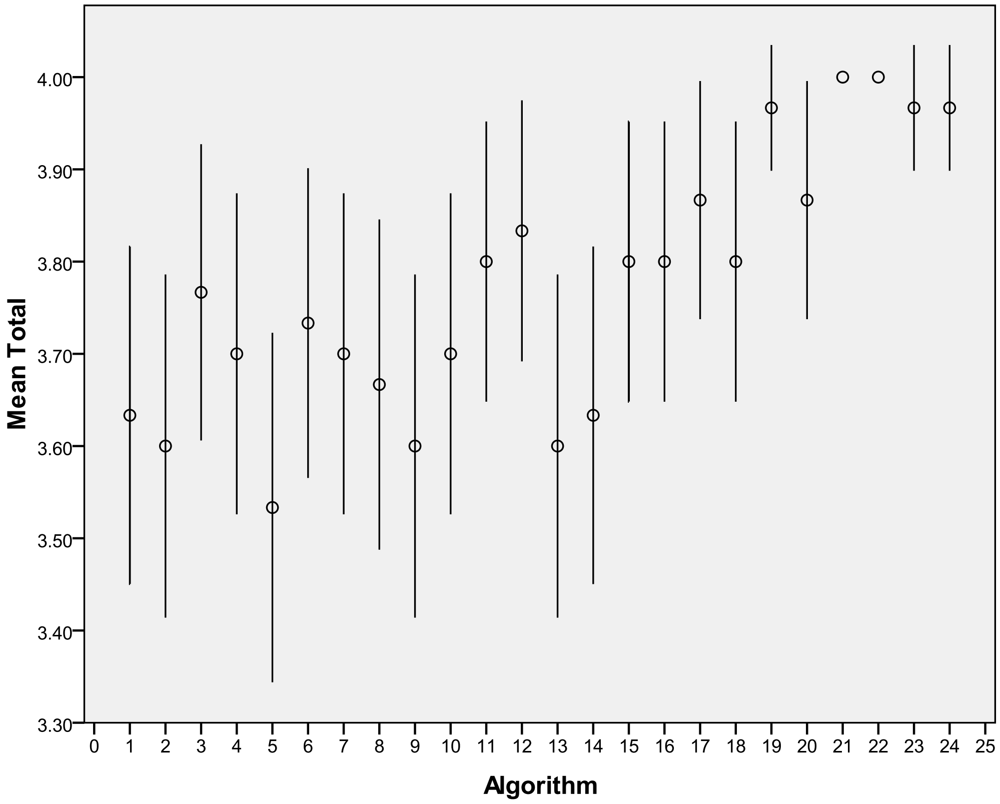


Figure 5.5: Mean and 95% confidence intervals for Total results on Himmelblau's function

Discussion

The results shown in table 5.1 and figure 5.5 indicate that the CCCES is indeed capable of locating and maintaining multiple search space optima. This niching behaviour is a consequence of the introduction of speciation and inter-species collaboration to the CCEA model, which would otherwise only be capable of maintaining one solution.

When applied to Himmelblau's function, the results indicate a positive linear association between the population size and the Optima and Total measures. Statistical analysis confirms this observation with a Pearson correlation of 0.77, significant at the 0.01 level (two-tailed). Furthermore, the multiple solution performance of the CCCES is also frequently enhanced when an elitist strategy is adopted; however, this trend was not found to be statistically significant.

In many of the test cases only three of the four optima are maintained; when subpopulation sizes are small, species frequently converge towards the same peak. To assist in understanding exactly why the CCCES fails to consistently maintain all four optima, the best response curves for Himmelblau's function are plotted in figure 5.6.

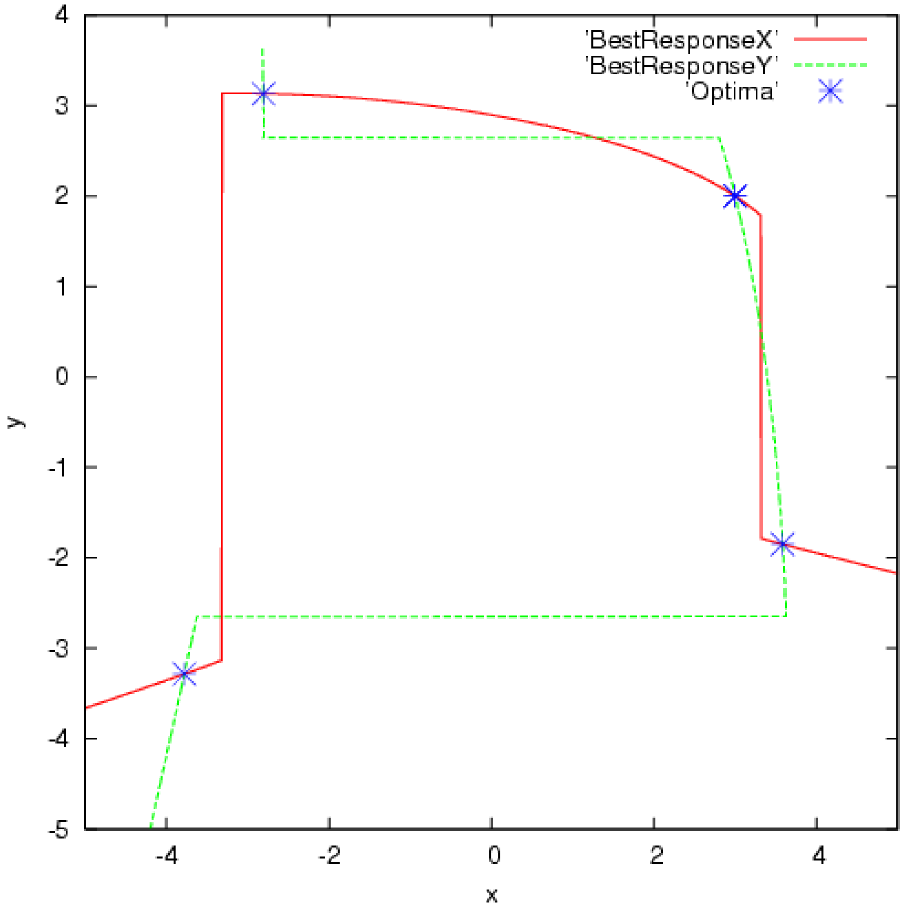


Figure 5.6: Best response curves for Himmelblau's function

The *BestResponseX* curve is easily obtained by plotting for each *x*-value the best *y* collaboration value; conversely, the *BestResponseY* curve is obtained by plotting for each *y*-value the best *x* collaboration value. Popovici introduced the concept of best response curves (Popovici and De Jong 2004) to analyse the convergence dynamics of competitive coevolution, but has since observed that the behaviour of CCEAs are heavily influenced by the best response curves of the application domain (Popovici and De Jong, 2005), (Popovici and De Jong, 2005b). In this instance the best response curves are plotted to identify complications that NCCEAs may face when optimising multimodal functions.

By decomposing the objective space of Himmelblau's function naturally into two components, the coevolutionary algorithm alternately freezes and explores the *x* and *y* dimensions of the search space. As illustrated in figure 5.6, pairs of optima are positioned along approximately (although not exactly) the same line. This topographic structure is evident from the large transitions in response curves as the position of the maximum collaborator swings between near and far peaks.

When a subpopulation is partitioned into separate species, there is no guarantee that individuals representing separate peaks are correctly divided into separate species. If there is sufficient separation between individuals representing peaks separated by a transition line, cluster analysis will be able to correctly partition the subpopulation in to corresponding species. However, if the separation is blurred, it is possible that individuals representing both peaks are incorrectly assigned to the same species. In the event that individuals *are* incorrectly assigned, a peak may be lost and individuals belonging to an incorrectly allocated cluster may converge to an already occupied peak. The close proximity of the peaks along the same line is better visualised using a *maximum-fitness* curve as provided for the *x* parameter in figure 5.7. Notice two pairs of peaks separated by a broad valley; individuals positioned at peaks either side of the valley may be easily differentiated; however, representatives of the closely positioned peaks may not.

In terms of niche maintenance this search space property is problematic for the NCCEA architecture, and in the most extreme case, when peaks are positioned along exactly the same projection, there is no way to differentiate between peaks at all. This property will be investigated further in the following section.

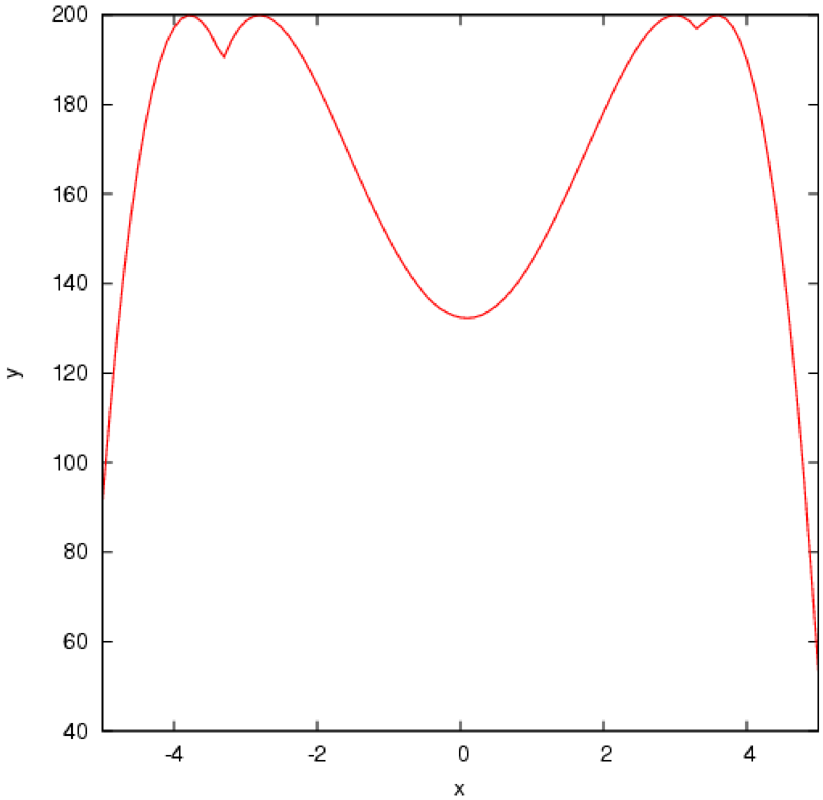


Figure 5.7: Maximum-fitness curve for x dimension for Himmelblau's function

Improving Performance on Himmelblau's Function

To ameliorate this problem, subpopulation sizes may be increased (as exemplified by the results in table 5.1 and figure 5.5) or, alternatively, the number of species may be specified in excess of the number of peaks. Table 5.2 and figure 5.8 show the results when each subpopulation is divided into first five and then six species, with subpopulation sizes adjusted to maintain a cluster cardinality of five.

					Himmelblau's Function		
		Algorithmic Parameters				Optima out of 4	
Algorithm		*(strat)*	*recomb*	*clusts*	Total out of 30	mean	σ
CCCES	1	(25,175)	*centroid*	*5*	30	4	0
	2	(25,175)	*discrete*	*5*	25	3.83	0.38
	3	(25+175)	*centroid*	*5*	29	3.97	0.18
	4	(25+175)	*discrete*	*5*	30	4	0
	5	(30,210)	*centroid*	*6*	30	4	0
	6	(30,210)	*discrete*	*6*	30	4	0
	7	(30,210)	*centroid*	*6*	30	4	0
	8	(30,210)	*discrete*	*6*	30	4	0

Table 5.2: Results of CCCES on Himmelblau's function with increased cluster quantity

For relatively small population sizes there is a notable improvement in performance with five clusters and even more so with six. By clustering each subpopulation into more species, the likelihood of assigning individuals representing distinct peaks to the same cluster is reduced.

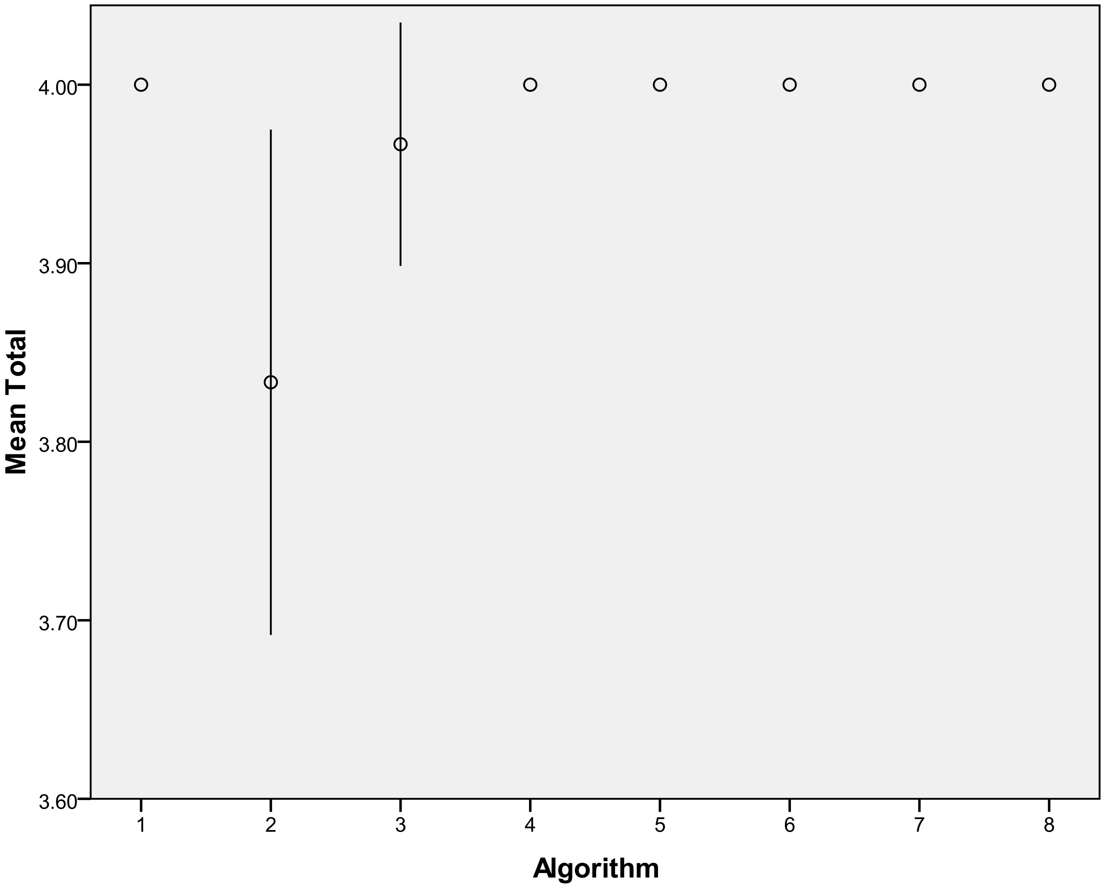


Figure 5.8: Mean and 95% confidence intervals for Total results on Himmelblau's function

However, due to the line search characteristics of cooperative coevolution, reliable location of multiple optima along the same projection cannot be guaranteed. By adopting the exclusive linkage procedure (defined in section 5.3.1), it can be ensured that, for each link, the peak in the region of the highest scoring collaboration is maintained. This behaviour is further explored in the following section, in which the CCCES is applied to the multimodal function, an environment in which many peaks are positioned along the same projection.

5.4.2 Experiments on the Multimodal Function

The multimodal function presents a search space in which there are many local-suboptima and one optimum positioned at the corner of the search space. Further details of the test function have already been provided in section 4.3.2.1. As the multimodal function exhibits many in-line optima, the CCCES is expected to consistently locate the global optimum, and then allocate species to the next highest peaks that are not in line with the global optimum.

For the experimental results that follow, the CCCES is repeatedly applied to the two-dimensional multimodal function with an increasing number of species in each test case. To maintain consistency, the population sizes are adjusted to ensure that the cluster cardinality is maintained at approximately five. As in previous experimentation, the problem space is decomposed naturally, such that each parameter is explored by a separate population. Each test case comprises an average of 30 runs executing for 50 generations. Performance is indicated using three measures:

- the *Global* optimum location performance: the number of runs, in which the optimum is successfully located.
- the *Optima* location average: The number of separate optima located, averaged over all runs.
- the average *Sum* of all peaks located: The average fitness of all located peaks. Indicated values are the equivalent maximisation figures, larger values are therefore preferred.

					Multimodal Function				
		Algorithmic Parameters				Optima out of 25		Sum	
Algorithm		*(strat)*	*recomb*	*clusts*	Global out of 30	mean	σ	mean	σ
CCCES	1	(5,35)	centroid	*1*	15	1.00	0.00	0.95	0.07
	2	(5,35)	discrete	*1*	25	1.00	0.00	0.99	0.03
	3	(5+35)	centroid	*1*	30	1.00	0.00	1.00	0.00
	4	(5+35)	discrete	*1*	29	1.00	0.00	1.00	0.01
	5	(10,70)	centroid	*2*	26	2.00	0.00	1.78	0.11
	6	(10,70)	discrete	*2*	30	2.00	0.00	1.79	0.11
	7	(10+70)	centroid	*2*	30	2.00	0.00	1.87	0.00
	8	(10+70)	discrete	*2*	30	2.00	0.00	1.83	0.07
	9	(15,105)	centroid	*3*	30	3.00	0.00	2.24	0.17
	10	(15,105)	discrete	*3*	29	3.00	0.00	2.21	0.16
	11	(15+105)	centroid	*3*	29	3.00	0.00	2.34	0.28
	12	(15+105)	discrete	*3*	30	3.00	0.00	2.23	0.24
	13	(20,140)	centroid	*4*	30	4.00	0.00	2.66	0.21
	14	(20,140)	discrete	*4*	30	4.00	0.00	2.69	0.14
	15	(20+140)	centroid	*4*	29	3.97	0.18	2.50	0.26
	16	(20+140)	discrete	*4*	30	2.62	0.24	4.00	0.00
	17	(30,210)	centroid	*5*	30	5.00	0.00	2.97	0.22
	18	(30,210)	discrete	*5*	30	4.97	0.18	2.93	0.18
	19	(30+210)	centroid	*5*	29	5.00	0.00	2.65	0.16
	20	(30+210)	discrete	*5*	30	5.00	0.00	2.71	0.17
	21	(50,350)	centroid	*10*	30	5.40	0.50	3.30	0.50
	22	(50,350)	discrete	*10*	30	5.20	0.48	3.08	0.39
	23	(50+350)	centroid	*10*	30	8.13	0.97	4.09	0.71
	24	(50+350)	discrete	*10*	30	7.90	1.16	4.03	0.66
	25	(75,525)	centroid	*15*	30	5.50	0.51	3.38	0.43
	26	(75,525)	discrete	*15*	30	5.40	0.50	3.24	0.43
	27	(75+525)	centroid	*15*	30	10.03	1.75	4.87	0.80
	28	(75+525)	discrete	*15*	30	10.20	1.40	5.08	0.66

Table 5.3: Results of CCCES on the multimodal function

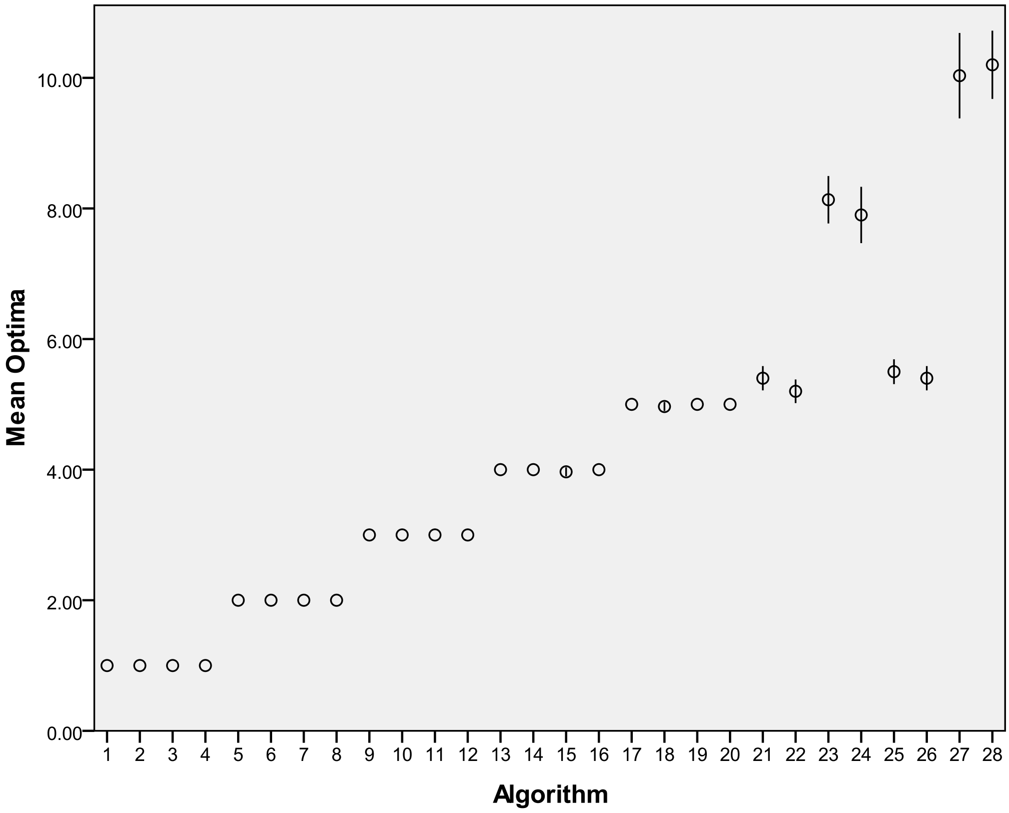


Figure 5.9: Mean and 95% confidence intervals for Optima results on the multimodal function

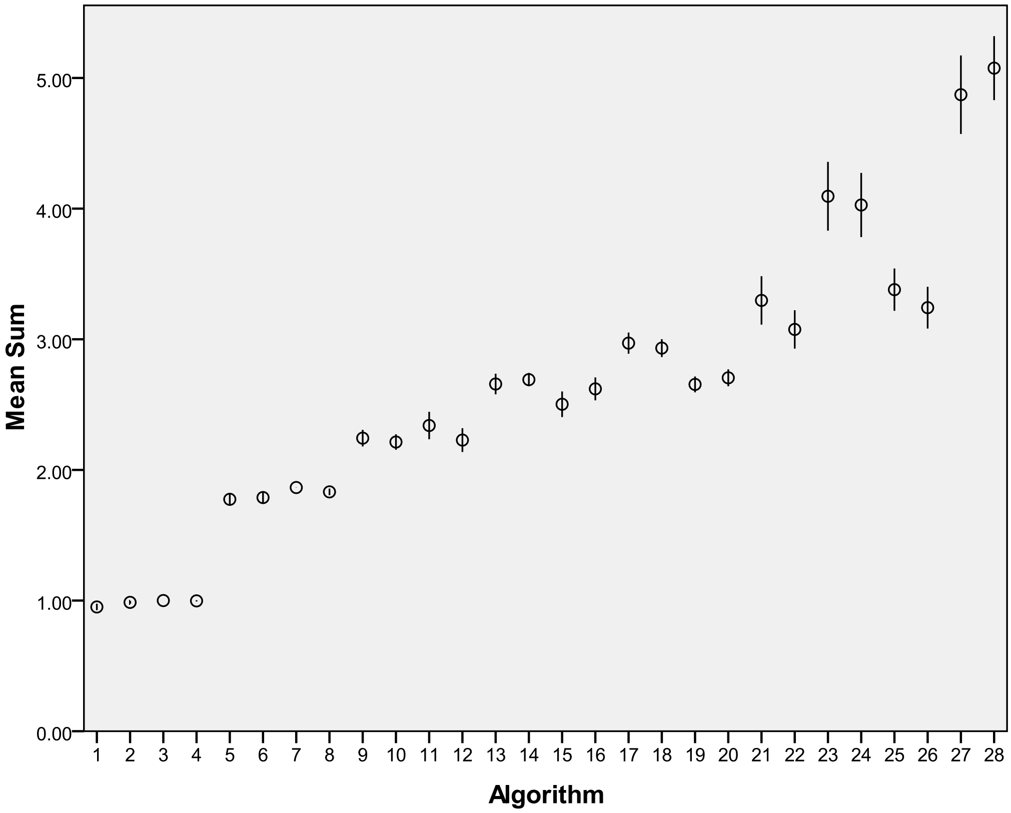


Figure 5.10: Mean and 95% confidence intervals for Sum results on the multimodal function

From the results shown in Table 5.3 and plotted in figures 5.9 and 5.10, it is evident that increasing the number of species beyond five does not result in a proportional increase in located optima. This is because there are rows of five peaks exactly aligned along each dimension of the problem space. As established in section 5.4.1, it cannot be guaranteed that more than one peak will be identified when multiple peaks lie along the same projection due to the search characteristics of the CCCES. Therefore, for the multimodal function, it cannot be guaranteed that more than five peaks in each run will be located. As predicted, the CCCES is excellent at locating a single optimum along each line of peaks the space. In the final set of results (algorithms 25-28), populations are divided into 15 species, yet, with extinctive selection, the average number of located peaks is only slightly greater than five. However, when elitist selection is employed, the number of maintained peaks is increased – an equality of means independent t-test on the results of algorithms 25-28 revealed a statistically significant (two-tailed) increase of 4.67 peaks with the elitist selection operator.

Maintenance of more than five peaks indicates that multiple species have been assigned to the same niche, but are linked differently than their cohabitating species. In most runs in which the number of clusters is equal to or greater than five, the located optima are positioned along the $x = y$ diagonal through the search space, as illustrated in figure 5.11.

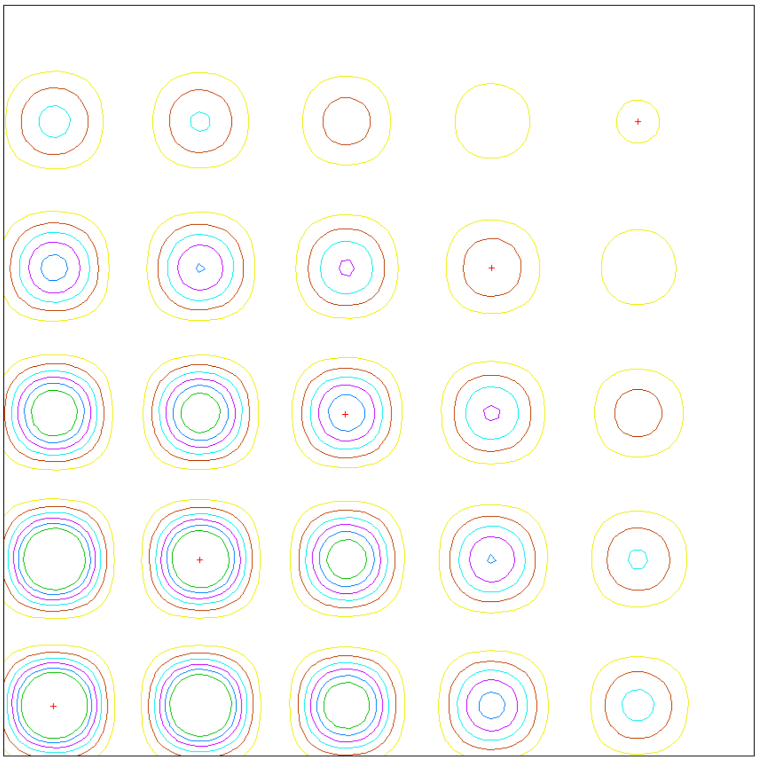


Figure 5.11: Contour plot of the multimodal function indicating final CCCES solutions

Although the located peaks do not represent the most optimal five peaks (although one of them is the global optimum), they do represent the five most optimal peaks that are not in line with any larger peaks already located by the algorithm. The CCCES thus searches for a set of diverse peaks that includes the global optimum.

n-Dimensional Multimodal Space

The examination of the CCCES is now extended to the n-dimensional case of the multimodal function. Throughout all test cases five species per population are maintained as the CCCES is repeatedly applied to the multimodal function over an increasing search space dimensionality. Each dimension naturally introduces a new subpopulation to the system. Results are provided in Table 5.4, in which the dimensionality of the function is indicated as n, and number of peaks within the space, q_F. Means and confidence intervals for the Optima and Sum measures are plotted in figure 5.12.

						n-dimensional Multimodal Function				
Search Space			***Algorithmic Parameters***				Optima out of q_F		Sum	
***n* / q_F**	***Algorithm***		*(strat)*	*Recombination type*	cluster quantity	Global out of 30	mean	σ	mean	σ
3 / 125	CCCES	1	(25,175)	*centroid*	*5*	30	5.00	0	2.65	0.30
4 / 625	CCCES	2	(25,175)	*centroid*	*5*	30	5.00	0	2.30	0.17
5 / 3125	CCCES	3	(25,175)	*centroid*	*5*	30	5.00	0	2.12	0.11
6 / 3125	CCCES	4	(25,175)	*centroid*	*5*	30	5.00	0	1.95	0.09

Table 5.4: Results of CCCES applied to the n-dimensional multimodal function

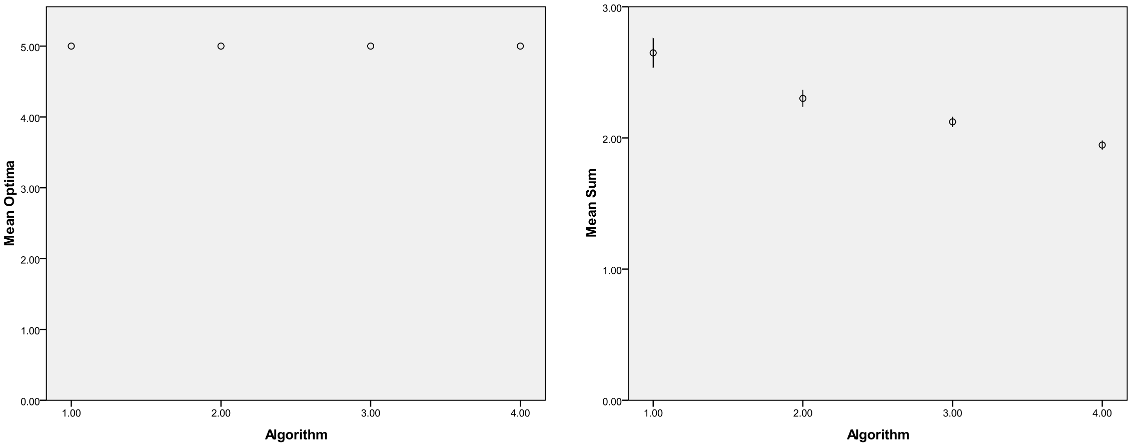


Figure 5.12: Mean and 95% confidence intervals for Optima and Sum results on the multimodal function

Discussion

As the dimensionality of the search space is increased, there is no deterioration in either the algorithm's ability to find the global optimum or locate multiple distinct solutions. However, there is a gradual, but consistent reduction in the sum measure. This indicates that, as the dimensionality of the space increases, the algorithm is less able to identify the largest optimum in each row of peaks (although it always finds the global optimum).

The linkage configuration procedure ensures that each offspring partakes in only one collaboration per fitness evaluation, equating to $p\lambda$ function evaluations per *round*. However, ps^p evaluations are also required to configure the linkage table, $p(\lambda + s^p)$ evaluations in total. If a high-dimensional search space is decomposed naturally, such that $p = n$, computation of the linkage configuration is very costly. Although significantly less costly than the $p\lambda\mu^{(p-1)}$ evaluations required when a complete mixing CCES is employed (given that $\lambda \gg s$, for the majority of experiments documented here $\lambda = 35s$). Several optimisations could be made to improve this problem. For example, the linkage configuration may be computed only once per round, or only before subpopulations have evolved to distinct species. In the latter case, neither cluster analysis nor linkage computation is necessary. However, for the work presented throughout the remainder of this thesis, the proposed exclusive linkage procedure is recalculated at the turn of each subpopulation generation.

5.4.3 Experiments on the Maximum of Two Quadratics Function

The Maximum of Two Quadratics (MTQ) function was originally developed by Wiegand (2004) to demonstrate the tendency for conventional CCEAs to optimise *robust*, rather than optimal collaborators. The proposed CCCES is principally designed to generate multiple solutions at distinct optima, but in doing so it is hoped that the effects of *relative overgeneralisation* will be lessened through the maintenance of population diversity (Wiegand and Sarma, 2004), resulting in an algorithm that is also better suited to function optimisation. Previous CCEA augmentations for improved optimal collaboration have been reviewed in section (5.2.3), in which the Maximum of Two Quadratics function is employed as a testing environment by Bucci and Pollack (2005) and Panit *et al* (2004). As demonstrated in chapter four, the MTQ function presents a particularly problematic search space for conventional CCESs, as it features one broad sub-optimal peak, and a second narrow optimal peak. The deceptive structure of the search space exasperates the relative overgeneralisation pathology, resulting in consistently poor results (see figure 4.9).

To establish whether the proposed CCCES algorithm increases the likelihood of optimal collaboration, a set of experiments are constructed to facilitate comparison between the proposed CCCES and a set of CCES variants equipped with more traditional collaboration techniques. Throughout subsequent experiments, the subpopulation size for each algorithm is fixed, and the CCES and CCCES are applied to the MTQ function with various parameter, and collaboration settings. Three variants of the CCCES are applied with the subpopulation divided into two, four and six species respectively.

For comparison, three collaboration techniques are employed for the conventional CCES:

- *CCES1* – Offspring collaborate with the best individual from the other populations.
- *CCES2* – Offspring take the highest reward in collaboration with the best and random individuals of each population.
- *CCES-μ* – Offspring take the highest reward of all possible collaborations.

For each test case, algorithms run for 50 generations and results reflect the outcome of 30 runs. Performance is tabulated for the number of runs in which the optimum peak is successfully located (*Global*), and the number of function evaluations required for a complete run of each algorithm (*Evals*).

	Algorithmic Parameters			**MTQ function**	
Algorithm	*(strat)*	*Recombination type*	cluster quantity	**Global out of 30**	**Evals**
CCES1	*(12,84)*	intermediate	-	0	8424
	(12,84)	discrete	-	2	8424
CCES2	*(12,84)*	intermediate	-	0	16824
	(12,84)	discrete	-	9	16824
CCES-μ	*(12,84)*	intermediate	-	2	100824
	(12,84)	discrete	-	19	100824
CCCES	*(12,84)*	centroid	*2*	9	8828
	(12,84)	discrete	*2*	7	8828
	(12,84)	centroid	*4*	27	10040
	(12,84)	discrete	*4*	27	10040
	(12,84)	centroid	*6*	30	12060
	(12,84)	discrete	*6*	30	12060

Table 5.5: Results of CCES (various) and CCCES applied to the MTQ function

Discussion

It was hoped that, in addition to being able to locate multiple search space optima, the modifications made to the CCEA in this chapter would also facilitate improved performance in terms of global optimum discovery. Indeed, it is clear from the results provided in table 5.5 that this is the case. The MTQ is a function that presents deceptive characteristics which are known to be problematic for the CCEA. Equipped with an adequate number of species, the CCCES is consistently able to locate the optimum with only a moderate increase in fitness evaluations. Additionally, it is important to note that in all successful runs, the sub optimal peak is also maintained.

The conventional CCES, equipped with intermediate recombination, consistently exhibits poor performance, even when the complete mixing collaboration method is employed. This is presumably as a result of the averaging effects of the intermediate recombination of all parents: the consensus is favoured over the best. As the majority of the parents occupy the local optima, their mass will draw individuals away from the narrow optimal peak. Discrete recombination improves performance somewhat, but even with complete mixing only 63% of runs are successful.

Performance of the *CCCES* with two species is relatively poor, although significantly better than the *CCES-1* with a comparable number of function evaluations. This suggests that with a similar quantity of collaborations, the CCCES extracts more useful information from the interaction space through the introduction of niching and exclusive linkage. This is further substantiated by the performance of the *CCCES* with four and six species when compared with the *CCES-*μ. With less collaborations, there is a significant improvement in performance. The tendency for the model to distribute solutions along the component space ensures that sufficient subpopulation diversity is maintained in order to locate the optimum in all runs.

In a comparable study, Bucci and Pollack (Bucci and Pollack, 2005) observed good results when applying the Pareto CCEA (pCCEA) to the MTQ function. To draw direct comparison between the algorithms is misleading as it is impossible to match the population sizes between the algorithms as pCCEA is a GA-based algorithm. However, the authors did experiment with a second MTQ function (MTQ$_2$) in which the performance of a CCEA with complete mixing was found to deteriorate further, while the pCCEA did not. The MTQ$_2$ function is identical to the MTQ function, with the exception of the H_1 parameter (see table 4.1), which is set to 125. Comparable results are achieved by the CCCES and shown in table 5.6.

	Algorithmic Parameters			**MTQ Function**	
Algorithm	*(strat)*	*recomb*	*clusts*	**Global out of 30**	**Evals**
CCES-μ	*(12,84)*	intermediate	-	1	100824
	(12,84)	discrete	-	14	100824
CCCES	*(12,84)*	centroid	*6*	30	8828
	(12,84)	discrete	*6*	30	8828

Table 5.6: Results of CCCES applied to the MTQ$_2$ function

The results for the CCCES remain unchanged, while the performance of the *CCES-MU* is reduced.

5.5 Convergence Dynamics

The CCCES was designed to ensure that the rapid convergence properties of the conventional CCES are preserved, while the robustness and niching abilities of the search are improved. To illustrate and compare convergence characteristics, the trajectory of the best collaborators is plotted at each generation for all algorithms in figure 5.13. Each curve is generated from an average of up to five successful plots from each algorithm when applied to the MTQ function with discrete recombination. The algorithms are minimising, therefore 0 represents optimum fitness.

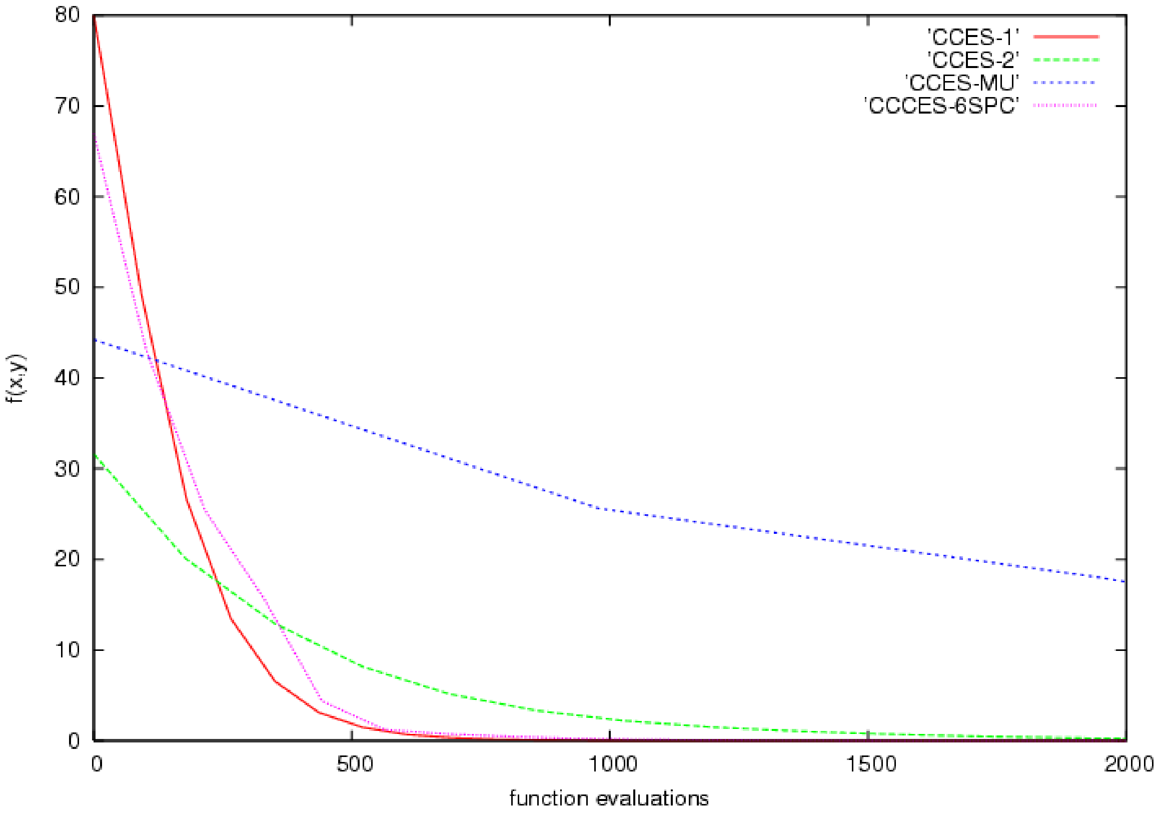


Figure 5.13: Convergence trajectories for all algorithms when applied to the MTQ function with discrete recombination

As the CCCES requires only one collaboration per offspring, the convergence velocity is comparable with the CCES-1, exhibiting faster convergence than all other CCEA variants. From the results provided in this section, it is evident that the CCCES presents a niching algorithm that is both fast and robust.

It is interesting that the results presented here show similarities to the work of Bucci. The motivation here is to locate multiple optima; however, in doing so, the optimum collaboration performance is enhanced. The work of Bucci and De Jong (2005) was motivated by the desire to achieve optimum collaboration within the CCEA and, as a side effect, the multiple solution performance was enhanced (Bucci, 2007). Perhaps these two techniques have common properties that bring about similar effects, despite the differences between them. Further comparison between these methods is beyond the scope of this work; however it is certainly a topic for future research.

5.6 Summary of this Chapter

In this chapter, the second algorithmic contribution of this work was introduced in the form of a niching cooperative coevolutionary algorithm named CCCES. A review of cooperative coevolution for parameter optimisation was provided in which a variety of alternative collaboration techniques were discussed. Relative overgeneralisation, a property of CCEAs that limits the likelihood of optimal collaboration, was also reviewed along with various techniques which have been proposed to counter its effects. The ideas from these techniques were considered and a novel cooperative coevolution algorithm was developed that implements the CES (developed in chapter four) to provide a niching cooperative coevolutionary algorithm designed to maintain multiple solutions at distinct optima, and also improve the likelihood of optimal collaboration.

An instance of the niching cooperative coevolution algorithm named CCCES was presented and examined in application to a selection of test environments. It was demonstrated that the algorithm was able to maintain multiple distinct solutions and consistently locate the global optimum, when the dimensionality of the problem domain was increased. In the final set of experiments the CCCES was applied to the maximum of two quadratics function to establish whether the effects of relative overgeneralisation had been lessened through the maintenance of subpopulation species. The novel algorithm was compared with the traditional CCES equipped with a variety of different collaboration methods, and shown to be the most reliable at locating the optimum peak, while still maintaining convergence velocities comparable with the original CCES-1.

Chapter 6

The Exploration of FM Parameter Space with Evolutionary Computation

The previous two chapters focused primarily on the design and development of niching EAs for optimising multiple distinct solutions to highly-rugged, multimodal engineering problems. The potential application domains for these algorithms are numerous, but the aim in this work is to evolve solutions to an unsupervised sound matching problem: the process of deriving parameters that cause a synthesiser to reproduce a given target sound.

The three performance attributes for which each algorithm has been tested correlate with the intended application. That is, a sound matching device that is intended to assist in relating synthesis parameters to sonic character. Each algorithmic attribute has benefits within this context:

- *Optimality* – by locating highly-fit solutions, it is ensured that derived synthesis parameters give rise to sounds that most accurately resemble their targets. That is, high fitness (optimal) solutions will represent matches at the limit of matching synthesiser's capabilities.

- *Niching* – it is desirable that multiple high fitness solutions are located, in order to provide users with a variety of match candidates, which may assist in the sound design process.
- *Dimensionality* – typically, the interface to a synthesiser presents a multitude of parameters that together specify the characteristics of the generated sound. Parameters affect the (time-varying) characteristics of the synthesised sound and, in many instances, there are complicated interactions between the parameters (epistasis). Therefore, it is important that the optimisation algorithm is able to exploit rugged search spaces that span multiple dimensions.

In chapter seven an experimental method is described which tests the ability of six ES-based optimisers to produce matches to a variety of target tones using a frequency modulation (FM) audio synthesiser. This chapter contextualises this work introducing FM synthesis and previous sound matching work.

6.1 Introduction

There is a considerable variety of synthesis architectures, each of which is capable of producing a wide range of timbres (sound characters). Often a synthesiser control interface is a reflection of the underlying synthesis process, and rarely relates to sound in human terms. As a consequence of the complex mapping between the dimensions of the synthesis parameter space and the perceived sound space, synthesiser control is often unintuitive and difficult to learn. Synthesiser users/programmers would benefit from a procedure that aids the process of mapping timbral qualities on to synthesis parameter values.

Previous studies have attempted to establish more intuitive synthesis control with the development of a timbre-based lexicon (Johnson, 2006), or a parametric representation of timbre space[7] (Nicol *et al*, 2004). Further examples include Vertegaal and Bonis (1994) and Ashley (1986), in which synthesis parameters were ascribed timbre labels that described their perceived effect, and Miranda (1998) in which subtractive synthesis parameter settings were represented by a set of non-numeric attributes.

Many of these control techniques attempt to map timbral attributes directly onto the parameters of the associated synthesis technique. The control technique is therefore inextricably linked with the form of the underlying synthesiser. If the process could be

[7] A multidimensional space in which perceptually similar sounds are positioned close together (Grey, 1978).

divided such that the control interface targets a generic intermediate representation of sound, the control domain could be detached from the synthesis domain. Thus, any control technique that is compatible with the intermediate sound representation might be used to control any synthesis technique for which parameters may be derived from the intermediate representation. This system is shown in figure 6.1.

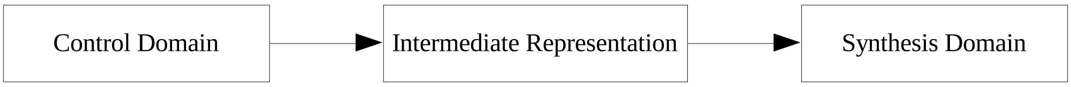


Figure 6.1: Control to synthesis mapping via intermediate representation

Etherington and Punch (1994) developed a system that resembled the first part of this control process by enabling separate temporal regions of an additive synthesised sound to be controlled via a set of *timbral features*. Additive synthesis is frequently referred to as Fourier synthesis, as the sound synthesis parameters reflect the spectral form of the synthesised sound. Since different timbres may be identified by spectrum analysis (Roads, 1996), the spectral representation of sound is considered sufficiently generic to act as the intermediate representation of the system depicted in figure 6.1. A search procedure is then required that can efficiently locate specific spectral (and thus timbral) attributes within the synthesis parameter space.

Etherington and Punch note in their conclusion that, in order to apply the system to alternative synthesis architectures, a translation method is required that is able to map the additive synthesis parameters produced by their system onto the parameters of other synthesis techniques. This translation method forms the second part of the control process and represents the procedure that is explored within this thesis: the mapping of spectral information onto the parameters of a non-additive synthesis type (FM). By viewing the first part of the process as specifying a target spectral profile, the second part leads naturally to the idea of sound matching: searching for parameter values which cause the synthesiser to create a sound that matches the target profile. As the development of an intuitive synthesiser interface is beyond the scope of this work, target profiles here are constructed from the frequency analysis of pre-existing sounds, enabling algorithm success to be determined by comparison between target sounds and their corresponding matches.

6.2 Synthesiser Choice

Since the focus of this work is the derivation of parameters for standard synthesis techniques, it is necessary to choose one with which to work. Frequency Modulation (FM) synthesis has been selected for multiple reasons:

- FM synthesis presents a method for generating sound which has seen wide application in commercial systems, and thus represents a real-world synthesis technique.
- since its introduction, there have been many efforts in which FM synthesis has been employed to simulate specific sound types; see for example Schottstaedt (1977), Delprat (1997) and Risberg (1980). This provides a historical context for the sound matching problem.
- the synthesis space is *non-linear*. A synthesis model is considered to be non-linear when the perceived timbre does not change in a consistent and proportional manner as the synthesis parameters are varied; there is a complex parameter space mapping, as described earlier. For example, the linear incrementation of a single parameter may cause a sound to move through many dimensions of the timbre space with a complex trajectory. Moreover, this trajectory may be entirely different when other synthesis parameters are changed. For fuller description of these issues see Ashley (1986).
- with only a limited number of parameters, it is possible to generate a wide range of complex time-varying sound textures with as little as two sinusoid calculations, two multiply and one addition operation for each synthesis sample (Roads, 1996). The FM synthesis model is compact and efficient.

6.3 Frequency Modulation Synthesis

FM audio synthesis provides a neat synthesis method by which complex sound forms can be created simply and efficiently. This section will further expand the method, reviewing developments since Chowning’s (1973) original simple FM model which was reviewed in section 1.1.2 of this thesis.

6.3.1 FM Extensions

Schottstaedt (1977) developed two extensions to Chowning’s simple FM arrangement in which the carrier wave is modulated by a *complex* (i.e. non-sinusoidal) modulating wave. In the first extension, called double-modulator FM, the carrier frequency is modulated by the sum of two sinusoids:

$$e = Asi\,n\big(\omega_c t + I_{m1} sin(\omega_{m1} t) + I_{m2} sin(\omega_{m2} t)\big) \tag{6.1}$$

where I_{m1} and I_{m2} represent the modulation indices of each modulating sinusoid, and ω_{m1} and ω_{m2} represent their angular frequencies. With double-modulator FM, the spectrum that results is as though each partial produced by the modulation of the carrier by one modulating oscillator, is modulated by the second. In other words, each partial produced by the modulation of the carrier by the first modulating oscillator exhibits its own side frequency partials as if modulated individually by the second modulating oscillator. For the mathematical expansions of the double-modulator FM spectrum, in terms of the Bessel Functions, the interested reader is referred to LeBrun (1977).

The second of Schottstaedt's expansions, nested-modulator FM, operates by modulating the instantaneous frequency of the carrier with a sinusoidal oscillator, which is itself modulated by a second modulating oscillator; given by the equation:

$$e = A\sin\big[\omega_c t + I_{m1} sin\big(\omega_{m1} t + I_{m2} sin(\omega_{m2} t)\big)\big] \tag{6.2}$$

The sidebands that result are identical to simple FM, only each side frequency partial exhibits its own set of side bands with partials distributed at frequency intervals of the second modulating oscillator frequency. The trigonometric expansion of the nested-modulator spectrum can be found in Horner (1998).

From the models presented by Schottstaedt, the FM model may be extended inductively to form any combination of nested and/or parallel arrangements. Nonetheless, while the author has experimented previously with both the nested and *multi*-modulator synthesis models (Mitchell and Sullivan, 2005), the work presented here uses a standard expansion of the most fundamental FM synthesis form, in which multiple simple FM elements are accumulated in parallel, see section 7.1 for further discussion of this synthesis model. This arrangement is shown in figure 6.2.

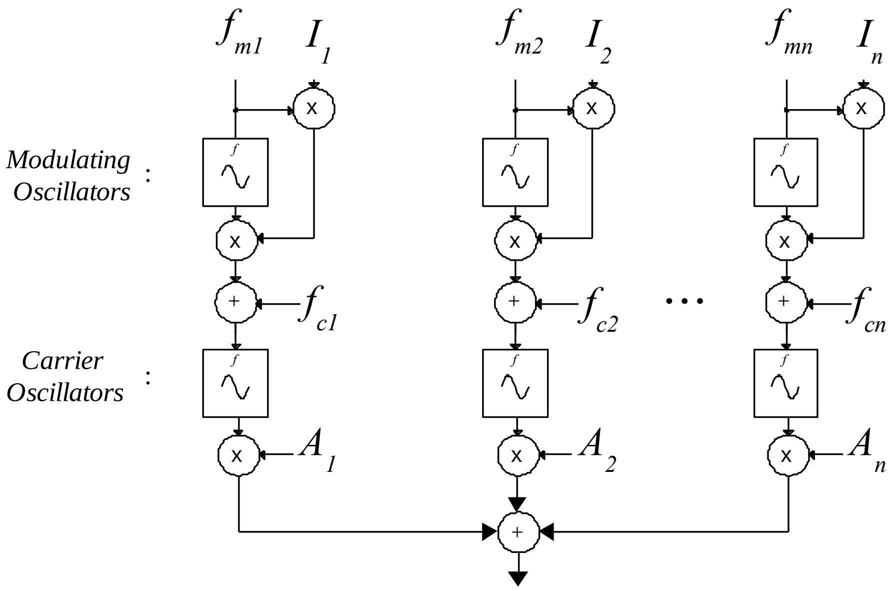


Figure 6.2: Parallel Simple FM model

6.3.2 Target Matching with FM Synthesis

Due to the non-linear relationship between the FM synthesis parameters and the spectral shape of the synthesised sound, it is often difficult to achieve desired target sounds through manual control of the parameters (without extensive mathematical analysis). This drawback has been noted by numerous researchers:

- *"It is difficult to control the shaping of the sound"*

  Kronland-Martinet *et al* (2001)

- *"It is notoriously difficult to make a sound like a given instrument"*

  Horner (2003)

- *"It is not easy to determine the values of the synthesis parameters"*

  Delprat (1997)

The desire to achieve 'natural' tones with FM space has motivated a series of studies intent on providing a systematic means by which FM synthesis can be employed to simulate real acoustic instruments. Chowning's original paper initiates interest in this direction, providing example parameters that simulate brass, woodwind and percussive tones with the simple FM architecture. Schottstaedt (1977) later provided example parameters for simulating stringed instruments, including piano and violin tones. Subsequently, many researchers set out to develop a system to automatically derive FM synthesis parameters to reproduce particular target sounds.

One of the earliest attempts to automate sound design with FM was proposed by Risberg (1980). Risberg’s system employs a stochastic filtering technique to decompose target sounds into FM synthesis parameters. The technique is effective, provided the target sound is simple and not rapidly changing. However, at the end of the paper, Risberg notes that matching complex sounds is beyond the capabilities of the system, stating also that it would not be possible to expand the model to support more complex FM arrangements.

An alternative target matching system for FM was developed by Justice (1979), in which parameters were derived using a phase-analysis procedure based on the Hilbert transform. To verify the success of the algorithm, Justice presented some successful experimentation retrieving parameters that reproduce *contrived* FM target signals (matching sounds that originate from within the synthesis space). However, the procedure is specific to nested-modulator FM, works only sounds that develop slowly and does not make allowances for reflected side-frequency partials. Some theoretical analysis is provided for the application of the model to general signals, but further experimentation is left as future work. Justice suggests that such a system may provide users with a means of jumping to approximate regions of the synthesis sound space, leaving finer adjustment to be performed by hand.

Justice’s analytical process was later extended by Payne (1987) to process multiple (parallel) carrier nested-modulator FM arrangements. The paper outlines numerous restrictions to which the target signal should conform, but, even when all constraints are met, the process is not always successful. A comparable technique for FM parameter decomposition, called FM *law extraction*, was also proposed by Delprat (1997). FM law extraction estimates synthesis parameters by interpreting formations in the Gabor transform coefficients of the target signal. Experimental results are presented that show promising partial interpretation of the coefficients. However, the system is not complete and, like many of the procedures outlined in this section, full development is left as future work.

More recent advances in sound matching with FM synthesis have used evolutionary algorithms to optimise synthesis parameters. This work is reviewed in section 6.4.4 of this chapter, after related research in which EC has also been applied within the context of sound synthesis has been discussed.

6.4 Sound Synthesis Applications of Evolutionary Computation

Research at the intersection of artificial intelligence and sound synthesis has produced a collection of studies in which EC has been used to provide more intuitive synthesis control. This is generally in one of two forms: interactive evolution, in which the user controls the direction of the search as evolution takes place; or autonomous sound matching, where the evolutionary search explores the space, without supervision, to find a close match to a given target sound.

6.4.1 Interactive Evolutionary Synthesis

Interactive evolution is a method of search in which human evaluation is included within the reproductive cycle of a conventional EA; a procedure exemplified by Dawkins' (1986) *biomorph* software for evolving complex two-dimensional image structures. Population members are selected based upon the user's subjective preferences and recombined to create offspring. Such a model is easily augmented to evolve sounds rather than images, and this is the approach adopted in several studies. Interactive evolutionary synthesis enables users to explore complicated synthesis spaces without the need for expert knowledge of the underlying system. Examples in the literature include Johnson (1999, 2003) in application to granular synthesis; Yee-King (2000) for evolving or growing synthesis structures; and Mandelis (2002, 2003a, 2003b, 2004), and Dahlstedt (2001) for exploring the sound space of commercial synthesisers.

6.4.2 Evolutionary Sound Matching

There are many benefits to the interactive model outlined above. However, the method is not without its drawbacks. The progress of the search is significantly impaired by the bottleneck of user evaluation. As each offspring must be examined independently by hand, the rate at which generations elapse is low, and the population size must be significantly constrained compared to an autonomous system. Furthermore, the interactive procedure does not provide an efficient method for directly achieving specific sound timbres, should a user have a clear requirement for the type of sound that they wish to create. Non-interactive evolutionary sound matching attempts to automate the search procedure by enabling a target sound to be specified by the user. The matching system then sets about locating a synthesis parameter set that best approximates the characteristics of the target.

The earliest evolutionary sound matching systems were presented by Horner for matching sounds produced by real acoustic instruments with FM and wavetable synthesis[8]; see for example Horner *et al* (1993a) and Horner *et al* (1993b) respectively. Horner has since focused primarily on wavetable matching techniques; a comprehensive summary of this research can be found in Horner (2003).

6.4.3 A Conspectus of Non-FM Evolutionary Sound Matching Research

Wavetable Synthesis - For matching brass, string and voice sounds, Horner *et al* (1993b) and Horner (1995) used a GA to select a small number of spectral snapshots (wavetable basis spectra), taken from the original sounds, that may be combined to accurately reproduce the entire target sound. Results in this study were presented in the form of spectrum error curves, and time/frequency plots to provide visual comparison between the target and corresponding matched sounds.

Plucked String Physical Modelling Synthesis - Riionheimo and Välimäki (2003) applied EC to match target sounds using a plucked string physical model, defined originally by Karjalainen *et al* (1998). The nine-dimensional parameter space of this synthesis model was contracted by encoding parameter ranges into discrete steps. Significant knowledge of the model was applied in this process to ensure that the parameter intervals were set just below the threshold of perceptual discrimination. A GA was then employed to optimise the synthesis parameters to reproduce three target sounds, two of which originated from the matching synthesis model itself (contrived targets), and a third sound recorded from a real string instrument. Good results were reported and visualised with time and time/frequency plots.

Alternative Synthesis Structures – EAs have also been employed to evolve and grow modular synthesis circuits. For example, Wehn (1998) evolved the arrangement and interconnection of different signal elements (oscillators, noise generators, filters, etc) to reproduce target sounds. Synthesis arrangements were represented as graphs, the structure of which was varied and optimised by GA. In a related system, Garcia (2000, 2001, 2002) evolved representations of synthesis models, referred to as *expression trees*, by Genetic Programming (Koza, 1992). Matches were performed using piano and FM target sounds

[8] For fuller treatment on wavetable synthesis see the work of Bristow-Johnson (1996)

with results again presented using time and frequency visualisation.

6.4.4 Evolutionary Sound Matching with Frequency Modulation Synthesis

In addition to matching instrument tones with wavetable synthesis, Horner also studied the application of GAs to FM synthesis parameters for the same purpose. Horner's (1993a) FM matching algorithm optimises a set of static basis-spectra, produced by FM synthesis, which are dynamically recombined to simulate time-variant harmonic sounds. The amplitude envelopes for the basis-spectra are then determined by a direct least-squares solution. The synthesis process is thus equivalent to wavetable synthesis, with FM used only in the production of basis-spectra.

The wavetable basis-spectra are generated by a special configuration of the simple FM model, known as *formant* FM, in which the modulator frequency is tied to the fundamental frequency, and the carrier frequency is restricted to integer multiples thereof. Restriction of the carrier frequency to integer multiples of the modulating frequency ensures that only harmonic basis-spectra are considered (see section 1.1.2).

Horner's hybrid FM/wavetable synthesis model provides a means by which dynamic sounds may be generated from static FM spectra. However, FM arrangements that synthesise dynamic sounds have existed for many years. A simple model is provided in Chowning's original paper, which formed the precursor to the models implemented by commercial FM synthesiser manufacturers. Consequently, Horner's model cannot be applied directly to explore the sound space of conventional FM synthesisers as it adopts an alternative synthesis paradigm. Nonetheless, excellent results were achieved when matching certain acoustic instrument tones with results provided in the form of error and spectral envelope plots.

Later, Horner (1998) extended his FM/wavetable model by employing the larger double- and nested-modulator FM arrangements to generate matches to trumpet, tenor voice and Chinese pipa sounds. Again, good results were achieved and presented in terms of error measures and spectral envelope plots. Also included in this study was a simple comparative listening test, in which five participants were asked to differentiate an equal number of synthetic matches from their real targets. For all three sounds, indistinguishable matches were achieved when synthesis models constructed from between three and five double- or nested-FM arrangements were used to generate each basis spectrum.

Tan and Lim (1996) developed a novel EA based on the GA with a modified crossover operator that incorporated a simulated annealing procedure. The Genetic Annealing Algorithm (GAA) was applied to match a selection of static tones taken from real acoustic instruments with a basic double-FM (DFM) (Tan *et al*, 1994) synthesis model. Results were presented in the form of spectrum error results, indicating a performance advantage of the GAA when compared with the equivalent results produced by simple GA.

Tan later extended his model to facilitate the matching of dynamic sounds (Lim and Tan, 1999). Time-varying sounds were treated as a time extension of the static tone matching procedure outlined above (Tan and Lim 1996). A static match was performed for the frequency spectrum extracted from each timeframe of the target sound. The accuracy of the model when applied to five acoustic target sounds was calculated using the mean average of the spectrum error at each frame. Exactly how the sequential static matches were blended to reproduce the target sound is not indicated. Also included in Lim and Tan's article is a three-dimensional plot of a sub-space of the problem domain. The topology of the search space is shown to be extensively multimodal, with significant variation in the landscape when the model is applied to match different target sounds.

In a more recent study, the author of this thesis compared the performance of several ES-based EAs for matching a set of static target tones with the simple FM model (Mitchell and Pipe, 2006). The target set consisted of tones generated by the matching synthesiser, enabling the performance of the EAs to be measured quantitatively in terms of the number of successful parameter retrievals that were achieved. The EA designed for multimodal optimisation was found to perform the best. The same testing method is employed for the experimentation documented in chapter seven.

6.4.5 New Developments in Evolutionary Sound Matching

The articles reviewed above provide confirmation that EAs can be successfully applied to assist in the process of matching target sounds with FM. However, all of these examples impose limitations on the parameter ranges of synthesis models and, for matching time-varying sounds, employ non-standard synthesis structures to make the problem space more tractable. As such, sound matching with existing time-varying FM synthesis models, employed by commercial synthesis manufacturers, has not yet been addressed in the literature. The majority of reviewed articles use a hybrid FM/wavetable synthesis form, as their intention is to produce matches which are indistinguishable from their targets. From this perspective, development of the matching synthesiser structure is a natural progression, as exact target reproduction is often beyond the capabilities of existing FM synthesis forms.

The motivation behind the work presented in this thesis differs from the existing body of work in this area. The intention here is to develop an unsupervised sound-matching tool for existing, commercially implemented FM structures, with no restrictions placed on the parameters of synthesis. The goal is to evolve the most accurate match within the limits of the synthesiser, and not matches which are indistinguishable from their target sound. Consequently, the synthesis models are not developed beyond their traditional forms here, emphasis is instead placed on the development of an improved matching technique; hence the extensive algorithmic developments already presented in chapters four and five.

The author of this thesis has previously published the results of early experiments applying a clustering EA to match dynamic-sounds produced by the matching synthesiser (Mitchell and Sullivan, 2005). These experiments involve the simple, double and nested time-varying FM synthesis models. Results were presented in terms of spectrum error results, time and frequency plots and some informal listening tests. It was determined that it is easiest to retrieve sounds generated by the simple-FM matching synthesiser, referred to as *contrived* sound.

The complexity of the objective space is ultimately linked with three components of the sound matching process:

- the synthesis model being employed.
- the method for deriving the quality of a given match.
- the characteristics of the target sound.

The following section considers several possibilities for assessing the quality of a match.

6.5 Sound Similarity Measures

To perform unsupervised sound matching, it must be possible to determine the quality of a sound simulation in the absence of human listeners. A metric is then required that indicates the 'distance' between a target sound and its synthesised match; where a good match is positioned close to its target. This distance measure may then be used to indicate fitness, enabling strong individuals to be identified, selected and bred to produce new matches.

6.5.1 Content-Based Analysis

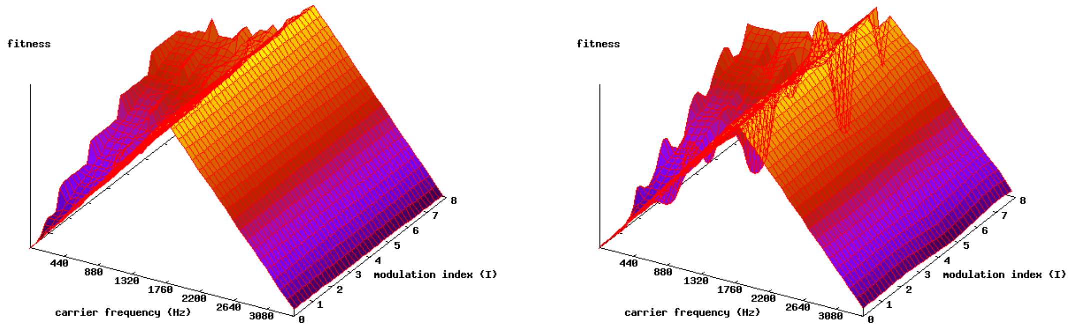
Content-based analysis of music is a rapidly growing field of research, which is likely to expand in the coming years as personal digital audio devices and computers continue to become the central medium for the access, dissemination and storage of music. There is an increasing requirement for large music repositories to be organised automatically, enabling music to be browsed by the content of the audio, rather than the appended data tags (artist name, song title, genre and so forth). A good introduction to this field has been compiled by Downie (2003, 2004), and for a comparison between a variety algorithms has been conducted Typke (2005).

Within the field of automatic music information retrieval, a large body of work is concerned with the generalisation of entire compositions into useful groups (*audio taxonomies*). This is often achieved by identifying characteristics that are shared by pieces of music that are considered similar by the subjective consensus of listeners. This process facilitates the automatic identification of genre and, in some instances, composer (Cilibrasi, 2004). However, for the purposes of sound matching, a very specific means to identify the character of short, single-voiced, quasi-periodic sounds is required.

A subset of the music information retrieval research is concerned with the identification of timbral sound features. A selection of potential metrics is listed in Burred (2004), and McDermott *et al* (2006), and examples include:

- *Zero Crossings* – the number of time domain polarity switches in a given frame.
- *Centroid* – the centre of gravity of the frequency spectrum.
- *Rolloff* – the frequency below which the majority of the spectrum is concentrated.
- *Flux* – the rate of change of the spectral form between successive time frames.
- *Spread* – the distribution of the spectrum around the centroid.
- *Flatness* – the deviation of the signal spectrum from that of a flat response.
- *Harmonic Ratio* – the proportion of harmonic partials in the signal spectrum.

While these metrics are able to successfully identify attributes of timbre, they do not provide sufficient discrimination between sounds to facilitate accurate matching. For example, two sounds may be perfectly correlated in terms of their spectral centroids, but perceptually, they may be very dissimilar. This shortcoming can be demonstrated with a simple visualisation of the FM search space. Figure 6.3 illustrates two landscape plots of a simple FM synthesiser, where target sound is chosen from within the same space (i.e. there is only one search space optimum) with the following parameters: $f_c = 1760Hz$, $f_m = 1760Hz$, $I = 4$ and $A = 1$. The parameters f_c and I are plotted against fitness, which is measured according to the zero crossings (figure 6.3a) and centroid (figure 6.3b) difference metrics as defined in (Burred, 2004); the remaining parameters (f_m and A) are fixed to match those of the target tone.



(a)

(b)

Figure 6.3: Simple FM landscapes with contrived target at fc = 1760Hz, fm = 1760Hz, I = 4.0 and A = 1.0 produced by the Zero Crossing fitness function (a), and Centroid function (b)

No clear optimum is visible in either the zero crossings or the centroid fitness landscape. There are, however, several regions within the space that do appear to be optimal. For example, both functions exhibit a strong ridge along I, which extends towards low values for f_c at higher I values. The explanation for this landscape structure is fairly intuitive if the effect of the FM synthesis parameters on the spectral form of the synthesised tone is recalled (figure 1.2, page five). The carrier frequency intuitively relates to the centre of gravity and fundamental frequency of the synthesised tone, which results in a strong ridge at $f_c = 1760Hz$ for all I. The ripples in the surface of the landscape are caused by reflected sideband partials, which are more pronounced at higher values of I.

Neither the centroid nor zero-crossings metrics provide an accurate indication of sound similarity for all parameters. This property is illustrated in figure 1.2, in which the centroid of each spectrum is identical (at the carrier frequency), when clearly the spectra are not. By focusing on individual characteristics which are known to be of perceptual significance, the metrics identified above only provide a limited measure of difference/similarity. Better results could, perhaps, be achieved by combining metrics to produce composite fitness functions (McDermott *et al*, 2005, 2006). However, preliminary experimentation by the author of this thesis with weighted metrics resulted in a considerable increase in computational expense with negligible improvements in match quality.

6.5.2 Spectrum Error

For the evolutionary matching systems presented in this thesis, sound similarity is measured by computing the relative spectral error between spectra of the target and candidate sounds. This error measure, and variations thereof, has proved effective in previous evolutionary matching studies and offers an excellent balance between detail and execution speed ; see for example the efforts of Wehn (1998), Garcia (2000), Horner *et al* (1993a and 1993b), Horner (1998) and Riionheimo and Välimäki (2003).

The relative spectral error is computed by accumulating the normalised difference between each frequency component of the candidate spectrum against their corresponding components in the target spectrum, both of which are extracted by Short-Time Fourier Transform. The error metric is defined by:

$$E = \frac{1}{N_{frames}} \sum_{t=0}^{N_{frames}} \sqrt{\frac{\sum_{b=0}^{N_{bins}} (T_{tb} - S_{tb})^2}{\sum_{b=0}^{N_{bins}} T_{tb}^2}} \tag{6.3}$$

where E is the relative error, T is a vector of target spectrum amplitude coefficients, S a vector of synthesised candidate spectrum amplitude coefficients, N_{frames} the number of static spectra analysed over the duration of the sound and N_{bins} the number of frequency bins produced by spectrum analysis. A relative error of zero indicates an exact match, and comparison between the target sound and silence results in an error of 1.0. A match that achieves an error of 0.1 indicates an average difference of 10% between the target and candidate spectra. Studies performed by Beauchamp *et al* (Beauchamp and Horner, 2003), (Horner *et al*, 2006) with acoustic musical instrument sounds have established that the relative spectral error delivers the best correspondence to average discrimination data extracted from human listeners, when compared with alternative measures of spectrum error. Furthermore, when the relative error was calculated using less than 10 frames of each sound, the correlation compared favourably with those attained when the entire frame set was used.

Unlike the metrics considered in section 6.5.1, the relative spectrum error is able to indicate when an exact match is achieved. However, this increased detail results in a significantly more rugged landscape than those depicted in figure 6.3; an equivalent landscape plot for the relative spectrum error is provided in figure 6.4. This landscape plot was created using exactly the same target sound as was used to produce figures 6.3a and b. Despite the same target sound, the landscape characteristics are significantly different.

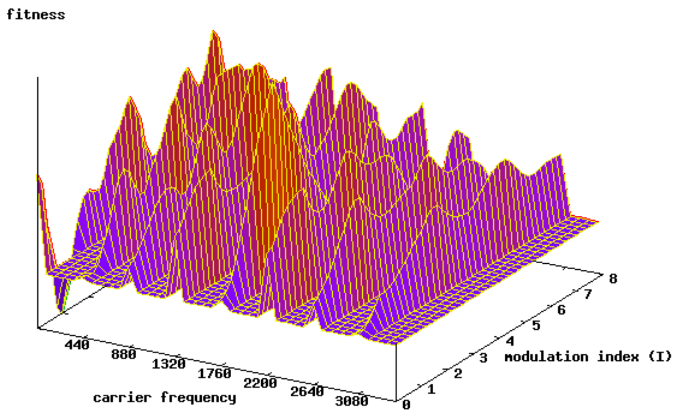


Figure 6.4: Relative spectral error landscape with contrived target at f_c = 1760Hz, f_m = 1760Hz, I = 4.0 and A = 1.0

The landscape comprises a series of peaks and troughs running along the modulation index. This is because partials in the candidate spectrum only result in a fitness improvement when they exactly coincide with partials found in the target spectrum. This property is illustrated in figure 6.5.

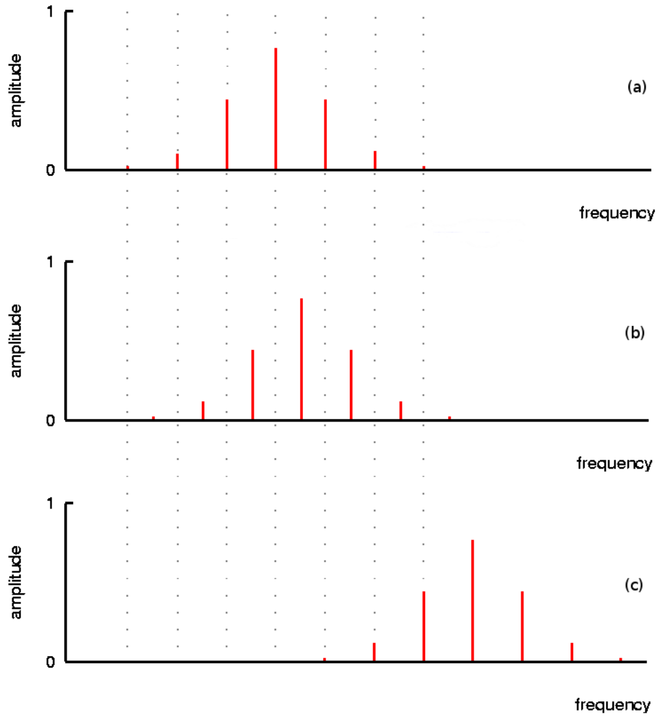


Figure 6.5: Frequency spectra of three FM tones

In figure 6.5, the three FM tones (a), (b) and (c) are identical in all parameters apart from the carrier frequency ω_c. If (a) represents the target spectrum, it might be considered that the spectrum (b) is closer to the target than that of tone (c). However, the metric defined by equation 6.3 will classify spectrum (c) to be a more accurate match than spectrum (b). This is because none of the frequency partials in spectrum (b) correspond exactly with the partials of the target spectrum (a), whereas, some of the outer partials of tone (c) are coincident with the target side-frequencies of (a). This effect results in the landscape depicted in figure 6.4.

Interestingly, this is a problem that was faced by Horner (1993a, 1998), but was largely avoided by restricting the carrier and modulator frequencies to integer multiples of the target fundamental frequency. This has the effect of removing the troughs in the landscape, resulting in a more tractable problem. However, as a consequence of this limitation, the majority of the FM sound space is omitted from the search, precluding non-harmonic sound matches. In this work, emphasis is placed on managing the synthesis space *as is*, modifying the search operators to work with the standard architecture, and not the other way around.

6.4.3 Windowed Relative Spectrum Error

To restrict the problems associated with frequency bin alignment a windowing function has been developed by the author that is designed to smooth the surface of the landscape by identifying when frequency components are in the proximity of, and not just directly coincident with, target partials. This function forms one of the minor contributions of this work, which was published previously in Mitchell and Pipe (2005). Prior to comparison by relative spectral error, both the target and candidate spectra are modified by the formula:

$$FW_b = \sum_{v=0}^{w-1} \frac{w-v}{w} (F_{b-v} + F_{b+v}) \quad , \quad \forall b, \qquad F_n = 0 \; when \;\; N_{bins} < n < \; 0 \qquad (6.4)$$

Where F_b represents the bth frequency bin of the frequency spectrum F, w controls the width of the proximity window (measured in frequency bins), and FW_b represents the bth bin of the windowed spectrum FW. This function allows the energy from each frequency partial to bleed into surrounding bins.

Figure 6.6 provides a plot of the landscape using the same target sound as in figures 6.3 and 6.4 with the difference measured by relative spectrum error when both the target and candidate spectra are modified by equation 6.4 with $w = 20$.

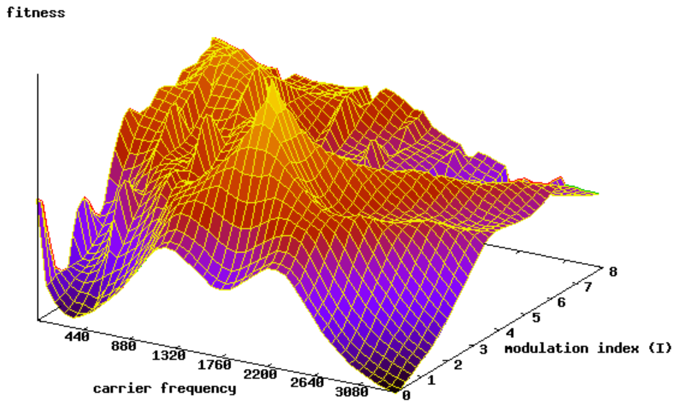


Figure 6.6: Equivalent windowed relative squared error landscape with contrived target at f_c = 1760Hz, f_m = 1760Hz, I = 4.0 and A = 1. 0

Unlike the metrics considered earlier in this section, the global-optimum can be observed as a clearly visible peak within the landscape. The windowing function enables error to be measured across a weighted band, which has a smoothing effect on the fitness landscape. However, the optimum window size is largely dependent upon the distribution of the target frequency components and the size of the Fourier transform. As such there is no general value for w that can be recommended for matching all applications. If the window size is set too small, the corresponding search space may be too rugged to optimise and, conversely, if the window size is set too large, detail in the spectrum may be lost, which may preclude an accurate match. For the experimental work performed here, the sample rate is set at 44.1kHz with a Fourier transform size of 1024. For the forthcoming experimentation the proximity window was set to an initial value of $w = 5$, thus including a weighted sum of 2% of the frequency spectrum within each window.

Further analysis of the search space is difficult to perform, as its topology is entirely dependent on the spectral form of the target sound. However, some preliminary analysis of the FM search space has been performed by Horner (1997). Horner compared the search space characteristics of several wavetable-based synthesis methods in which the basis spectra originate from several sources, one of which was FM synthesis. Thousands of randomly generated tones from each synthesis method were compared with two target sounds produced by a trumpet and pipa to indicate the availability of good matches within each synthesis space. The accuracy of a match was determined using the relative spectral error metric (equation 6.3). Of the synthesis methods that were compared, basis spectra chosen from the FM-based models produced the least accurate simulations, with the double-modulator FM model performing worst of all. Tones matching those of the tested sound targets were least abundant in the FM sound space. These results correlate well with the results in Mitchell and Sullivan (2005) in which experimentation with the double-modulator FM model produced significantly poorer results than the simple FM equivalent.

Also included in Horner's analysis of the FM search space was a basic one-dimensional visualisation, created by plotting the relative error against the modulation index. Comparable analysis of the alternative synthesis methods produced plots which contained significantly less local optima. Horner concludes from these visualisations that a simple hill-climbing search strategy would be insufficient for successful exploitation of the FM parameter space.

6.4.4 Perceptual Error

When measuring similarity between two sounds, it is generally the case that a small relative spectral error results in a better subjective match (Horner *et al*, 2006). However, as noted by several authors, this is not always true (Horner,1997), (Wun and Horner, 2001) and (Riionheimo and Välimäki, 2003). The relative spectral error provides an analytical indication of spectral similarity and does not directly reflect the properties of auditory perception. For example, in equation 6.3 all frequency components are considered to be of equal importance; however, in terms of perception, the ear is more sensitive to frequency partials positioned at the lower range of the audio spectrum (Zwicker and Fastl, 1999). Furthermore, including the psychoacoustic effects of frequency masking within the similarity measure may result in an improved correlation between measured error and perceived error. This is an approach that has been adopted by Wun and Horner (2001) and also by Riionheimo and Välimäki (2003). However, the calculation of a perceptual error measure introduces additional complexity to the matching process with reports of 'relatively small' improvement (Wun and Horner, 2001). This thesis, therefore, considers matching experiments with only the relative spectral error as an indicator of sound similarity. Further investigation into perceptual metrics, especially their relative effects on the tractability of the synthesis search space, is an area of interest for future research.

6.5 Summary of this Chapter

In this chapter, the application domain for the FM synthesis matching problem was introduced. Included is a short introduction to FM synthesis, different FM synthesis models and arrangements, as well as a review of previous research that attempts to address the problems of locating specific sound timbres within the synthesis space. A review of work in which EC has been applied to assist in the use of sound synthesisers is also included, along with the previous sound matching research on which this work is built. Several metrics for quantifying sound similarity have been reviewed, and an extension to the relative spectral error, known as the windowed relative spectral error, was introduced.

Chapter seven develops the work reviewed in this chapter by examining the performance of multiple ES-based algorithms in application to the FM synthesis parameter estimation problem. A test method is developed that enables the performance of different optimisation algorithms to be compared by their ability to exploit the synthesis problem. To maintain a general approach to sound matching, no knowledge of the problem domain is used to derive parametric representation; all synthesis parameters may take any value within their specified range. Furthermore, the algorithms are applied to optimise the parameters of known FM synthesis forms, capable of producing time-varying sounds. This shifts the emphasis from complete and accurate sound matching, to effective synthesis space navigation. That is, the ultimate goal here is not to exactly reproduce targets but to find synthesis parameters that produce the most accurate simulation of the target, given the capabilities of the matching synthesiser.

Chapter 7

Experiments in Evolutionary Sound Matching with FM Synthesis

In this chapter, the architecture and testing method for the FM sound matching problem are formally introduced. Thereafter, the performance of six ES-based optimisers will be examined and compared in application to this problem domain. Included amongst the tested algorithms are the novel niching EAs introduced in chapters four and five of this thesis: the clustering evolution strategy (CES) and the clustering cooperative coevolution strategy (CCCES). In addition to setting forth a novel application and testing method for sound matching with EC, the experimentation in this chapter also represents a real-world testing ground for the EAs, which have been tested within only theoretical environments previously in this thesis.

Experimentation is divided into two parts. The first part considers the matching of static spectrum tones, with an FM synthesis model in which the parameters remain stationary throughout the synthesis process. The second part is concerned with matching time-varying, dynamic spectrum sounds, by allowing certain parameters to change as synthesis takes place. This terminology is maintained in this chapter, referring to timbres with a constant spectral form as static tones, and dynamic timbres in which the spectrum changes as time-varying sounds. The division of the experimental results into these two parts represents natural progression in tackling the synthesis matching problem, and corresponds directly to the chronological development of this work. Sections 7.1 – 7.4 will introduce both the static and time-varying FM synthesis models, matching algorithms and testing method employed, while sections 7.5 onwards will outline the experimental work with initially static target tones and then time-varying target sounds.

7.1 Evolutionary FM Matching System

A significant motivation for this work has been the desire to produce a sound matching procedure that may be used to match sounds with existing FM synthesisers. Three synthesis forms are therefore examined which are constructed from parallel extensions of the *simple* FM model. These configurations are depicted in figure 7.1 and follow the notation set out in chapter six (see figure 6.2, page 126).

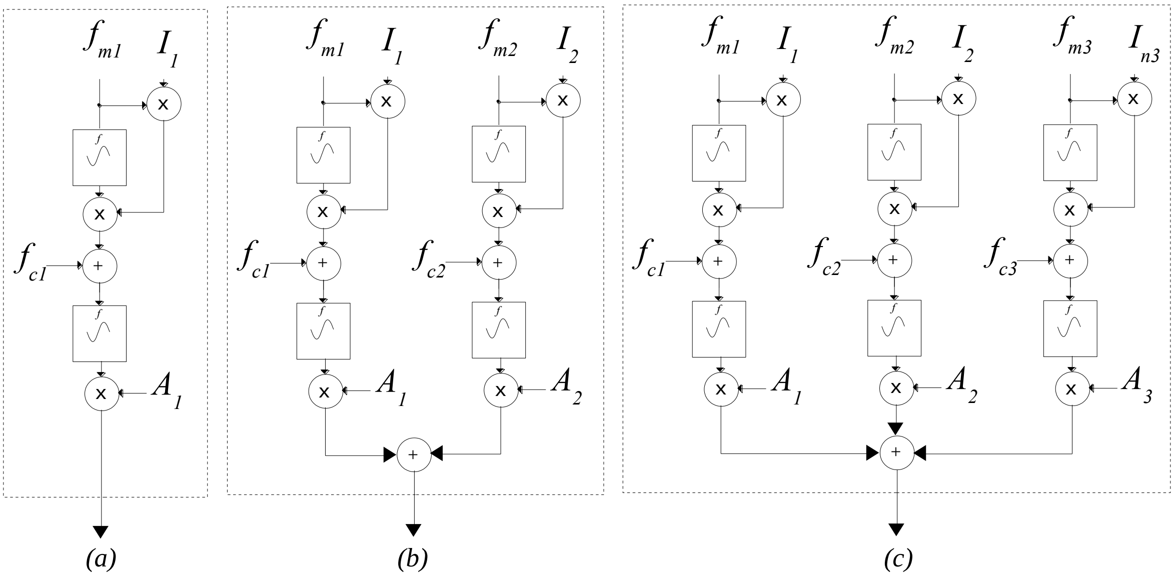
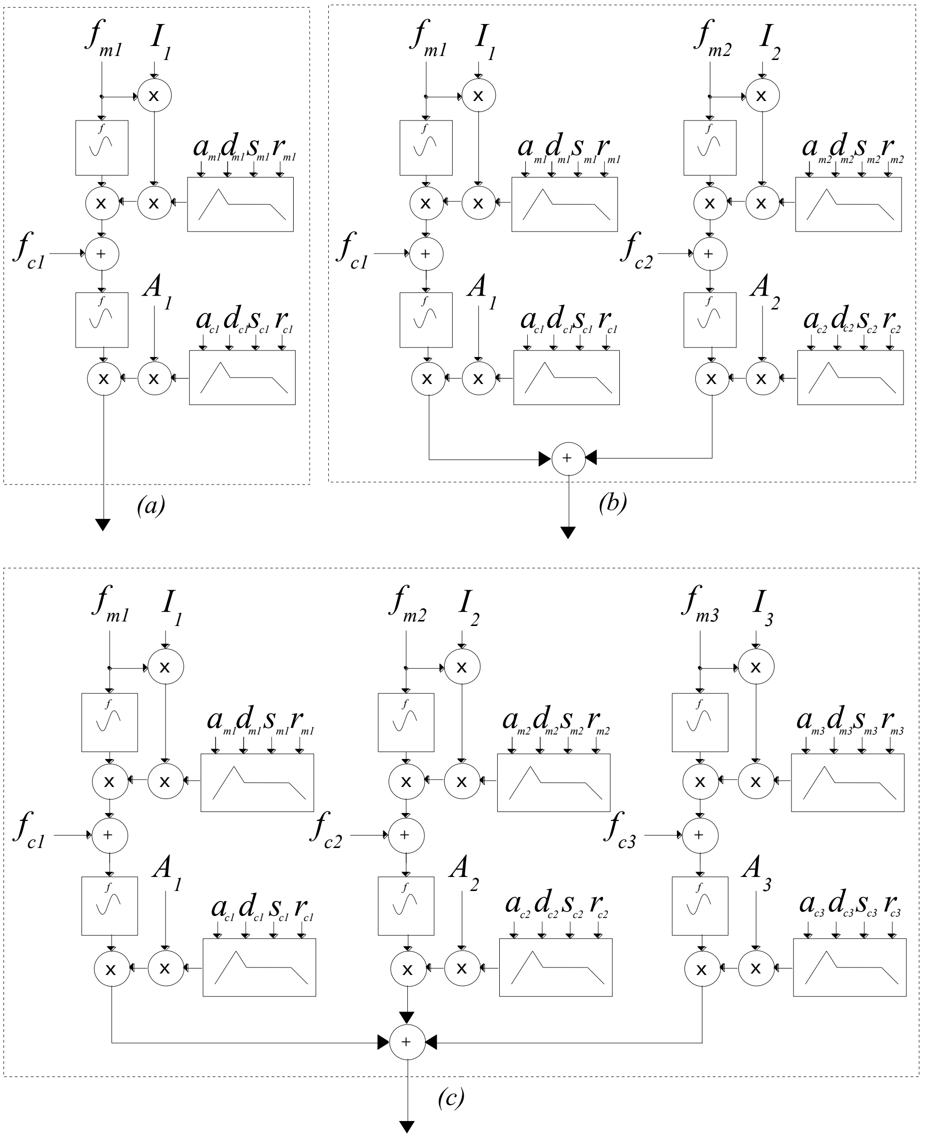


Figure 7.1: (a) single, (b) double and (c) triple parallel static simple FM arrangements

Figure 7.1 illustrates the three static FM synthesis models which will be employed in section 7.6.1. Figure 7.1a represents the simple FM model in its most fundamental form, as defined originally by Chowing (1973) in his canonical FM synthesis paper. Figures 7.1b and c represent parallel extensions, composed of two and three simple FM elements respectively. These latter models equate to the synthesis models employed by Winduratna (1998), and, more importantly, represent FM synthesis configurations available on the DX and TX series Yamaha synthesisers (without feedback) (Anon, 1987), (Massey, 1986), (Chowning and Bristow, 1986). Throughout the remainder of this work these synthesis models are referred to as single, double and triple simple FM respectively.

Each square block in figure 7.1 represents a sinusoidal oscillator, whose input parameter determines the instantaneous frequency of oscillation and which produces a wave with peak amplitude ± 1. Notice that the amplitudes of the modulating oscillators are computed from the product of the modulation index I and the modulation frequency f_m. The amplitude of the modulating oscillator controls the degree to which the frequency of the carrier oscillator is varied above and below f_c (see section 1.1.2), and is thus referred to as the frequency deviation (d). To aid clarity, complete synthesis structures are referred to as *models*, and individual simple FM structures, which are internal to each synthesis model, are referred to as *elements*. That is, each FM model is constructed from one to three parallel simple FM elements. Figure 7.2 illustrates the three time-varying FM synthesis models which will be employed in section 7.6.2.



f_{m1} I_1
$a_{m1} d_{m1} s_{m1} r_{m1}$
f_{c1}
A_1
$a_{c1} d_{c1} s_{c1} r_{c1}$
(a)
f_{m1} I_1
$a_{m1} d_{m1} s_{m1} r_{m1}$
f_{m2} I_2
$a_{m2} d_{m2} s_{m2} r_{m2}$
f_{c1}
A_1
$a_{c1} d_{c1} s_{c1} r_{c1}$
f_{c2}
A_2
$a_{c2} d_{c2} s_{c2} r_{c2}$
(b)
f_{m1} I_1
$a_{m1} d_{m1} s_{m1} r_{m1}$
f_{m2} I_2
$a_{m2} d_{m2} s_{m2} r_{m2}$
f_{m3} I_3
$a_{m3} d_{m3} s_{m3} r_{m3}$
f_{c1}
A_1
$a_{c1} d_{c1} s_{c1} r_{c1}$
f_{c2}
A_2
$a_{c2} d_{c2} s_{c2} r_{c2}$
f_{c3}
A_3
$a_{c3} d_{c3} s_{c3} r_{c3}$
(c)

Figure 7.2: (a) single, (b) double and (c) triple parallel time-varying simple FM arrangements

Figure 7.2a represents the most fundamental time-varying simple FM synthesis model, also defined originally by Chowing (1973). In the time-varying synthesis models the carrier amplitude A and modulation index I are controlled by a simple synthesis component known as an envelope generator. Each envelope generator introduces the four parameters $adsr$ (attack, decay, sustain and release), which enable the envelope modulated parameters to change over time. This temporal control results in the production of time-variant sound textures[9] and is implemented in many commercial synthesisers (including the Yamaha DX/TX series).

The general form of the *adsr* envelope is depicted in figure 7.3. The attack parameter a sets the time taken for the output of the envelope generator to reach the value of 1.0 from a starting value of 0.0. The decay parameter d controls the time taken for the output to fall from 1.0 to the value specified by the sustain parameter s. Finally, the release parameter r controls the time taken for the output to reach 0.0 from s, after the sustain period is complete. Typically, the duration of the sustain period is controlled by the user as a performance parameter, ending when a note is released. However, since the matching procedure is automated, the overall duration of the envelope is matched to that of the target sound, and sustain time is set equal to the period that remains between the end of the decay period and the beginning of the release period.

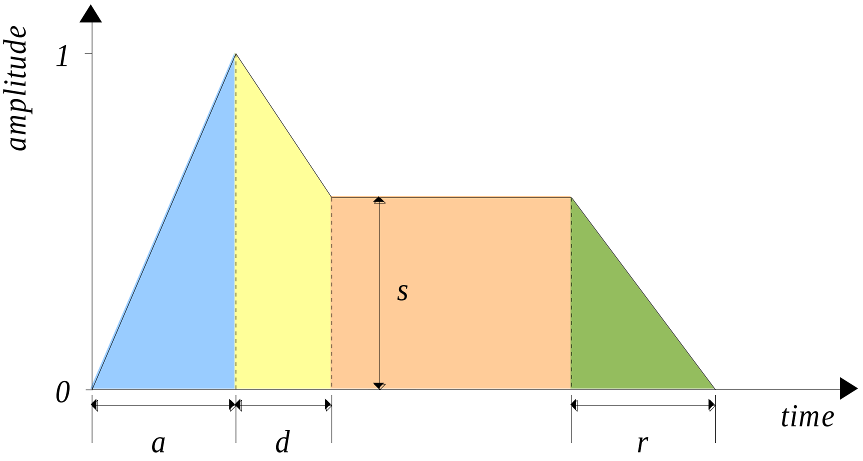


Figure 7.3: adsr envelope generator

Later in this chapter a complete FM matching system is developed and tested. When matching static tones, the system will employ the synthesis models depicted in 7.1; when matching time-varying sounds, the models depicted in figure 7.2 will be employed.

[9] It should be noted that this method of dynamic sound production differs significantly from the FM/Wavetable synthesis models implemented by Horner (Horner et al, 1993a, and Horner 1993).

7.1.1 FM Synthesis Parameter Ranges

The function and range of each synthesis parameter for the matching models used in the experiments presented here are summarised in table 7.1. For the experimentation, the fundamental frequency is fixed at 440Hz (concert pitch).

Parameter	***Function***	***Range***
f_c, f_m	Frequency of the carrier and modulator sinusoid oscillators, specified as multiples of the synthesiser fundamental frequency (440Hz)	0.0 – 8.0
A	Amplitude of the carrier oscillator	0.0 – 1.0
I	Modulation index, to control the amplitude of the modulating oscillator	0.0 – 8.0
a_c, a_m	Attack parameter for the envelope generator	0 – 50% target duration
d_c, d_m	Decay parameter for the envelope generator	0 – 25% target duration
s_c, s_m	Sustain parameter of the envelope generator	0.0 – 1.0
r_c, r_m	Release parameter of the envelope generator	0 – 25% target duration

Table 7.10: Synthesis parameter summary

7.2 EA Representation

Unlike previous matching studies, including Horner *et al* (1993a), Horner (1993) and Riionheimo and Välimäki (2003), the matching system is not tuned for matching particular types of sounds (harmonic or otherwise) by configuring the parametric representation with knowledge of the underlying problem domain. Each synthesis parameter is represented by a real number, which may take any value in the range specified in table 7.1.

The number of synthesis parameters (the problem dimensionality) is indicated in table 7.2 for each FM synthesis model depicted in figures 7.1 and 7.2. Each oscillator in the static model requires two parameters, while each oscillator in the time-varying model requires six parameters.

Model	***Static***	***Time-varying***
Single simple FM (Fig 7.1a)	4	12
Double simple FM (Fig 7.1b)	8	24
Triple simple FM (Fig 7.1c)	12	36

Table 7.11: Problem space dimensionality summary

7.3 Evolutionary Matching Synthesis Procedure

The sections that follow empirically analyse the performance of six ES-based algorithms when applied to optimise the FM synthesis matching problem. The evolutionary sound matching procedure is represented by the block diagram depicted in figure 7.4.

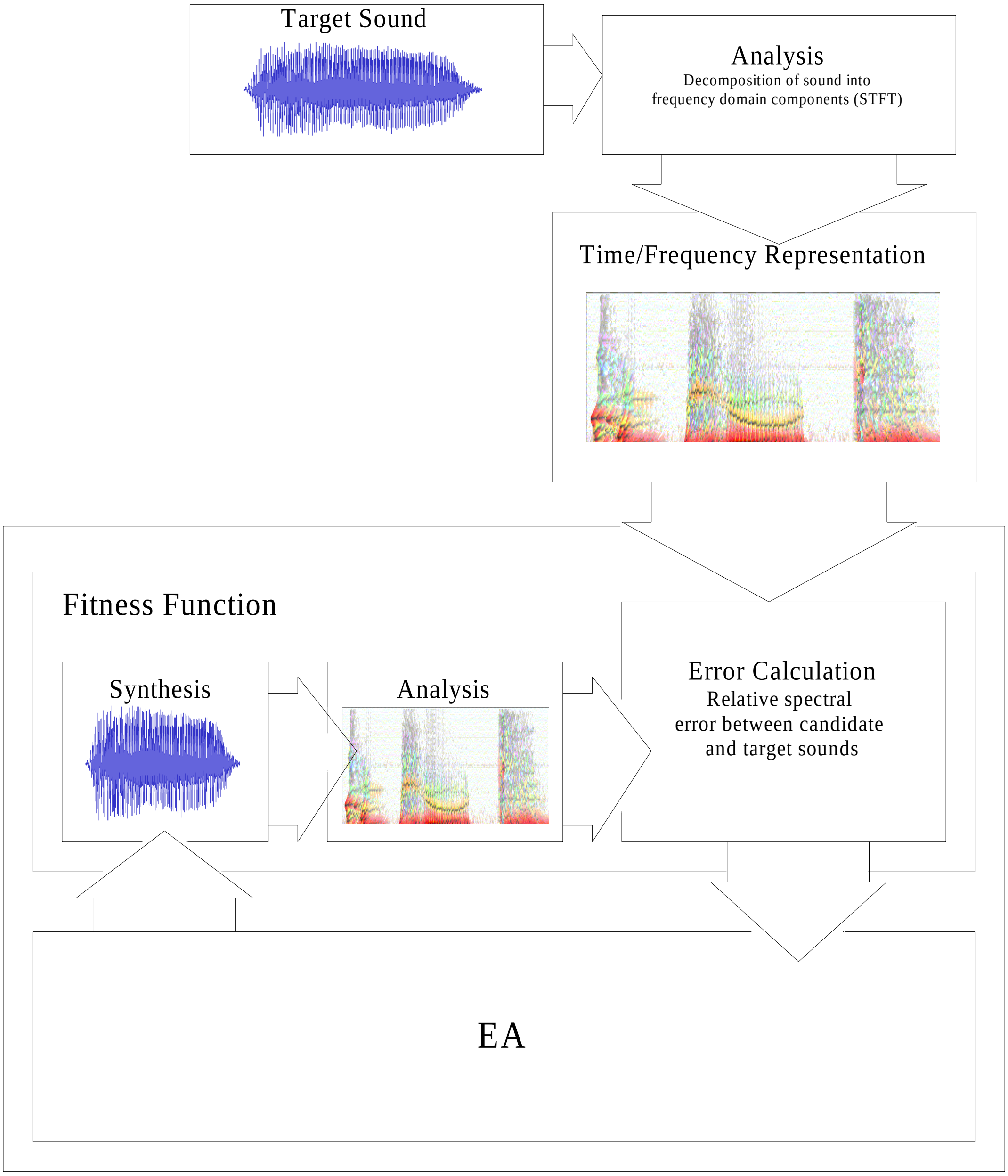

Target Sound
Analysis
Decomposition of sound into frequency domain components (STFT)
Time/Frequency Representation
Fitness Function
Synthesis
Analysis
Error Calculation
Relative spectral error between candidate and target sounds
EA

Figure 7.4: Evolutionary sound matching model

Initially, the system is supplied with a target sound to be matched. A frequency domain representation of the target is then extracted by Short-Time Fourier Transform (STFT). This is achieved by dividing the continuously-sampled target signal $x(n)$ into frames, which are transformed into the frequency domain data by Discrete Fourier Transform (DFT):

$$X(m\,,k) = \sum_{n=0}^{N-1} w(n)x(n+ms)e^{-j\frac{2\pi kn}{N}} \quad , \quad m = 0,1,2,3,\ldots, \quad k = 0,1,\ldots,N-1 \quad (7.1)$$

where $X(m\,,k)$ is the STFT of the signal $x(n)$, with integers m and k referring to the frame index and frequency bin respectively. N is the DFT frame size, s is the step-size between successive time frames, and $w(n)$ is a window function.

If $w(n) = 1$ for $0 \leq n \leq N-1$, the *rectangular window* is assumed and the continuously sampled signal frame is transformed directly. However, if the signal contains frequencies which are not periodic with the transform size (*N*), discontinuities are likely to occur at frame boundaries. These discontinuities result in unwanted artefacts across all frequencies of the output spectrum, an effect referred to as *spectral leakage* (Harris, 1978). When $w(n)$ is a non-rectangular symmetrical function that attenuates the signal at frame boundaries, the effects of spectral leakage are suppressed. There is a plethora of windowing shapes, none of which may be considered optimal for all signals and transform settings (Roads, 1996). For the forthcoming experiments in this thesis, the widely used and often cited Hamming function is employed to reduce boundary discontinuities (Miranda, 2002), (Ifeachor and Jervis, 2002), (Smith, 2003). The Hamming window is defined by the equation:

$$w(n) = 0.54 - 0.46\,cos\left(\frac{2\pi n}{N-1}\right) \quad , \quad 0 < n \leq N \qquad (7.2)$$

The frame size of the transform determines the resolution of the frequency spectrum. If *N* is set too small, insufficient frequency resolution will preclude an accurate match. The same is true for time resolution: when matching time-varying sounds, multiple spectra are required to capture the development of the frequency spectrum. Previous matching efforts have utilised the complete set of short-time spectra, measuring the average error computed for all frames (Riionheimo and Välimäki, 2003). However, since many musical sounds develop slowly with time, often only a small number of frames are required to sufficiently represent the target sound (Beauchamp and Horner, 2003). This has been exploited previously by Horner *et al* (1993a and b), Horner (1993) and Mitchell and Sullivan (2005). Both the quantity (N_{frames}) and the size (N) of the frames must therefore be assigned values that enable the target tone to be sufficiently characterised by its transform data.

If the frame size N is assigned a power of two, the frequency transformation may be computed more efficiently by Fast Fourier Transform (FFT) (Cooley and Tukey, 1965). Typical values include 256, 512, 1024, 2048, 4096 and 8192 (Zolzer, 2002). For the presented experimentation with time-varying sounds, 10 frames of size 1024 are taken at uniform intervals throughout the duration of the target. For static tones, a single frame of size 1024 is taken. These values were shown to be adequate for producing accurate matches in preliminary investigations. Furthermore, from a practical standpoint, it is desirable to keep these values as small as possible due to the computational advantage which ultimately results in faster matching.

However, it should be noted that with this computational advantage comes certain limitations. For example, 10 frames may be inappropriate for characterising target sounds with very complex variations. Moreover, their uniform distribution may be sub-optimal for target sounds that vary rapidly. There are certainly alternative placement strategies that might prove more suitable in certain applications. For example, Horner (1993) biases frame placement at the beginning of the sound to ensure that the perceptually significant initial transients of acoustic instrument sounds are well represented. However, this specialisation is inappropriate when critical time-varying components may be present elsewhere in the target sound. Furthermore, given the simplicity of the time-varying synthesis components of the matching system, and the unbiased nature of this preliminary exploration, this frame size and positioning configuration is considered appropriate. This subject is revisited in chapter eight where system limitations and improvements are discussed in greater detail.

Following the analysis of the target signal, its spectral representation is inserted into the error metric for subsequent comparison with potential matches (fitness evaluation). The EA population is then randomly initialised and optimised, in a cycle of variation and selection, to breed increasingly closer matches to the target signal. Fitness for each individual is determined by the following procedure:

- Insert candidate solution into the FM model to synthesise a corresponding waveform.
- Transform waveform into frequency domain representation by spectrum analysis.
- Compute fitness by comparing the target and synthesised candidate spectra using the relative spectral error metric.

The procedure then iterates until an error or generational threshold is reached.

7.4 Contrived Sound Matching – An Experimental Test Method

In this section, a test method is presented that enables the performance of different optimisation algorithms to be assessed comparatively in application to the FM synthesis matching problem. One of the principal goals is to measure the ability of each EA to access all regions of the synthesis space, and consistently identify high fitness matches.

In previous research, the performance of sound matching systems was frequently quantified by measuring the quality (fitness) of the optimised solutions when matching arbitrary target sounds. Target sounds may be real dynamic sounds originating from acoustic instruments (Horner, 1993 and 1998), or simple periodic tones generated by additive synthesis (McDermot *et al*, 2005). An alternative method is proposed here, whereby performance is measured by the ability of each optimiser to match randomly generated *contrived* targets. This approach is inspired by the early FM matching work presented by Justice (1979) and Payne (1987), and compares favourably with the approach adopted by Riionheimo and Välimäki (2003).

A contrived target is a sound or tone that originates from within the search space of, and is generated by, the matching synthesiser. This approach has been adopted by the author of this thesis in previous studies to assess the performance of the FCES when applied to optimise three different FM synthesis models (Mitchell and Sullivan, 2005), and to compare the performance of three evolutionary algorithms for matching tones with the simple FM arrangement (Mitchell and Pipe, 2006).

Contrived target sounds provide two significant advantages over experimentation with non-contrived alternatives, both related to easing the task of benchmarking the performance of the matching system:

- firstly, it is simple to determine when an optimal solution has been evolved as it matches the target sound exactly, achieving a relative spectral error of zero. If non-contrived target sounds are chosen as test specimens, confirmation of optimal convergence is not as easy. For example, the matching synthesiser may not be capable of exactly reproducing a particular target sound recorded from a real acoustic instrument, in which case a match delivering a relative error of zero cannot be achieved. In these circumstances an optimal match may only be confirmed when an exhaustive search yields no better result. An approach which becomes infeasible as the problem dimensionality increases.

- secondly, producing targets by randomly generating points within the synthesis space ensures that the test set constitutes a diversity of search space positions, and thus assesses performance on a variety of search space landscapes, as the topology of the landscape is dependent upon the properties of the target sound. Moreover, repeated matching of random *contrived* targets demonstrates that it is possible to access all regions of the search space.

In this context, the contrived sound matching method becomes a retrieval problem: the target is known to exist within the search space, and the ability of each EA to retrieve its location is tested. The results of this experimentation may then be used as an indicator of the system's ability to find the most accurate match of any sound type, and not just the limited scope of musical instruments. It may then be postulated that, if it is possible to consistently and accurately match contrived targets, the system will be capable of evolving an optimal match of any arbitrary target sound. This is confirmed later in this chapter by matching a selection of sounds originating from non-FM synthesis models and real acoustic instruments.

7.5 An Analysis of Evolutionary FM Synthesis Sound Matching Performance

In this section, a comparative analysis of a set of ES-based algorithms is performed when applied to the FM matching problem. The intention is to identify the algorithms most suited to this real-word problem domain, and to provide an environment in which the performance of the EAs proposed earlier in this thesis can be assessed.

Algorithm Selection

Six evolutionary algorithms are tested and compared in the following experimentation:

1. Evolution Strategy (ES)
2. Multi Start (1+1) Evolution Strategy (MSES)
3. Cooperative Coevolution Strategy (CCES)
4. Fuzzy Clustering Evolution Strategy (FCES)
5. Clustering Evolution Strategy (CES)
6. Clustering Cooperative Coevolution Strategy (CCCES)

Algorithms one to four were briefly described in section 4.3 while algorithms five and six correspond to the niching evolution and coevolution strategies presented in chapters four and five. In the later experimentation, a random search algorithm (RAND) is also included for comparison in which solutions are generated completely at random.

Algorithm Structure and Parameters

To ensure parity across all experiments, consistent algorithmic parameters and operators are fixed for all test cases. Indicated results are calculated by the mean average of 30 runs, matching 30 randomly generated contrived targets. Each algorithm is tested when matching the same target set and, where possible, populations are initialised with the same random data points, enabling observed differences between results to be attributed to the search properties of the EAs. Each algorithm runs for exactly 50 generations, except the cooperative coevolutionary algorithms, which run for approximately the same number of fitness evaluations, unless indicated otherwise. Where applicable, both intermediate (or centroid for the CES and CCCES) and discrete recombination are employed. Exceptions to this include the CCES in which only the discrete recombination operator is adopted (based upon the outcome of earlier experimentation in chapter four), the MSES and RAND algorithms, in which there is no recombination. For the purposes of brevity only results from experimentation with extinctive (*comma*) selection are included, as performance was found to be superior to the elitist (*plus*) selection strategy. It has been widely accepted, that the extinctive selection mechanism is most appropriate when a self-adaptive mutation operator is adopted (Schwefel, 1995), (Bäck and Schwefel, 1993). As in previous experimentation, selection pressure is maintained at a constant ratio of $\mu/\lambda = 1/7$, with exact figures indicated for each run. The objective of each algorithm is to minimise the relative spectral error (equation 6.3). All population sizes for each algrothm type are set relative to an ES population size of (200, 1400). Cooperative coevolutionary algorithm subpopulations are scaled to produce comparable fitness evaluations in an equivalent number of *rounds*.

Algorithmic Variation

Two variants of the cooperative coevolutionary algorithm are included in the results that follow: the conventional CCES and the proposed CCCES. In all experimentation with the CCES, results are provided for both the *single-best* and *single-best plus one random* collaboration strategies (Potter and De Jong, 1994), referred to as CCES1 and CCES2 respectively. Furthermore, a *natural* decomposition is adopted for the both variants of

CCES, in which each dimension of the problem space is considered to be a separate subproblem, optimised independently by a dedicated subpopulation of potential component solutions. Due to the linkage procedure (see section 5.3.1), it is impractical to implement a natural decomposition for the CCCES in high-dimensional problem spaces; consequently, an alternative decomposition is adopted here. Given that the FM synthesis architecture is inherently modular it is possible to identify natural parameter groupings that guide the chosen decomposition. For example, the synthesis models that feature in subsequent experimentation are differentiated by the number of *simple* FM synthesis elements that make up their structure. The parameters of each element do not directly affect other elements, therefore it is sensible to decompose the problem space such that each synthesis element is considered as a separate subproblem. The problem space for the synthesis model can be decomposed further still, such that the oscillator parameters of each element constitute separate subproblems. In general, each synthesis problem space can be decomposed into the number of separate FM synthesis oscillators or FM synthesis elements that make up their structure. Exact parameter decompositions are indicated for each experiment.

The FCES included in the subsequent experiments is of the original type proposed by Sullivan in his PhD thesis (Sullivan, 2002), in which a global selection scheme is used. Consequently, the model is only capable of resolving a single solution to the matching problem. The CES, on the other hand, employs the *restricted cluster selection* procedure, defined in chapter three, facilitating the concurrent maintenance of multiple distinct search space solutions (*niching*). In both algorithms the cluster cardinality is fixed at five.

7.6 A Performance Analysis of Evolutionary Static Tone Matching

In this section, the performance of each EA on the static tone FM matching problem is assessed, according to the contrived matching method introduced above. Thereafter, results when matching non-contrived target tones that do not originate from the matching synthesiser are provided. Matches are performed using two types of non-contrived tones, the first of which originates from a simple exciter/resonator synthesis system, and the second of which is extracted from the sustained period of real acoustic trumpet and oboe sounds.

7.6.1 Experiments with Static Tone Contrived Targets

In this first group of experiments, each evolutionary algorithm is applied to match 30 randomly generated contrived targets, using the three simple FM synthesis models depicted in figure 7.1. Each contrived target sound is created by drawing synthesis parameters uniformly at random from within the object range of each parameter. The procedure is repeated for all evolutionary algorithm variants, notated in the results according to the following format:

(strat)	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*

in which *strat* refers to the strategy type indicating the population sizes and selection mechanism in the regular $(\mu \overset{+}{,} \lambda)$ ES notation; *algo* specifies to the type of evolution strategy; *recomb* indicates whether intermediate (*int*), centroid (*cent*) or discrete (*disc*) recombination is employed; *gens* specifies the number of generations for which the algorithm runs (or *rounds* in the case of CCES-based algorithms); *evals* provides the corresponding number of objective function calls for each run; *sel* signifies the scope of the selection operator (global (*glob*) or restricted (*rest*), where restricted is only applicable to the clustering-based algorithms); *clusters* specifies the number of clusters, again only applicable to cluster-based strategies; and *subpopulations* specifies the number of coevolving populations, applicable only to the cooperative coevolution algorithms.

Performance Criteria

Results are presented for the single, double and triple FM matching models, according to the following criteria:

- ***Success* -** The number of runs in which the contrived targets are accurately retrieved.
- ***Average*** - The mean average relative spectrum error of the best solutions located for each of the 30 runs.
- ***Remaining*** - The mean average of the best solutions for all unsuccessful matches.

In the results that follow, a successful match is considered to be achieved when at least one of the population members delivers a relative spectral error below 0.01, i.e. the spectrum is 99% matched. A successful match then ensures that all parameters have been successfully retrieved.

7.6.1.1 Contrived Matching with Single Simple FM

The static tone single simple FM model presents a four-parameter synthesis model as depicted in figure 7.1a. This model constitutes the most fundamental FM network and is constructed from two sinusoidal oscillators. The data retrieved from the experiments with this synthesis model are tabulated and plotted below.

										Single Simple FM (n = 4)				
		Algorithmic Parameters									*Average*		*Remaining*	
Algorithm		*(strat)*	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*	**Success**	*mean*	*σ*	*mean*	*σ*
CCES	1	*(50,350)*	*CCES1*	*disc*	*50*	*70200*	*glob*	*-*	*4*	2	0.43	0.23	0.46	0.20
	2	*(50,350)*	*CCES2*	*disc*	*25*	*70200*	*glob*	*-*	*4*	15	0.19	0.23	0.37	0.18
ES	3	*(200,1400)*	*ES*	*disc*	*50*	*70200*	*glob*	*-*	*-*	13	0.25	0.26	0.44	0.19
	4	*(200,1400)*	*ES*	*int*	*50*	*70200*	*glob*	*-*	*-*	6	0.46	0.27	0.58	0.15
MSES	5	*1400×(1+1)*	*MSES*	*-*	*50*	*70000*	*-*	*-*	*-*	0	0.16	0.11	0.16	0.11
	6	*350×(1+1)*	*MSES*	*-*	*200*	*70000*	*-*	*-*	*-*	6	0.12	0.13	0.15	0.13
	7	*175× (1+1)*	*MSES*	*-*	*400*	*70000*	*-*	*-*	*-*	8	0.10	0.12	0.14	0.12
	8	*100× (1+1)*	*MSES*	*-*	*700*	*70000*	*-*	*-*	*-*	9	0.15	0.18	0.22	0.18
FCES	9	*(200,1400)*	*FCES*	*disc*	*50*	*70200*	*glob*	*40*	*-*	27	0.04	0.12	0.40	0.03
	10	*(200,1400)*	*FCES*	*int*	*50*	*70200*	*glob*	*40*	*-*	14	0.27	0.27	0.50	0.14
CES	11	*(200,1400)*	*CES*	*disc*	*50*	*70200*	*rest*	*40*	*-*	29	0.00	0.01	0.03	0.00
	12	*(200,1400)*	*CES*	*cent*	*50*	*70200*	*rest*	*40*	*-*	19	0.05	0.12	0.12	0.17
CCCES	13	*(50,350)*	*CCCES*	*disc*	*50*	*45200*	*rest*	*10*	*2*	24	0.02	0.07	0.12	0.13
	14	*(50,350)*	*CCCES*	*cent*	*50*	*45200*	*rest*	*10*	*2*	18	0.05	0.10	0.13	0.13
	15	*(50,350)*	*CCCES*	*disc*	*100*	*90200*	*rest*	*10*	*2*	25	0.02	0.07	0.14	0.14
	16	*(50,350)*	*CCCES*	*cent*	*100*	*90200*	*rest*	*10*	*2*	24	0.03	0.16	0.17	0.16

Table 7.12: Results when matching Single Simple FM contrived tones

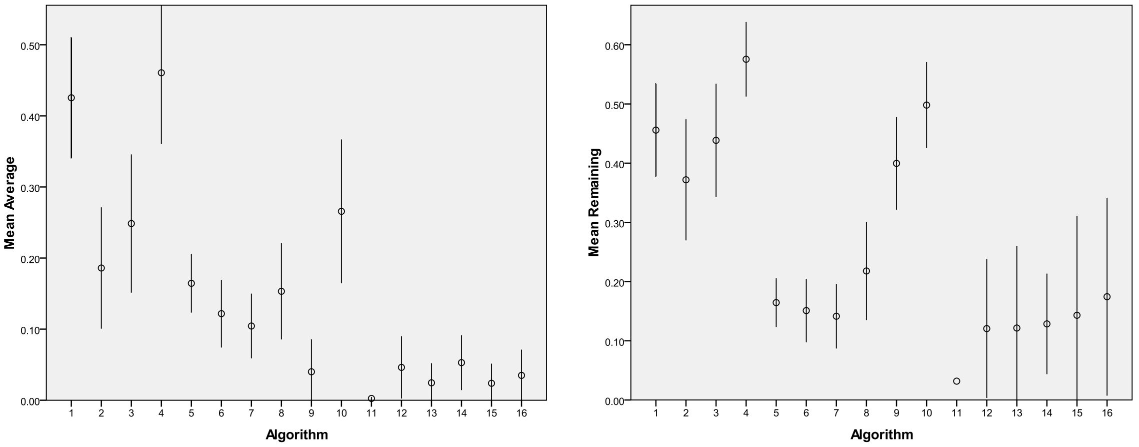


Figure 7.5: Mean and 95% confidence intervals for Average and Remaining error results when matching Single Simple FM contrived tones

Discussion

As indicated in figure 7.5 and confirmed by running a one-way ANOVA on these results, the average error values are not equal across all algorithms, there are clear performance differences between them.

A post hock comparison between means indicated that algorithms one and four, the CCES1 and ES with intermediate recombination, produced larger errors than all other algorithms (significant at the 0.05 level). Compared with the CCES1, the CCES2 demonstrated a considerable performance improvement, retrieving exactly half of the contrived target tones while also reducing the average error of the unsuccessful matches, again significant at the 0.05 level. Similar comparisons indicated significant improvements for the canonical ES when the discrete recombination operator is employed, successfully retrieving more than double the number of target tones than the intermediate algorithm.

Several instances of the MSES algorithm are included in these results. As was established in chapter four, the MSES progresses at a lower rate than the other EAs as recombination is omitted. Consequently, four MSES test cases were included in the experimentation, varying in the number of generations for each run. To maintain parity, the population sizes were adjusted to ensure that the same number of fitness evaluations was computed for each variant. In terms of successful tone retrieval, the MSES does not compare favourably with other algorithms. However, in terms of the average error, the MSES delivers results that tend towards the more successful algorithms included in this comparison. This behaviour suggests that, within this domain, the MSES is unable to retrieve optimal matches consistently but it is able to consistently locate closer matches than the conventional ES, especially when the number of $(1 + 1)$ strategies is set to 175.

The cluster-based algorithms are clearly advantageous within this problem domain, confirming that the search space is well suited to a speciation-type EA. The results indicate that the novel CCCES architecture, specifically algorithm 13, produced a smaller mean error than all other algorithms except the discreet recombination CES (significant at the 0.05 level against algorithms one, two, three, four, five and 10). Algorithm 13 (CCCES) was also able to successfully retrieve more target tones than either instance of the CCES in considerably fewer fitness evaluations. The CES and FCES both performed well, with the discreet recombination CES retrieving the highest number of target tones and achieving the smallest mean error (significant at the 0.05 level against algorithms one, two, three, four, five, six, seven, eight and 10).

Unlike the FCES, both the CES and CCCES are able to extract multiple solutions from the search space. The fittest solution from each CES cluster provides a selection of alternative matches of varying accuracy. Example matches extracted from one of the CES runs are provided in table 7.4. For this run, 40 separate matches were extracted, of which the relative spectral error ranged from 0.0 to 1.0. For brevity, only the top ten solutions are tabulated. The parameters of the first match are identical to those used to synthesise the target tone and thus achieve a relative spectral error of zero.

	Synthesis Parameters				
Cluster	f_c	A	f_m	I	***Error***
14	1.580	2.682	6.146	2.222	0.000
28	4.582	2.870	3.080	2.901	0.228
10	7.681	2.478	3.070	1.356	0.355
30	4.745	2.906	3.068	4.470	0.373
29	1.728	2.527	3.064	3.227	0.428
15	4.634	2.650	3.052	1.398	0.500
20	6.161	2.397	1.603	1.889	0.532
21	7.813	2.852	3.081	6.097	0.562
25	3.159	2.805	2.635	4.563	0.569
5	3.041	2.643	3.718	3.261	0.586

Table 7.13: Top 10 multiple solutions delivered by the CES for a contrived match

Each solution in table 7.4, and the 30 solutions that are not shown, are positioned at different regions of the search space, indicating that each cluster has converged to an independent niche. The equivalent solutions from the CCCES are provided in table 7.5.

Synthesis Parameters				***Error***
f_c	A	f_m	I	
1.580	2.682	6.146	2.222	0.000
4.584	2.568	3.078	4.358	0.314
7.713	2.061	6.140	4.023	0.640
4.607	2.397	1.527	3.789	0.683
6.142	1.884	1.555	2.040	0.711
0.692	2.003	1.416	7.582	0.732
1.575	1.664	6.145	5.691	0.785
3.073	1.575	7.645	1.888	0.808
4.540	1.278	4.632	3.600	0.829
0.458	1.475	4.105	2.364	0.833

Table 7.14: 10 multiple solutions delivered by the CCCES for a contrived match

These CCCES solutions represent the final results extracted by *exclusive* linkage amongst representatives of each species (section 5.3.1). The table represents the complete set of final solutions, which produce a higher mean error than the top ten solutions produced by the CES (significant at the 0.05 level).

The convergence characteristics of the CCCES and CES are plotted in figure 7.6, with variation performed by discrete recombination for both algorithms. Plots are averaged over five matches performed on the same set of five randomly generated target sounds. Both strategies exhibit comparable rates of convergence, with the CES producing less error in the early stages of the search.

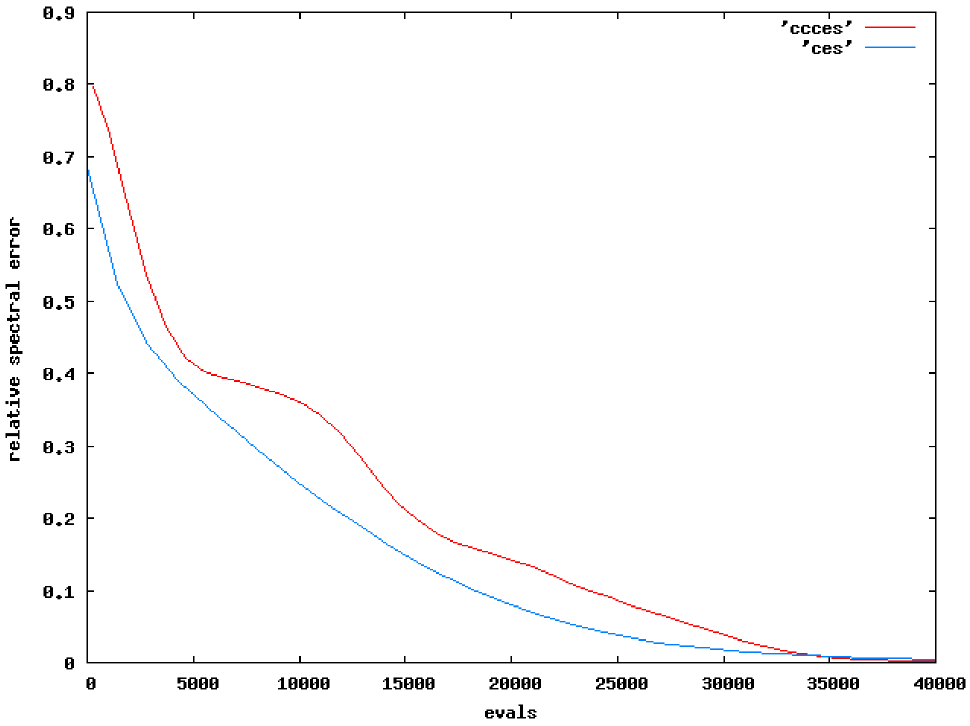


Figure 7.6: Convergence plot for the CCCES and CES

7.6.1.2 Contrived Matching with Double Simple FM

The static tone double simple FM model (figure 7.1b) presents an eight-dimensional FM synthesis matching problem. Composed of four sinusoidal oscillators, this synthesis model represents a parallel extension of the basic simple FM network. The results from the contrived-target experimentation with this model are provided in table 7.6 and plotted in figure 7.7.

Discussion

Compared with results obtained with the single simple FM model, the results indicate larger errors with fewer successful matches for all algorithms. This is principally because the search space is much larger than the single simple FM space, while the parameters of each evolutionary algorithm remain exactly the same.

Due to the parallel double simple FM arrangement, the search space may be considered to be symmetrical. That is, for any given contrived target, there exist two zero error search space optima. The first, where parameters match those used to generate the target tone, and the second, where the parameters for each of the parallel simple FM elements are swapped. So while the space is larger, there are more potential solutions within it.

										Double Simple FM (n = 8)				
		Algorithmic Parameters									*Average*		*Remaining*	
Algorithm		*(strat)*	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*	*Success*	*mean*	*σ*	*mean*	*σ*
CCES	1	*(25,175)*	*CCES1*	*disc*	*50*	*70200*	*glob*	*-*	*8*	4	0.35	0.20	0.41	0.15
	2	*(25,175)*	*CCES2*	*disc*	*25*	*70200*	*glob*	*-*	*8*	6	0.27	0.13	0.34	0.13
ES	3	*(200,1400)*	*ES*	*disc*	*50*	*70200*	*glob*	*-*	*-*	2	0.32	0.15	0.34	0.15
	4	*(200,1400)*	*ES*	*int*	*50*	*70200*	*glob*	*-*	*-*	0	0.50	0.18	0.50	0.18
MSES	5	*1400 x (1+1)*	*MSES*	*-*	*50*	*70000*	*-*	*-*	*-*	0	0.38	0.10	0.38	0.10
	6	*350 x (1+1)*	*MSES*	*-*	*200*	*70000*	*-*	*-*	*-*	0	0.38	0.13	0.38	0.13
	7	*175 x (1+1)*	*MSES*	*-*	*400*	*70000*	*-*	*-*	*-*	0	0.41	0.15	0.41	0.15
	8	*100 x (1+1)*	*MSES*	*-*	*700*	*70000*	*-*	*-*	*-*	0	0.40	0.13	0.40	0.13
FCES	9	*(200,1400)*	*FCES*	*disc*	*50*	*70200*	*glob*	*40*	*-*	1	0.31	0.15	0.32	0.14
	10	*(200,1400)*	*FCES*	*int*	*50*	*70200*	*glob*	*40*	*-*	0	0.49	0.16	0.49	0.16
CES	11	*(200,1400)*	*CES*	*disc*	*50*	*70200*	*rest*	*40*	*-*	1	0.20	0.10	0.21	0.10
	12	*(200,1400)*	*CES*	*cent*	*50*	*70200*	*rest*	*40*	*-*	1	0.31	0.14	0.32	0.13
CCCES	13	*(50,350)*	*CCCES*	*disc*	*50*	*45200*	*rest*	*10*	*2*	2	0.18	0.13	0.19	0.13
	14	*(50,350)*	*CCCES*	*cent*	*50*	*45200*	*rest*	*10*	*2*	1	0.23	0.13	0.24	0.13
	15	*(50,350)*	*CCCES*	*disc*	*100*	*90200*	*rest*	*10*	*2*	8	0.15	0.14	0.21	0.13
	16	*(50,350)*	*CCCES*	*cent*	*100*	*90200*	*rest*	*10*	*2*	6	0.17	0.14	0.21	0.13

Table 7.15: Results when matching Double Simple FM contrived tones

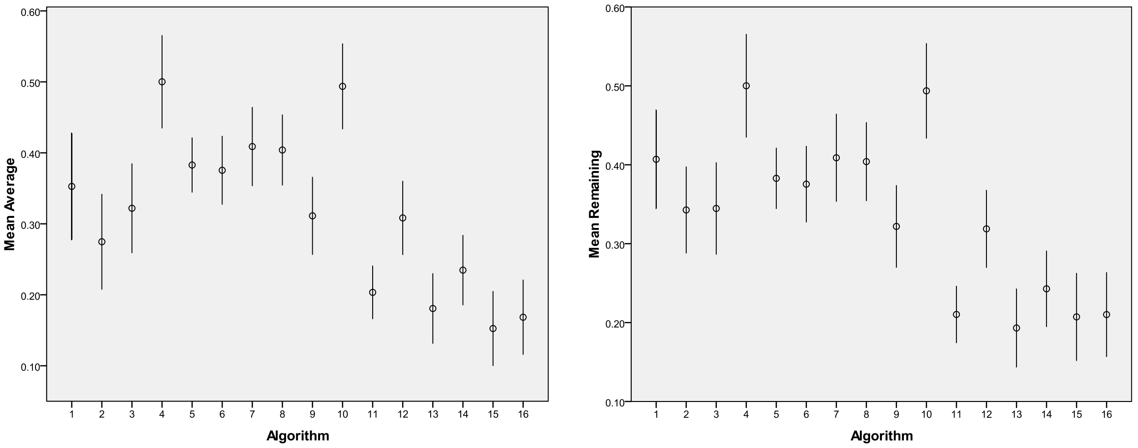


Figure 7.7: Mean and 95% confidence intervals for Average and Remaining error results when matching Double Simple FM contrived tones

As with the previous results, the ANOVA confirms a disparity of means across all algorithms, with the CES and CCCES algorithms delivering the smallest average errors. A post-hock comparison indicated that algorithm 11, the discreet recombination CES, produced a smaller mean error than all algorithms but the CCCES (significant at the 0.05 level against algorithms four, five, six, seven, eight and 10). Similar tests revealed that algorithm 15, the discreet recombination CCCES, produced a smaller mean error than all other algorithms (significant at the 0.05 level against algorithms one, two, three, four, five, six, seven, eight, nine, 10 and 12). In this respect, the FCES performs relatively badly, as does the ES, CCES and also the MSES, in which no parametric variation appears to result in any significant improvement. Although the CCES2 does successfully retrieve six of the 30 contrived targets, the quality of the remaining unsuccessful matches is poor, lowering its mean error considerably. The minimum average error of all tested algorithms is achieved by the CCCES followed by the CES. It should be noted that this small average error is achieved by the CCCES in significantly fewer fitness evaluations than the competing algorithms. Furthermore, when permitted to run for 100 generations, this error is further reduced, and the number of successful matches, increased.

7.6.1.3 Contrived Matching with Triple Simple FM

The contrived matching experimentation is now repeated using the last of the static synthesis models: the static tone triple simple FM model, as depicted in figure 7.1c. This adds a set of four parameters to the previously tested model, increasing the search space dimensionality to 12. The results are tabulated below.

										Triple Simple FM (n = 12)				
		Algorithmic Parameters									*Average*		*Remaining*	
Algorithm		*(strat)*	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*	*Success*	*mean*	*σ*	*mean*	*σ*
CCES	1	*(17,119)*	*CCES1*	*dis*	*50*	*71604*	*glob*	*-*	*12*	0	0.29	0.13	0.29	0.13
	2	*(17,119)*	*CCES2*	*dis*	*25*	*71604*	*glob*	*-*	*12*	0	0.23	0.13	0.23	0.13
ES	3	*(200,1400)*	*ES*	*dis*	*50*	*70200*	*glob*	*-*	*-*	0	0.28	0.14	0.28	0.14
	4	*(200,1400)*	*ES*	*int*	*50*	*70200*	*glob*	*-*	*-*	0	0.45	0.14	0.45	0.14
MSES	5	*1400 x (1+1)*	*MSES*	*-*	*50*	*70000*	*-*	*-*	*-*	0	0.41	0.11	0.41	0.11
	6	*350 x (1+1)*	*MSES*	*-*	*200*	*70000*	*-*	*-*	*-*	0	0.39	0.12	0.39	0.12
	7	*175 x (1+1)*	*MSES*	*-*	*400*	*70000*	*-*	*-*	*-*	0	0.40	0.10	0.40	0.10
	8	*100 x (1+1)*	*MSES*	*-*	*700*	*70000*	*-*	*-*	*-*	0	0.43	0.15	0.43	0.15
FCES	9	*(200,1400)*	*FCES*	*dis*	*50*	*70200*	*glob*	*40*	*-*	0	0.34	0.17	0.34	0.17
	10	*(200,1400)*	*FCES*	*int*	*50*	*70200*	*glob*	*40*	*-*	0	0.47	0.13	0.47	0.13
CES	11	*(200,1400)*	*CES*	*dis*	*50*	*70200*	*rest*	*40*	*-*	0	0.27	0.08	0.27	0.08
	12	*(200,1400)*	*CES*	*cent*	*50*	*70200*	*rest*	*40*	*-*	0	0.36	0.10	0.36	0.10
CCCES	13	*(30,210)*	*CCCES*	*dis*	*50*	*64206*	*rest*	*6*	*3*	0	0.22	0.09	0.22	0.09
	14	*(30,210)*	*CCCES*	*cent*	*50*	*64206*	*rest*	*6*	*3*	0	0.25	0.09	0.25	0.09
	15	*(30,210)*	*CCCES*	*dis*	*100*	*128106*	*rest*	*6*	*3*	0	0.21	0.09	0.21	0.09
	16	*(30,210)*	*CCCES*	*cent*	*100*	*128106*	*rest*	*6*	*3*	1	0.18	0.08	0.18	0.08

Table 7.16: Results when matching Triple Simple FM contrived tones

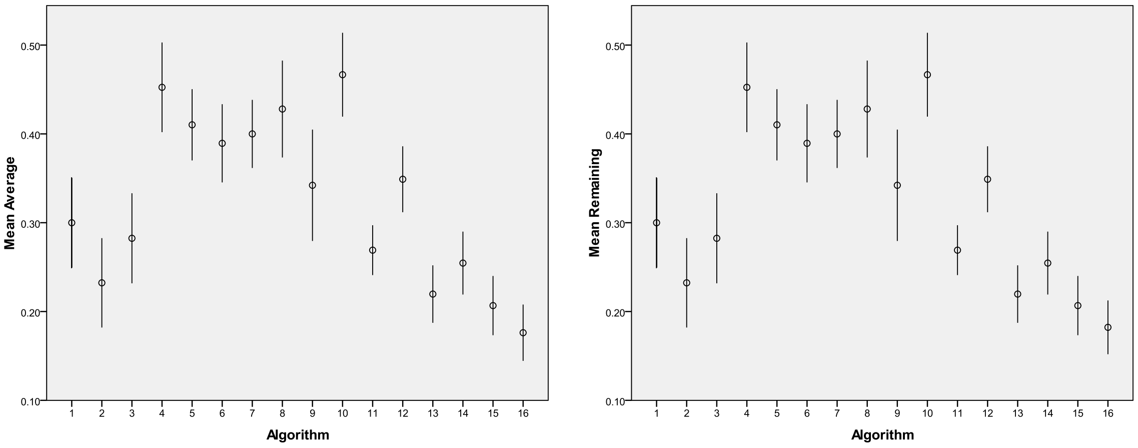


Figure 7.8: Mean and 95% confidence intervals for Average and Remaining error results when matching Triple Simple FM contrived tones

Discussion

The results of table 7.7 again indicate a general increase in error when compared with the results of the previous experimentation, with the exception of the CCES1. Interestingly, within this higher-dimensional problem space, the CCES1 achieves a lower mean error than either of the previous two rounds of experiments with the smaller synthesis models. The algorithms delivering the strongest performance are the CCES2, CES, and CCCES. In this larger problem domain, the CCES2 has outperformed the CES (significant at the 0.05 level for the centroid recombination CES), while the CCCES, most notably algorithm 13, outperforms all other algorithm types in fewer fitness evaluations (significant at the 0.05 level for algorithms one, four, five, six, seven, eight, 10 and 12).

The general increase in error correlates with an increase in search space complexity, which raises the question: are these results good enough? After all, a *good* match is a rather subjective quality. In Horner’s (1998) article, human listeners were unable to distinguish between acoustic instrument matches achieving a relative spectrum error of less than approximately 0.15 from their targets. By this comparison, the matches attained by both the discrete recombination based CES and CCCES may be considered to be acceptable. However, it cannot be assumed that the figures produced by Horner’s study are transferable to the static contrived FM sounds matched here. While a match might be mathematically accurate, it is difficult to determine whether it is perceptually accurate. An alternative visualisation of match difference is provided in figure 7.9, where the spectrum of a randomly generated triple FM target tone is plotted against an example match produced by the CES. This particular match achieves a relative spectrum error of 0.113.

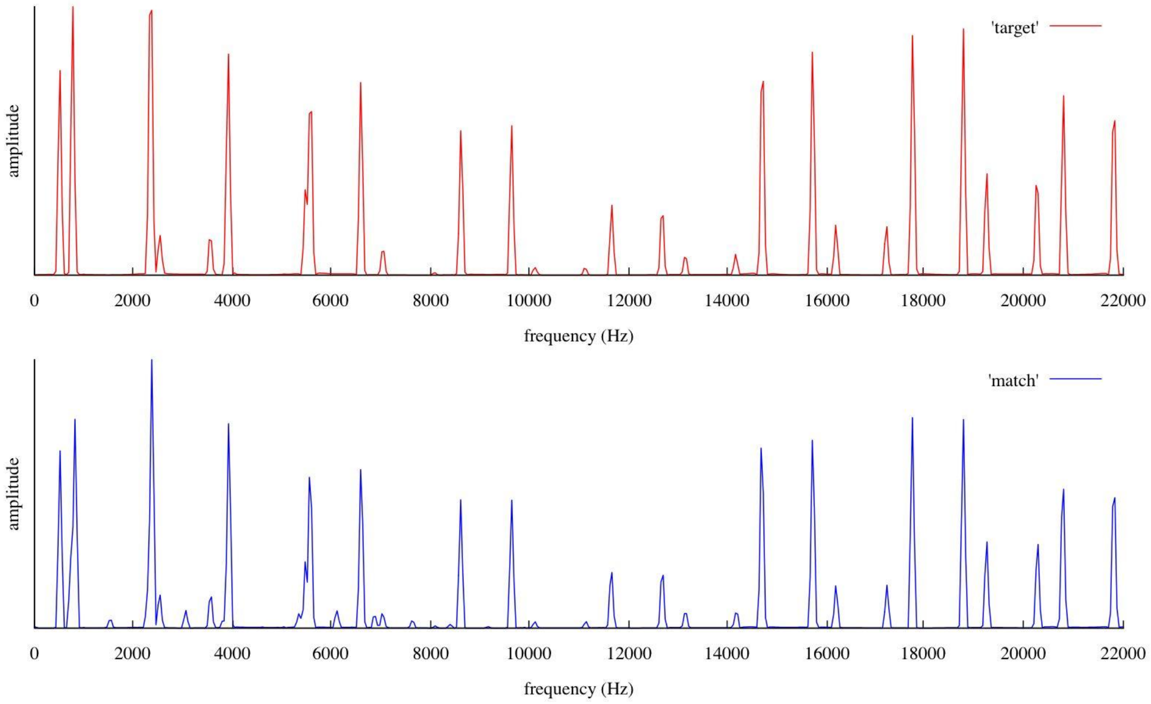


Figure 7.9: Static triple FM target (top) and corresponding CES match (bottom)

The FM target tone has a broad spectral envelope with partials distributed across the entire frequency spectrum. Many of the frequency partials are non-harmonic, range widely in amplitude and are reflected around 0Hz. Despite these characteristics, the CES match is accurate, with all target partials well represented in the match. There are only minor amplitudinal differences, with some additional low-amplitude partials present in the match. Erroneous partials may be observed in the match at approximately 1.5kHz, 3kHz and 6kHz. These differences are more apparent in figure 7.10 in which both spectra are overlaid on a logarithmic amplitude scale.

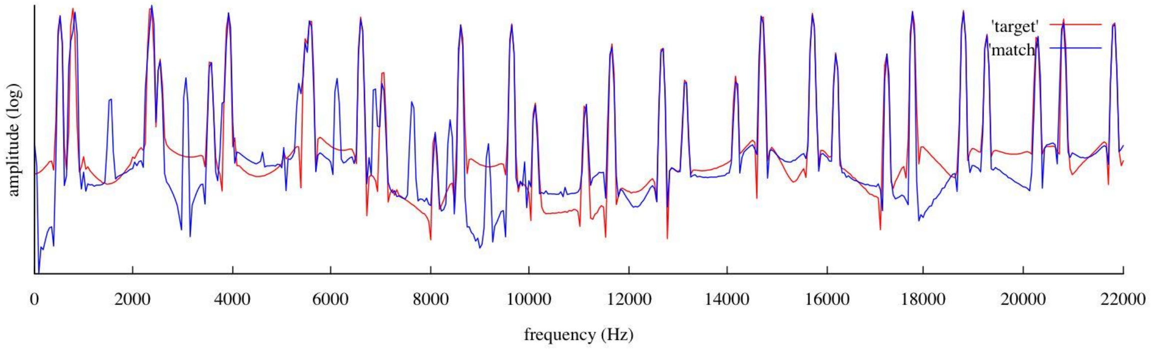


Figure 7.10: Static triple FM target and corresponding CES match with log amplitude scale

Despite the visible similarities between the target and match spectra, neither the low relative spectrum error nor the plots above constitute a perceptual test with human listeners. Given the quantity of target sounds and matches in this chapter, listening tests are deferred until chapter eight, where the correlation between discrimination by perceptual and relative spectrum error is examined. Thereafter, the sounds produced by the final matching system are analysed qualitatively by a panel of expert listeners.

In general, the final matches produced by the experimentation with the triple simple FM model indicate higher average error results than the earlier experimentation with the simpler FM models. However, the overall quality of the multiple solutions produced by the niching EAs are significantly improved with this more complex synthesis model. This is illustrated in table 7.8, where a list of 10 final CES solutions with the smallest relative spectral error is tabulated. For this particular run, the best individuals of all 40 clusters produce error results in the range [0.09, 0.6].

Cluster	f_{c1}	A_1	f_{m1}	I_1	f_{c2}	A_2	f_{m2}	I_2	f_{c2}	A_2	f_{m2}	I_2	**Error**
27	*4.206*	*0.746*	*1.545*	*5.412*	*5.776*	*0.382*	*1.882*	*3.465*	*7.121*	*2.766*	*0.501*	*0.206*	0.088
24	*2.704*	*0.575*	*3.083*	*1.877*	*2.067*	*0.617*	*1.880*	*4.281*	*7.114*	*2.787*	*3.236*	*0.149*	0.135
25	*1.833*	*0.444*	*1.944*	*4.030*	*7.125*	*2.543*	*0.199*	*0.212*	*4.894*	*0.646*	*1.727*	*8.000*	0.144
19	*6.575*	*0.407*	*3.780*	*1.399*	*7.123*	*2.766*	*1.386*	*0.109*	*2.105*	*0.684*	*1.876*	*4.513*	0.148
5	*1.794*	*0.525*	*1.467*	*5.101*	*7.126*	*2.749*	*7.891*	*0.043*	*1.778*	*0.651*	*1.436*	*7.856*	0.151
10	*1.823*	*0.559*	*2.146*	*6.167*	*7.130*	*2.743*	*0.513*	*0.162*	*7.647*	*0.583*	*2.038*	*2.481*	0.154
2	*7.271*	*0.650*	*1.611*	*0.865*	*7.101*	*2.264*	*0.310*	*0.149*	*7.751*	*0.668*	*0.839*	*7.286*	0.160
4	*7.119*	*2.751*	*4.751*	*0.114*	*0.369*	*0.836*	*1.537*	*6.198*	*1.087*	*0.158*	*7.040*	*4.178*	0.163
7	*6.308*	*0.745*	*0.660*	*6.067*	*1.955*	*0.356*	*4.970*	*3.250*	*6.933*	*2.496*	*0.042*	*5.133*	0.164
29	*7.125*	*2.704*	*1.406*	*0.156*	*4.094*	*0.229*	*6.386*	*2.801*	*6.357*	*0.732*	*0.726*	*6.210*	0.167

Table 7.17: Top 10 multiple solutions delivered by the CES for a contrived match

Each tone match in table 7.8 achieves a low relative spectral error despite being situated at independent points in the search space. Compared with the equivalent data for the single simple FM model (table 7.4), the overall quality of the solutions is significantly improved, despite the absence of an exact match. These results suggest that the larger model presents a problem domain in which the optimum is more difficult to locate, but high fitness solutions are more readily abundant.

7.6.2 Experiments with Static Tone Non-Contrived Targets

In the previous section, it was demonstrated that it is possible to optimise good matches to randomly generated contrived target tones using EAs to derive the parameters of the matching synthesiser. The algorithms shown to be most effective at exploiting the parameter space of the matching synthesiser were the CES and CCCES. In this section, these two derivatives of the FM matching system are applied to match non-contrived target tones originating from non-FM sources. Examples are presented with two sound types. The first set of target tones are synthesised using a simple additive-subtractive system and the second set are taken from recordings of real classical instrument sounds produced by Opolko and Wapnick (1989). Both of these tone types represent a more typical musical target than the randomly generated FM tones tested in the previous section. The relative spectral error measure is used to determine the quality of the match, and visualisations are provided in the form of frequency spectrum plots.

7.6.2.1 Evolutionary Synthesis Matching of Additive-Subtractive Synthesis Tones

The first of the non-contrived matching experiments applies the evolutionary matching FM synthesis model to three separate target tones generated by a simple additive-subtractive system. This synthesis model is analogous to the sound production form of many musical instruments: the tone of a broadband energy source vibration (additive synthesis component) is shaped by the modes of a connected resonant body (subtractive synthesis component) (Howard and Angus, 2000). Real world examples of this form include the source vibrations produced by an air jet across the mouthpiece of a flute which is coupled to a resonating column of air, or a hammer striking a piano string which is coupled to a resonant frame. Target tones are produced by a synthesis model based loosely on this form: a broadband spectrum is created by additive synthesis (Roads, 1996), which is subsequently shaped by a (subtractive) bandpass filter-bank (Roads, 1996). The broadband tone is generated using a 16-part sinusoidal additive synthesiser, with equally weighted partials positioned at intervals of 440 Hz. A single period of the corresponding waveform is depicted in figure 7.11.

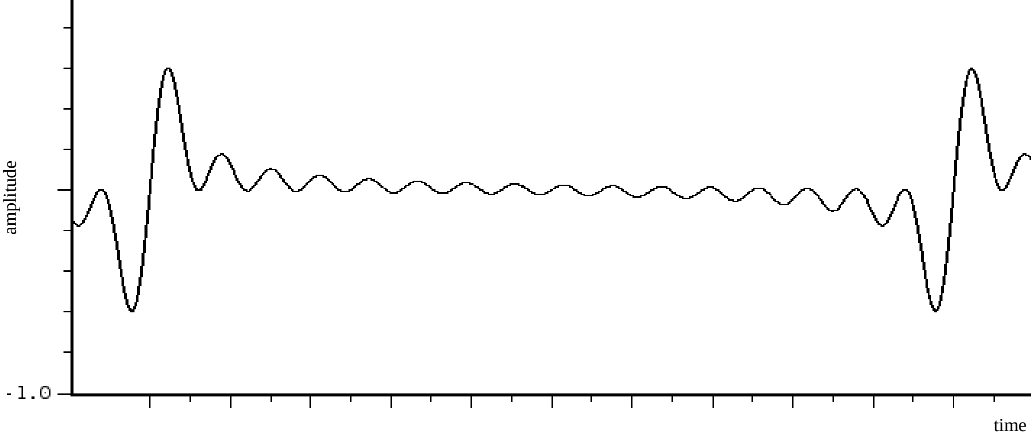


Figure 7.11: Unfiltered additive synthesis tone

The flat spectrum produced by the additive synthesiser is then passed through a subtractive bandpass filter network to shape the tone. The complete additive-subtractive synthesis model from which the non-FM target tones originate is depicted in figure 7.12.

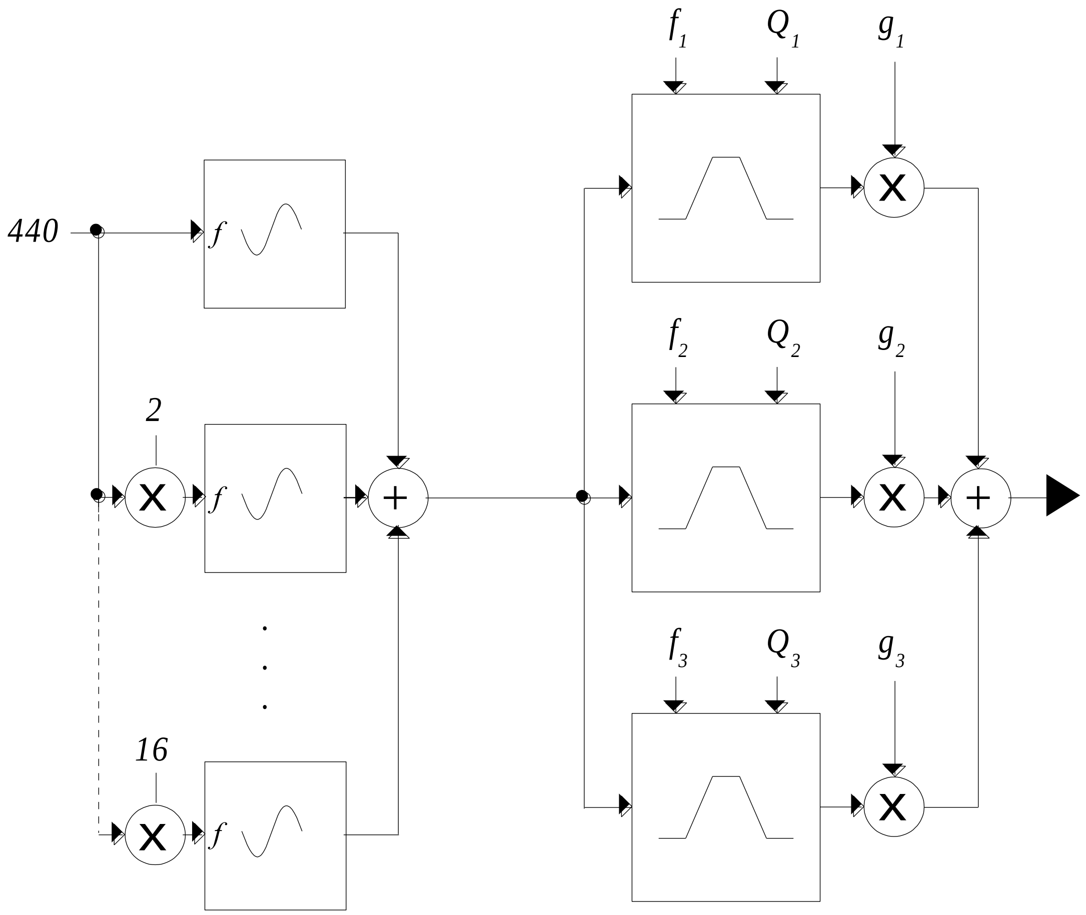


Figure 7.12: Additive-subtractive target tone synthesis model

For target tone one, only one bandpass filter is active; for tone two, two bandpass filters are active; and for tone three, all three filters are active. The bandpass filter parameters are centre frequency *(f)*, Q factor (ratio of the filter centre frequency to the width of the pass band), and gain *(g)*. The values for the additive-subtractive target tones are provided in the table 7.9; the corresponding spectra are plotted in figure 7.13.

	Bandpass 1			***Bandpass 2***			***Bandpass 3***		
Target Name	f_1	Q_1	g_1	f_2	Q_2	g_2	f_3	Q_3	g_3
Tone 1	440	2	1	-	-	-	-	-	-
Tone 2	440	2	1	1760	10	0.5	-	-	-
Tone 3	440	2	1	1760	10	0.5	3520	20	0.25

Table 7.18: Additive-subtractive target tone specifications

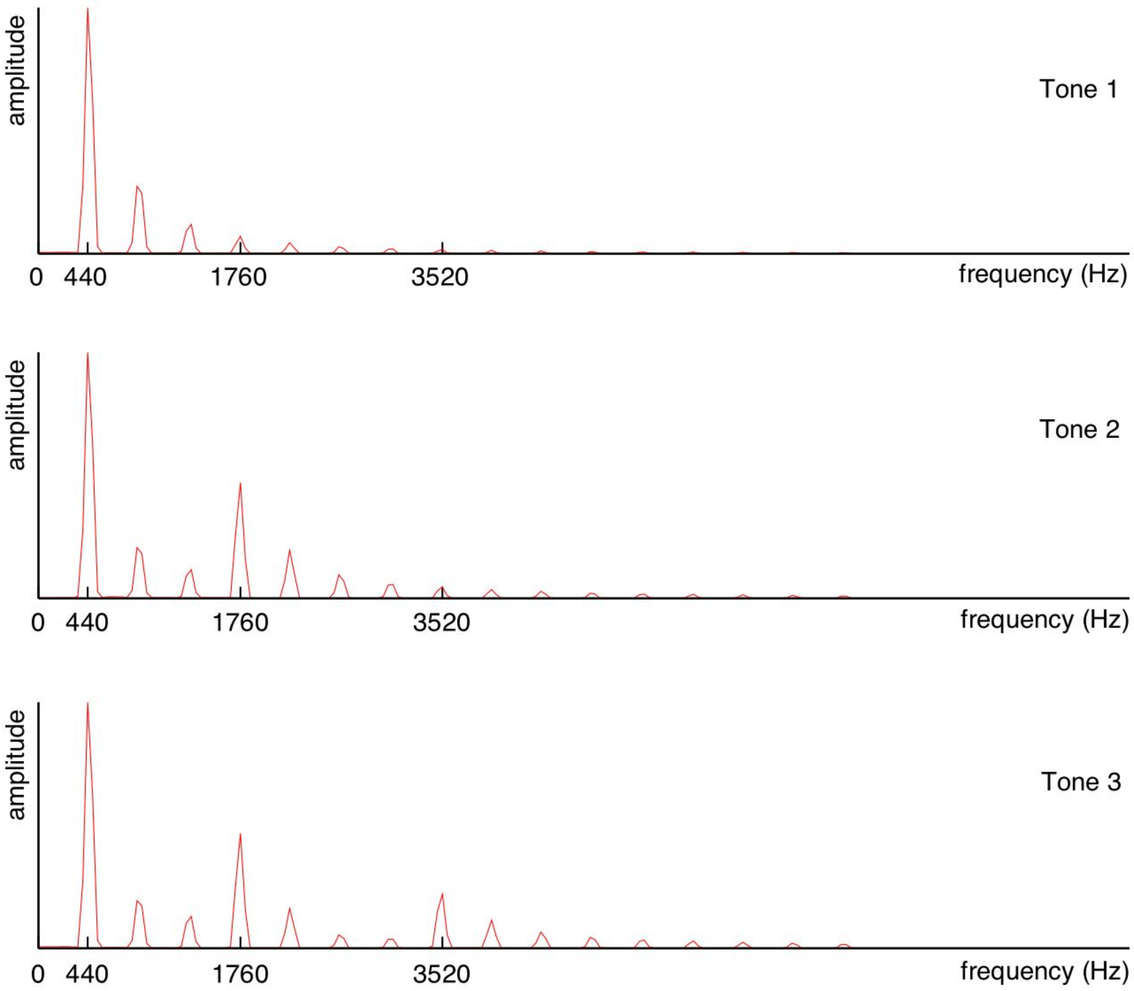


Figure 7.13: Additive-subtractive target tone spectra

Additive-Subtractive Tone Matching Results

In the previous section of this chapter, the CES and CCCES were found to be the most advantageous for matching contrived FM target tones, especially when discrete recombination operators were used. These algorithms are now applied to derive FM synthesis parameters that render tones matching the three additive-subtractive targets, introduced above. In addition, a random search is also included for comparison (RAND). Discrete recombination is employed for the evolutionary algorithms, with the exact same parametric settings as with the contrived experiments. Results are presented when matching each target tone with all three FM synthesis models depicted in figure 7.1. The accuracy of each match is indicated in terms of the relative spectral error, averaged over five independent runs. The results in table 7.10 and figure 7.14 illustrate some consistent trends. In all test cases, the CES and CCCES produce more accurate matches than the RAND algorithm, shown to be significant at the 0.05 level by a post hock ANOVA.

									Error					
Matching	**Test**	**Algorithmic Parameters**							Tone 1		Tone 2		Tone 3	
Model	**Case**	*(strat)*	*algo*	*gens*	*evals*	*sel*	*clusts*	*subpops*	mean	σ	mean	σ	mean	σ
Single Simple FM	1	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.14	0.01	0.23	0.00	0.30	0.00
	2	*(50,350)*	*CCCES*	*50*	*45200*	*rest*	*10*	*2*	0.14	0.02	0.23	0.00	0.30	0.00
	3	-	*RAND*	-	*70000*	-	-	-	0.27	0.04	0.35	0.06	0.40	0.05
Double Simple FM	4	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.09	0.02	0.13	0.02	0.19	0.02
	5	*(50,350)*	*CCCES*	*50*	*45200*	*rest*	*10*	*2*	0.06	0.01	0.11	0.03	0.17	0.05
	6	-	*RAND*	-	*70000*	-	-	-	0.28	0.02	0.39	0.03	0.42	0.03
Triple Simple FM	7	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.06	0.01	0.11	0.01	0.15	0.02
	8	*(30,210)*	*CCCES*	*50*	*64206*	*rest*	*6*	*3*	0.05	0.01	0.10	0.02	0.12	0.02
	9	-	*RAND*	-	*70000*	-	-	-	0.38	0.01	0.48	0.03	0.50	0.03

Table 7.10: Additive-subtractive target tone results

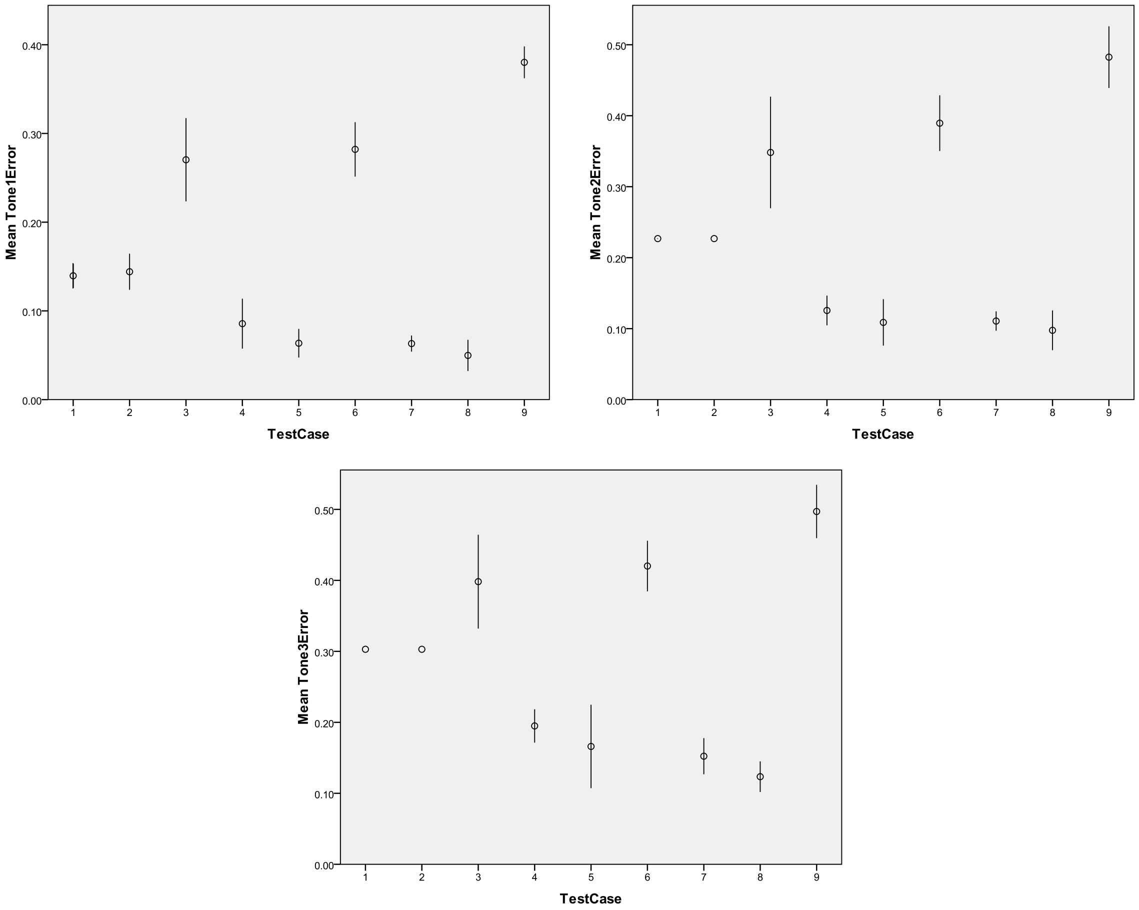


Figure 7.14: Mean and 95% confidence intervals for error when matching additive-subtractive target tones

The CCCES and CES often produced equivalent results; however, when the mean error was different, the CCCES was found to be consistently more accurate in fewer fitness evaluations, although the difference was not statistically significant. When matching with the most simple synthesiser model, the CES and CCCES error results are equivalent (significant at the 0.05 level) and both algorithms converge to the same optima, which may be the global optimum for all three target tones. For the evolutionary algorithms, there is an overall decrease in match error as the complexity of the matching synthesiser increases.

These results therefore exhibit the opposite characteristic to those produced by the RAND algorithm and, more interestingly, the contrived matching experiments of section 7.6.1.

Previously, it was shown that when both the target and match are created by the same synthesis model, the average relative error increases with the complexity of the synthesis model, whereas here, the error decreases with the larger synthesis model. These results illustrate the opposing limitations of the matching process. As shown in table 7.3, the CES is well suited to the problem domain of the single simple FM model. However, in attempting to match target tone one, the CES is likely to have reached the limitations of the matching synthesiser. It would not be possible to improve on this match until a more elaborate matching synthesiser is employed. Indeed this is evidenced by the smaller errors achieved when the double and triple parallel simple FM matching models are used. In the previous experimentation with contrived targets, the CES and CCCES were shown to be less effective at retrieving accurate matches as the model complexity was raised, but clearly, when the limitations of a simple matching synthesiser are reached, more elaborate FM synthesis structures are beneficial.

7.6.2.2 Evolutionary Synthesis Matching of Acoustic Instrument Tones

The experimentation performed above is now repeated, substituting the synthetic additive-subtractive tones with real tones extracted from the sustain (relatively stable, middle section) of three acoustic instrument sounds. The target tones originate from an oboe, trumpet and muted trumpet produced by Opolko and Wapnick (1989). Details of the three tones are provided in table 7.11, and their corresponding spectra are plotted in figure 7.15.

	Pitch	***frequency (Hz)***
Oboe	G5	783.99
Trumpet	C5	523.25
Muted Trumpet	F5	698.46

Table 7.11: Acoustic target fundamental frequencies

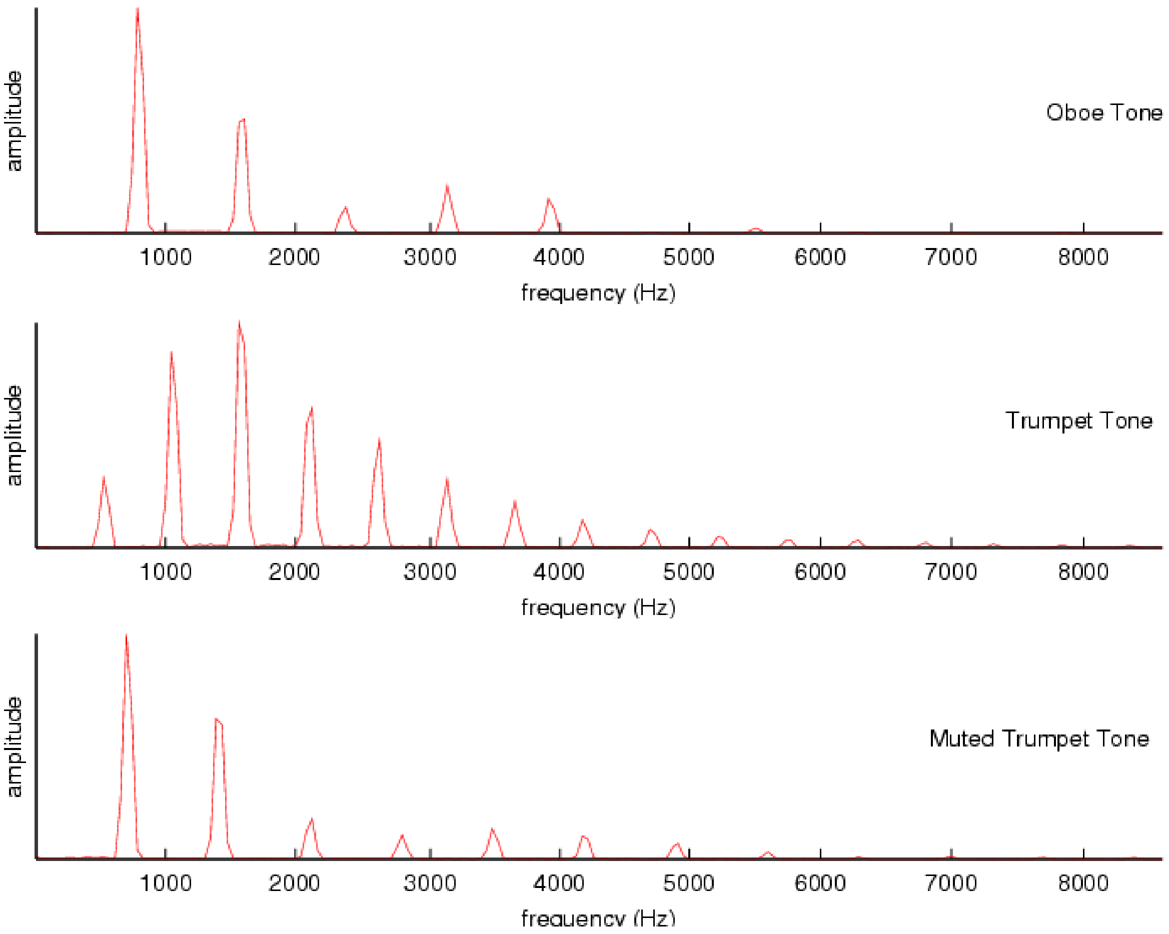


Figure 7.15: Acoustic target spectra

Acoustic Target Matching Results

As in the previous section, the CES, CCCES and RAND algorithms are applied to optimise the synthesis parameters of the FM models depicted in figure 7.1 to match the acoustic target tones plotted in figure 7.15. Results are provided in table 7.12 and plotted in figure 7.16 in terms of the relative spectral error, averaged over five runs, for each configuration.

Matching Model	***Test Case***								***Error***					
		Algorithmic Parameters							Oboe		Trumpet		Muted Trumpet	
		(strat)	*algo*	*gens*	*evals*	*sel*	*clusts*	*subpops*	mean	σ	mean	σ	mean	σ
Single Simple FM	1	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.22	0.00	0.19	0.03	0.18	0.01
	2	*(50,350)*	*CCCES*	*50*	*45200*	*rest*	*10*	*2*	0.24	0.01	0.18	0.03	0.26	0.08
	3	*-*	*RAND*	*-*	*70000*	*-*	*-*	*-*	0.37	0.05	0.44	0.05	0.41	0.06
Double Simple FM	4	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.14	0.01	0.15	0.03	0.14	0.02
	5	*(50,350)*	*CCCES*	*50*	*45200*	*rest*	*10*	*2*	0.11	0.02	0.14	0.03	0.12	0.02
	6	*-*	*RAND*	*-*	*70000*	*-*	*-*	*-*	0.39	0.06	0.54	0.05	0.42	0.04
Triple Simple FM	7	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.12	0.01	0.13	0.02	0.13	0.02
	8	*(30,210)*	*CCCES*	*50*	*64206*	*rest*	*6*	*3*	0.07	0.03	0.13	0.01	0.08	0.03
	9	*-*	*RAND*	*-*	*70000*	*-*	*-*	*-*	0.49	0.06	0.62	0.05	0.52	0.05

Table 7.12: Acoustic target tone matching results

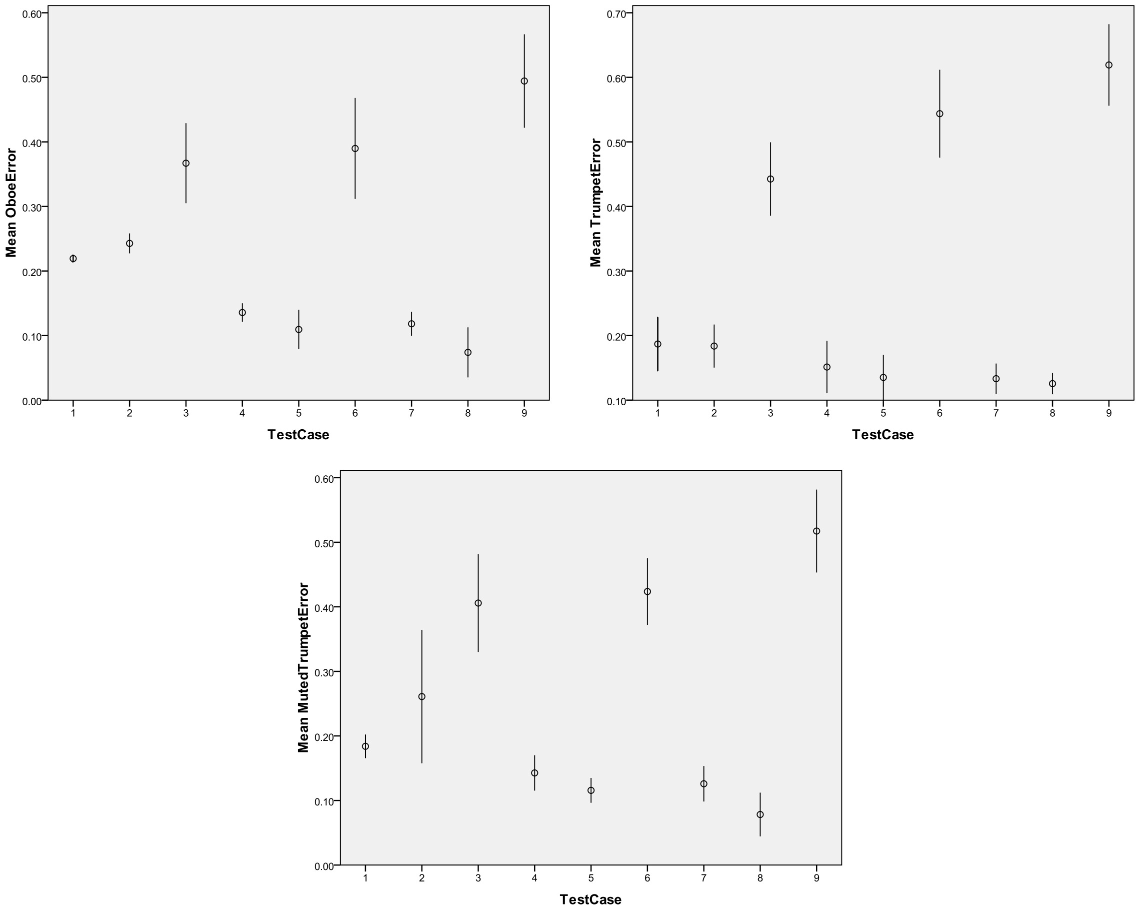


Figure 7.16: Mean and 95% confidence intervals for error when matching acoustic target tones

Discussion

The results from these experiments further corroborate the experiments recorded in the previous section:

- the evolutionary algorithms outperform the random algorithm, with a statistical significant difference between the mean errors of two types of algorithm.
- for the double and triple simple FM models, the CCCES consistently produces a lower mean error than the CES in fewer fitness evaluations, although the differences between the mean errors produced by the two algorithms are not statistically significant.
- the double and triple simple FM models consistently match the target tones more accurately than the single simple FM model, with mean error differences significant at the 0.05 level.

These tests also confirm that the matching system is able to produce good simulations of non-contrived target tones. Figure 7.17 shows the spectrum of the muted trumpet and an example match synthesised by the triple simple FM model. The parameters here were optimised by the CCCES.

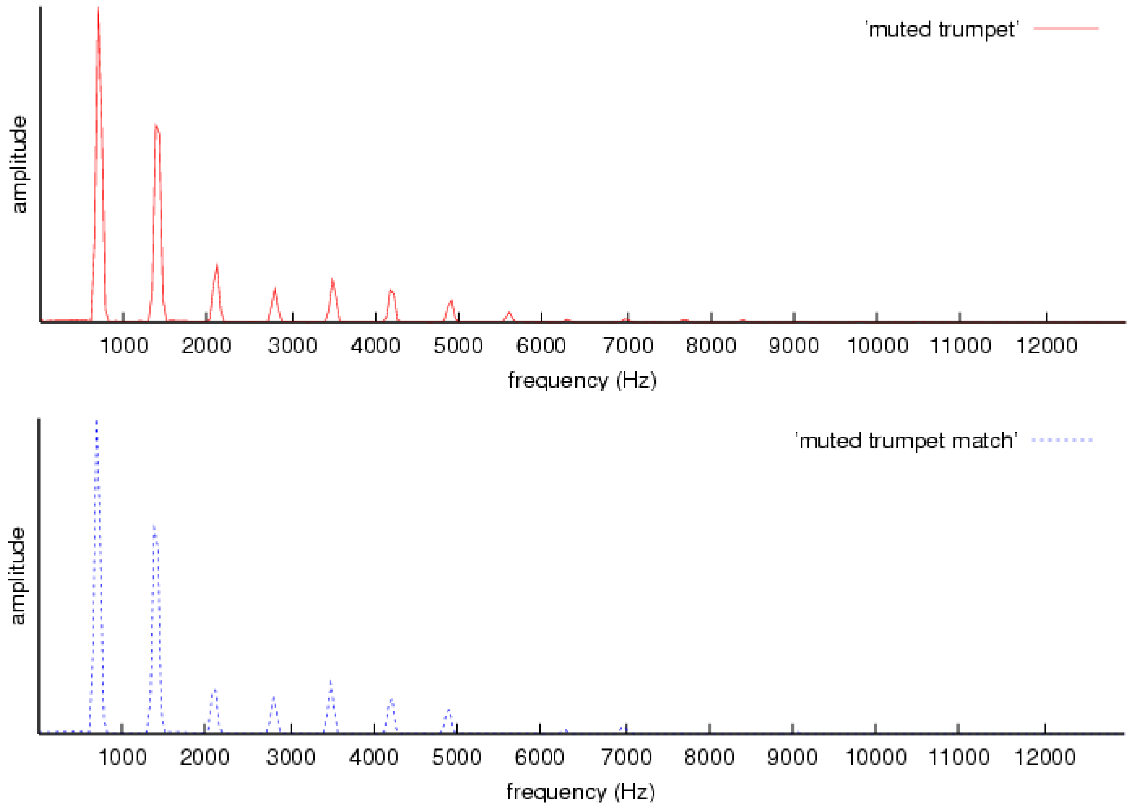


Figure 7.17: Muted trumpet tone (top), and corresponding match (bottom)

All partials towards the lower end of the spectrum are quite accurately matched by the system, with only minor differences in the higher frequency components. The differences are more apparent in figure 7.18, in which both the target and matched spectra are overlaid on a log amplitude scale.

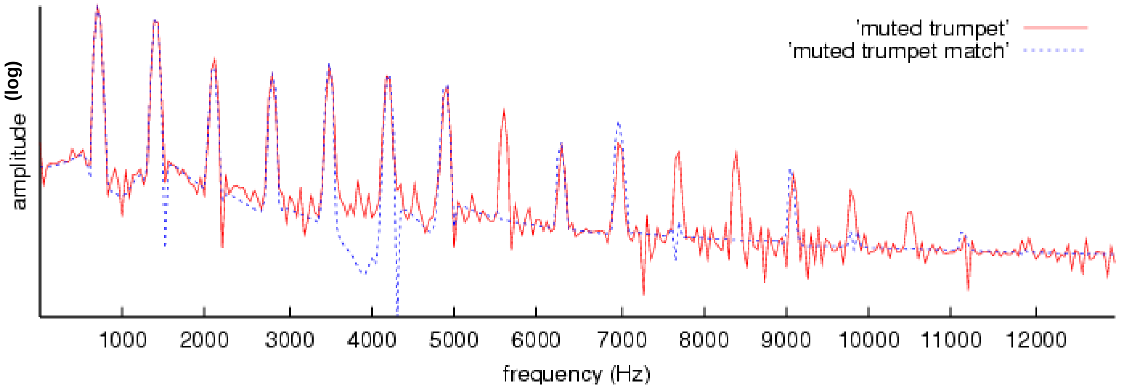


Figure 7.18: Muted trumpet tone and corresponding match with log amplitude

Figure 7.18 enables the limitations of the match to be more easily observed. All of the peaks in the matched tone coincide with partials of the target spectrum, matching the amplitudes and frequencies accurately. However, several higher-frequency harmonic partials (eighth, 11th, 12th and 15th harmonics) are absent or poorly rendered in the matched tone. Since fitness is computed from the spectral difference between target tones and their match candidates, partials with the highest magnitude are prioritised by the matching algorithm, as they have the largest impact on fitness. Perhaps the spectrum error could be reduced further by using a more elaborate matching synthesiser. It is clear, however, that the ability of the EA to effectively navigate the synthesis space diminishes as the complexity of the search space is increased, while the EA parameters remain constant.

7.7 A Performance Analysis of Evolutionary Time-Varying Matching

In this section, the experimental work performed above is developed, expanding the application to include time-variant sounds. For this purpose, the dynamic-sound FM synthesis models of figure 7.2 are employed. The performance of each evolutionary algorithm is assessed and compared according to the contrived matching method introduced earlier in this chapter. Experimentation is then extended further to include the matching of non-contrived dynamic sounds originating from recordings of the trumpet, French horn and oboe.

7.7.1 Experiments with Time-Varying Contrived Targets

In this first set of experiments, each evolutionary algorithm is applied to match 30 randomly generated contrived target sounds using the three simple FM synthesis models illustrated in figure 7.2. Each contrived target sound is created by drawing synthesis parameters at random from within the object range of each parameter (table 7.1).

The parameters for each EA are again annotated in the results according to the following format:

(strat)	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*

For a fuller description of each attribute, see section 7.6.1.

Performance Criteria

As before, results are presented for the single, double and triple FM matching models, according to the following criteria:

- ***Success* -** the number of runs in which the contrived targets are accurately retrieved.
- ***Average*** - the mean average relative spectrum error of the best solutions located for each of the 30 runs.
- ***Remaining*** - the mean average of the best solutions for all unsuccessful matches.

Matches are again classed as successful when at least one population member achieves a relative spectral error below 0.01. That is, the spectra of the target sound are 99% matched.

7.7.1.1 Contrived Matching with Time-Varying Single Simple FM

The time-varying single simple FM model constitutes the most fundamental time-varying synthesis structure described originally by Chowning (1973). This two-oscillator model presents a 12-parameter arrangement as depicted in figure 7.2a. The author has previously examined evolutionary sound matching with this architecture in Mitchell and Sullivan (2005). The data retrieved from the contrived sound matching experimentation with this model is provided in table 7.13 and plotted in figure 7.19.

										Single Simple FM (n = 12)				
		Algorithmic Parameters									Average		Remaining	
Algorithm		*(strat)*	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*	Success	mean	σ	mean	σ
CCES	1	(20,140)	CCES1	dis	50	84240	glob	-	12	6	0.13	0.15	0.17	0.15
	2	(20,140)	CCES2	dis	25	84240	glob	-	12	13	0.04	0.04	0.06	0.04
ES	3	(200,1400)	ES	dis	50	70200	glob	-	-	11	0.03	0.03	0.04	0.03
	4	(200,1400)	ES	int	50	70200	glob	-	-	0	0.27	0.15	0.27	0.15
MSES	5	1400 x (1+1)	MSES	-	50	70000	-	-	-	0	0.15	0.08	0.15	0.08
	6	350 x (1+1)	MSES	-	200	70000	-	-	-	0	0.13	0.08	0.13	0.08
	7	175 x (1+1)	MSES	-	400	70000	-	-	-	0	0.13	0.08	0.13	0.08
	8	100 x (1+1)	MSES	-	700	70000	-	-	-	0	0.13	0.09	0.13	0.09
FCES	9	(200,1400)	FCES	dis	50	70200	glob	40	-	12	0.03	0.04	0.04	0.04
	10	(200,1400)	FCES	int	50	70200	glob	40	-	0	0.25	0.13	0.25	0.13
CES	11	(200,1400)	CES	dis	50	70200	rest	40	-	3	0.03	0.03	0.04	0.03
	12	(200,1400)	CES	cent	50	70200	rest	40	-	2	0.08	0.06	0.09	0.06
CCCES	13	(50,350)	CCCES	dis	50	45200	rest	10	2	10	0.02	0.02	0.03	0.02
	14	(50,350)	CCCES	cent	50	45200	rest	10	2	2	0.04	0.03	0.04	0.03
	15	(50,350)	CCCES	dis	100	90200	rest	10	2	21	0.01	0.02	0.03	0.02
	16	(50,350)	CCCES	cent	100	90200	rest	10	2	9	0.03	0.03	0.04	0.03

Table 7.13: Results when matching time-varying single simple FM contrived sounds

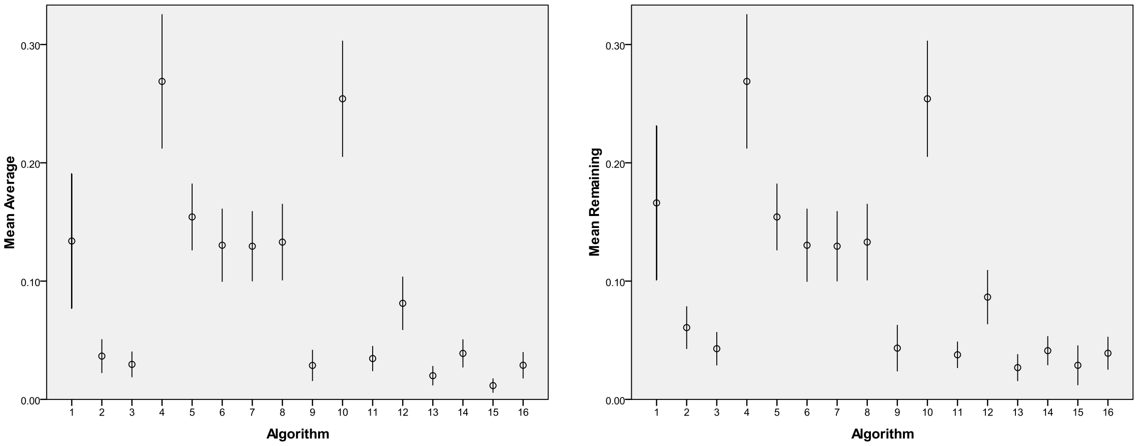


Figure 7.19: Mean and 95% confidence intervals for Average and Remaining error results when matching time-varying single simple FM contrived sounds

Discussion

Overall, there are fewer successful matches when compared with the single simple FM results of the static tone experimentation, and again, the discrete recombination operator is found to consistently outperform the intermediate and centroid techniques. T-tests between the results of the recombining algorithms support this observation with a statistically significant (two-tailed) improvement when discrete recombination is employed.

As observed in earlier experimentation, the CCES performs best when the *simple best plus one random* collaboration strategy is adopted (CCES2). Interestingly, if the average error delivered by both variants of the CCES is compared with the equivalent results of the static tone experiments (table 7.3), a statistically significant (two-tailed) improvement in performance is observed, despite the considerable increase in the dimensionality of the problem space. This suggests that the time-varying domain presents characteristics that benefit the CCES. This improvement is also statistically significant in the results produced by the ES which also demonstrates a considerable improvement in this domain. It may be that the temporal parameters are easy to retrieve, and much of the problem complexity is introduced by the carrier/modulator frequency parameters and the modulation index. The multiple spectral snapshots then provide the optimiser with multiple samples through which these principal parameters may be retrieved, subsequent optimisation of the temporal parameters is then a simple task.

The remaining EAs produce results that partially reflect those seen in the static tone experiments. A post-hock ANOVA revealed that the MSES was significantly less effective than at least the best performing instance of all other algorithms. The analyses also indicated the discreet recombination CCCES (algorithm 13) produced the smallest mean error and was thus more effective than all other types of algorithm in fewer fitness evaluations, significant at the 0.05 level against all but algorithms two, three, nine and 11. Interestingly, in this instance, the CES produces lower average errors and makes less successful retrievals than the FCES. As was shown in chapter four, the hard speciation architecture of the CES algorithm prolongs convergence. The low average error suggests that, with increased generations, the CES would make further successful retrievals. Indeed, running the CES for 100 generations resulted in 16 successful retrievals and a remaining error of 0.031.

7.7.1.2 Contrived Matching with Time-Varying Double Simple FM

The time-varying double simple FM model (figure 7.2b) presents a 24-dimensional synthesis matching problem domain. This model forms a parallel extension of the single simple FM arrangement and is constructed from four sinusoidal oscillators. The results of the contrived-target experimentation performed with this model are provided in table 7.14 and plotted in figure 7.20.

										Single Simple FM (n = 24)				
		Algorithmic Parameters									Average		Remaining	
Algorithm		*(strat)*	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*	Success	mean	σ	mean	σ
CCES	*1*	*(10,70)*	*CCES1*	*dis*	*50*	*84240*	*glob*	*-*	*24*	4	0.08	0.09	0.09	0.09
	2	*(10,70)*	*CCES2*	*dis*	*25*	*84240*	*glob*	*-*	*24*	7	0.06	0.09	0.08	0.10
ES	3	*(200,1400)*	*ES*	*dis*	*50*	*70200*	*glob*	*-*	*-*	0	0.08	0.07	0.08	0.07
	4	*(200,1400)*	*ES*	*int*	*50*	*70200*	*glob*	*-*	*-*	0	0.29	0.14	0.29	0.14
MSES	5	*1400 x (1+1)*	*MSES*	*-*	*50*	*70000*	*-*	*-*	*-*	0	0.22	0.10	0.22	0.10
	6	*350 x (1+1)*	*MSES*	*-*	*200*	*70000*	*-*	*-*	*-*	0	0.18	0.10	0.18	0.10
	7	*175 x (1+1)*	*MSES*	*-*	*400*	*70000*	*-*	*-*	*-*	0	0.18	0.10	0.18	0.10
	8	*100 x (1+1)*	*MSES*	*-*	*700*	*70000*	*-*	*-*	*-*	0	0.18	0.11	0.18	0.11
FCES	9	*(200,1400)*	*FCES*	*dis*	*50*	*70200*	*glob*	*40*	*-*	0	0.09	0.08	0.09	0.08
	10	*(200,1400)*	*FCES*	*int*	*50*	*70200*	*glob*	*40*	*-*	0	0.28	0.12	0.28	0.12
CES	11	*(200,1400)*	*CES*	*dis*	*50*	*70200*	*rest*	*40*	*-*	0	0.10	0.08	0.10	0.08
	12	*(200,1400)*	*CES*	*cent*	*50*	*70200*	*rest*	*40*	*-*	0	0.14	0.10	0.14	0.10
CCCES	13	*(50,350)*	*CCCES*	*dis*	*50*	*45200*	*rest*	*10*	*2*	0	0.06	0.06	0.06	0.06
	14	*(50,350)*	*CCCES*	*cent*	*50*	*45200*	*rest*	*10*	*2*	0	0.08	0.07	0.08	0.07
	15	*(50,350)*	*CCCES*	*dis*	*100*	*90200*	*rest*	*10*	*2*	8	0.04	0.06	0.05	0.06
	16	*(50,350)*	*CCCES*	*cent*	*100*	*90200*	*rest*	*10*	*2*	2	0.06	0.06	0.06	0.06

Table 7.14: Results when matching time-varying double simple FM contrived sounds

Discussion

These results again indicate that the CCES produces a mean error value that that tends towards, and frequently outperforms the most successful algorithms when matching dynamic sounds. This improved matching performance, however, is not unique to the CCES; the ES with discrete recombination performs significantly better on the time-varying target sound problem than the equivalent static tone problem tested earlier – statistically significant mean difference of 0.24 (t-test).

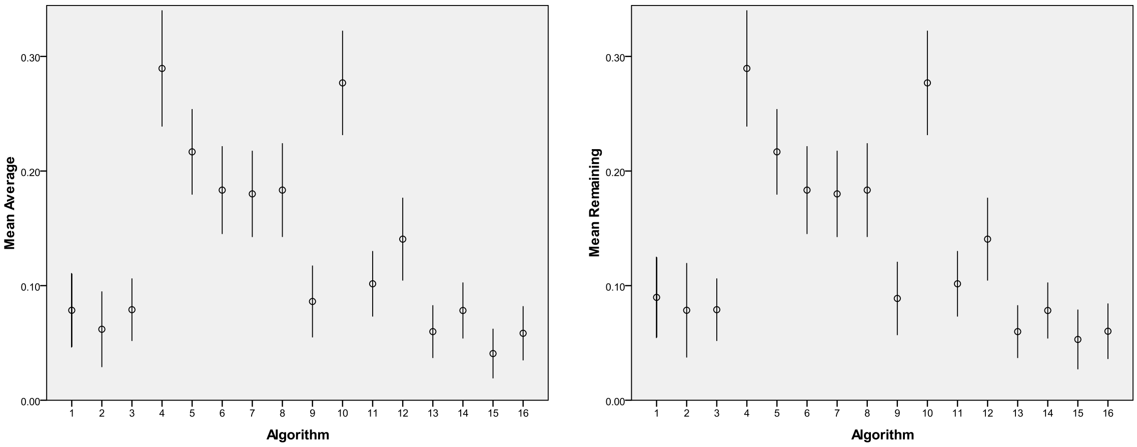


Figure 7.20: Mean and 95% confidence intervals for Average and Remaining error results when matching time-varying double simple FM contrived sounds

As the ES is more susceptible to becoming trapped at local optima than the niching-based algorithms, this result again suggests that the time-variant FM search space is more tractable than the equivalent static tone space. This difference in search space characteristics may also explain the improved CCES performance. However, the poor performance of the MSES, combined with the convergence of each CES and CCCES cluster to an independent niche (not shown), suggests that there are still many potential matches for each target sound, and thus the matching space is extensively multimodal.

If high-fitness solutions are easier to locate, why is it that the dynamic search space is more tractable than the static space? As discussed previously, the introduction of the time dimension must introduce search space characteristics that are beneficial to the EAs. Time-variation in the target sound is sampled by taking multiple spectrum snapshots through time. Error is then averaged across difference measurements between the target and candidate spectra. In the static experiments, there is only one target snapshot with which to perform the match. The averaging of multiple snapshots for dynamic sounds must, then, enable population members positioned closer to the optimum to be selected more easily. While the CCES2 retrieves the highest number of contrived sounds, CCCES also performs well achieving the lowest overall mean error (albeit in the highest number of evaluations). A post hock ANOVA supports this observation, confirming that the CCCES produces a lower mean error than algorithms four, five, six, seven, eight, 10 and 12 at the 0.05 level.

7.7.1.3 Contrived Matching with Time-Varying Triple Simple FM

The contrived matching experimentation is repeated for the triple simple FM model, depicted in figure 7.2c. This adds an additional pair of oscillators to the previously tested model, increasing the dimensionality of the search domain to 36. Results are provided in table 7.15 and plotted in figure 7.21.

										Single Simple FM (n = 36)				
		Algorithmic Parameters									Average		Remaining	
Algorithm		*(strat)*	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*	Success	mean	σ	mean	σ
CCES	*1*	*(6,42)*	*CCES1*	*dis*	*50*	*75816*	*glob*	-	*36*	4	0.10	0.09	0.12	0.09
	2	*(6,42)*	*CCES2*	*dis*	*25*	*75816*	*glob*	-	*36*	6	0.08	0.09	0.10	0.09
ES	3	*(200,1400)*	*ES*	*dis*	*50*	*70200*	*glob*	-	-	0	0.14	0.08	0.14	0.08
	4	*(200,1400)*	*ES*	*int*	*50*	*70200*	*glob*	-	-	0	0.32	0.10	0.32	0.10
MSES	5	*1400 x (1+1)*	*MSES*	-	*50*	*70000*	-	-	-	0	0.27	0.09	0.27	0.09
	6	*350 x (1+1)*	*MSES*	-	*200*	*70000*	-	-	-	0	0.24	0.08	0.24	0.08
	7	*175 x (1+1)*	*MSES*	-	*400*	*70000*	-	-	-	0	0.24	0.08	0.24	0.08
	8	*100 x (1+1)*	*MSES*	-	*700*	*70000*	-	-	-	0	0.25	0.09	0.25	0.09
FCES	9	*(200,1400)*	*FCES*	*dis*	*50*	*70200*	*glob*	*40*	-	0	0.13	0.08	0.13	0.08
	10	*(200,1400)*	*FCES*	*int*	*50*	*70200*	*glob*	*40*	-	0	0.31	0.11	0.31	0.11
CES	11	*(200,1400)*	*CES*	*dis*	*50*	*70200*	*rest*	*40*	-	0	0.14	0.07	0.14	0.07
	12	*(200,1400)*	*CES*	*cent*	*50*	*70200*	*rest*	*40*	-	0	0.20	0.09	0.20	0.09
CCCES	13	*(30,210)*	*CCCES*	*dis*	*50*	*64206*	*rest*	*10*	*3*	0	0.11	0.06	0.11	0.06
	14	*(30,210)*	*CCCES*	*cent*	*50*	*64206*	*rest*	*10*	*3*	0	0.15	0.08	0.15	0.08
	15	(30,210)	CCCES	*dis*	100	128106	*rest*	10	3	0	0.07	0.05	0.07	0.05
	16	(30,210)	CCCES	*cent*	100	128106	*rest*	10	3	0	0.12	0.07	0.12	0.07

Table 7.15: Results when matching time-varying triple simple FM contrived sounds

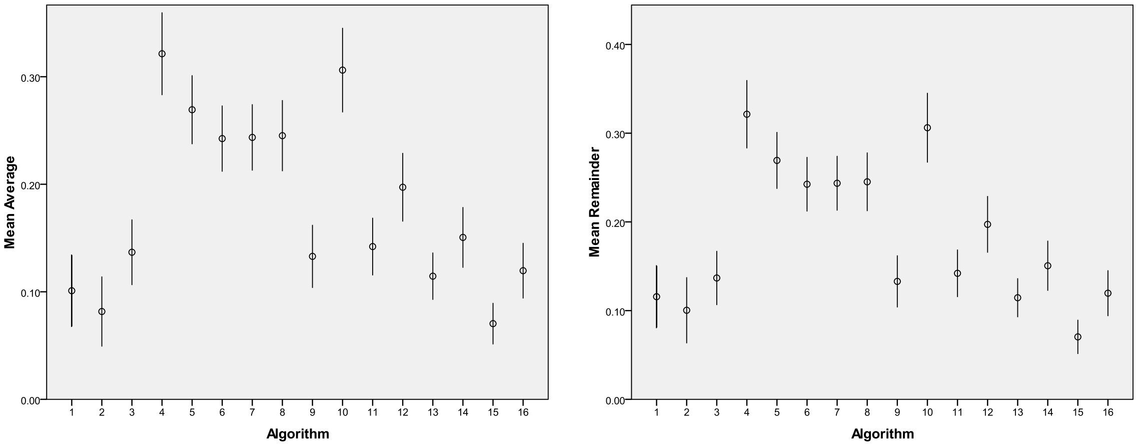


Figure 7.21: Mean and 95% confidence intervals for Average and Remaining error results when matching time-varying triple simple FM contrived sounds

Discussion

The results on the largest FM model exhibit a similar trend to the previous experiments, albeit with slightly larger error values. As before the CCES and CCCES are most effective at successfully retrieving matches from the FM search space. The CCES also appears to be relatively robust to the reduction in subpopulation size that occurs due to the increased problem dimensionality.

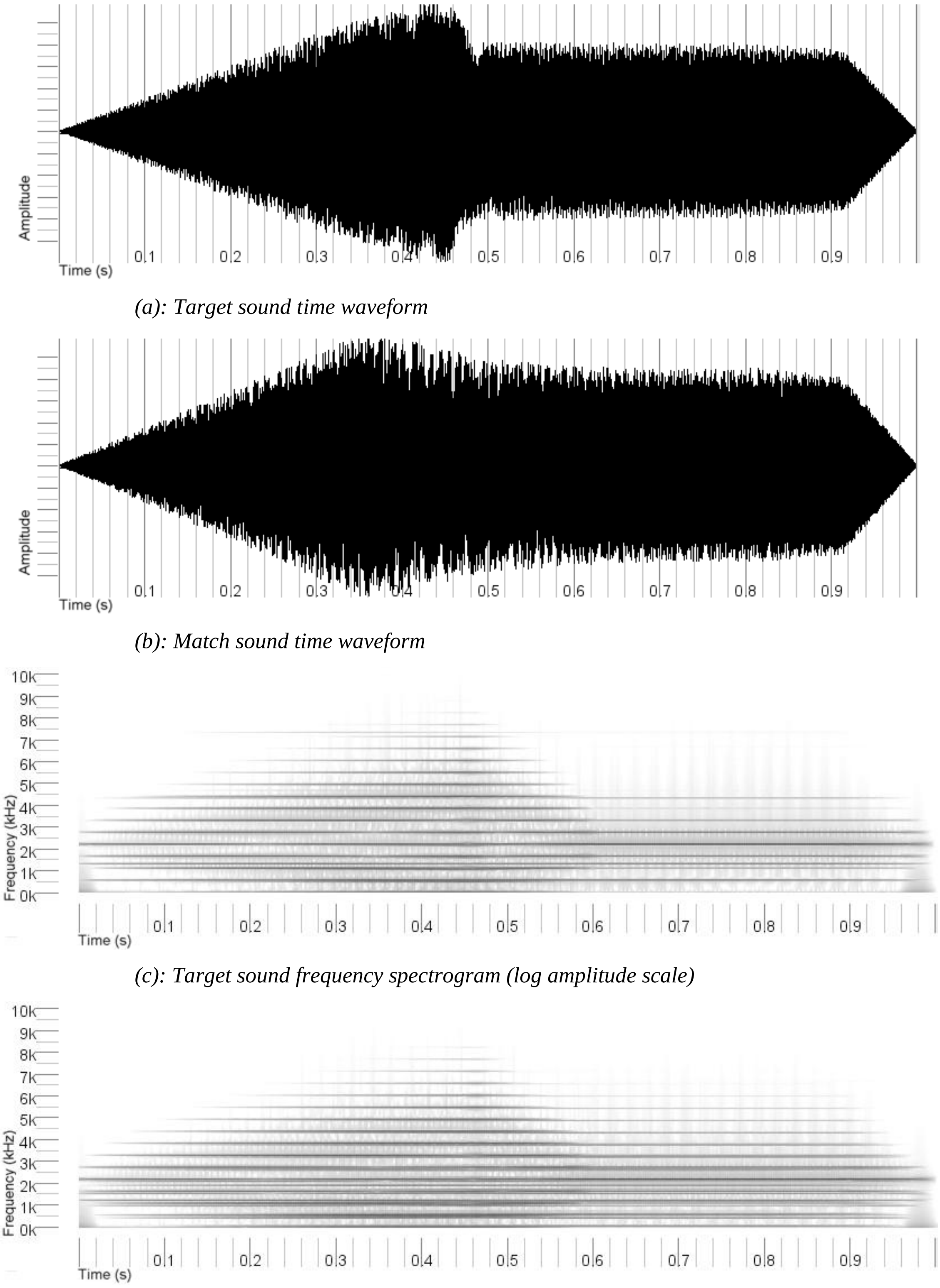
For the triple simple FM model, only the CCES-based algorithms are able to retrieve matches successfully. For the first time, the CCES1 has outperformed the CCES2 in this matching domain, although their mean errors are not statistically different. It is presumed that this is principally because the CCES2 runs for only half the number of rounds as the CCES1 in the same number of fitness evaluations. Had the CCES2 been able to continue, the advantage of the *single best plus one random* strategy would be clear.

Overall the smallest error results are achieved by the CCCES, where comparisons by post hock ANOVA revealed a statistically significant performance advantage over algorithms four, five, six, seven, eight, 10, 11, 12 and 14. However, in order to achieve these low-error ratings, the CCCES required more fitness evaluations than the other algorithms.

7.7.1.4 Time Waveform and Frequency Spectrogram Plots with Contrived Targets

Figure 7.22 provides time waveforms and frequency spectrograms for an example of contrived target sound and its corresponding match produced with the time-varying triple simple FM model and optimised by the CCES. The spectrogram enables time variation to be visualised in terms of the signal’s spectral content. The frequency content of the signal is plotted against time, and the amplitude is indicated by the darkness of the trace. The relative spectral error for this particular match is 0.064.

In the time waveform plots (Figure 7.22a and b) the amplitude envelopes of the sounds are similar: both exhibit long attack, short decay, long sustain and short release periods. However, there are subtle differences. For example, the attack period of the target sound lasts approximately 0.45s, whereas the matched sound appears to reach maximum amplitude at approximately 0.35s. To compensate for this difference, the decay period of the matched sound lasts approximately 0.1s longer than that of the target, as such, the sustain period for both sounds begins at approximately 0.5s.



(a): Target sound time waveform

(b): Match sound time waveform

(c): Target sound frequency spectrogram (log amplitude scale)

(d): Match sound frequency spectrogram (log amplitude scale)

Figure 7.22: Contrived time-varying triple simple FM target and CCES evolved match

Due to the fixed positions of the 10 sample points, it is likely that rapid changes in the target sound are under-sampled in the analysis procedure. As such, these minor differences are present because sample points are positioned either side of this rapid change. At the fixed positions the amplitude envelope may correlate well, however the changes that occur in between the sample points are interpolated by the matching process, and are not guaranteed to match the original target.

The spectrogram plots (figure 7.22c and d) also correspond well; however again there are subtle differences between the spectral envelopes of the two sounds. For example, the matched sound contains marginally more high-frequency content than the target during the sustain section. Furthermore, the matched sound features a low-amplitude partial at approximately 1.9kHz that does not feature in the target spectrum.

7.7.2 Experiments with Time-Varying Acoustic Targets

In the contrived experimentation above, the algorithms shown to be most effective at exploiting the parameter space of the time-varying matching synthesiser were the CCES and CCCES. In this section, these two algorithms plus the CES and RAND derivatives of the time-varying FM matching system are used to optimise matches to non-FM dynamic sounds. The target set is comprised of three acoustic instrument samples produced by Opolko and Warpnick (1989): muted French horn, trumpet and oboe. Details of each sound are provided in the table 7.16. All EAs optimise a match for each target sound using the dynamic-spectra FM synthesis models depicted in figure 7.2. The mean relative spectral error of the best individual for each test case is provided in table 7.17 and plotted in figure 7.23; results are computed from the average error of five independent and randomly initialised runs.

Target sound	***Pitch***	***frequency (Hz)***
Muted French Horn	D5	587.33
Trumpet	F5	698.46
Oboe	F#5	739.99

Table 7.16: Acoustic target fundamental frequencies

Discussion

When matching dynamic acoustic target sounds with FM synthesis, the evolutionary algorithms show a statistically significant performance advantage over the random search algorithm. Within the EA-based optimisers, there are no consistent findings that indicate the superiority of one particular type of algorithm.

									Target Tone					
Matching Model	***Test case***	***Algorithmic Parameters***							French Horn		Trumpet		Oboe	
		(strat)	*algo*	*gens*	*Evals*	*sel*	*clusts*	*subpops*	mean	σ	mean	σ	mean	σ
Single Simple FM	1	*(20,140)*	*CCES2*	*25*	*84240*	*glob*	*-*	*12*	0.19	0.04	0.21	0.00	0.15	0.00
	2	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.16	0.00	0.21	0.01	0.15	0.00
	3	*(50,350)*	*CCCES*	*50*	*45200*	*rest*	*10*	*2*	0.16	0.00	0.20	0.00	0.15	0.00
	4	*-*	*RAND*		*70000*	*-*	*-*	*-*	0.31	0.03	0.31	0.03	0.27	0.01
Double Simple FM	5	*(10,70)*	*CCES2*	*25*	*84240*	*glob*	*-*	*24*	0.07	0.01	0.11	0.05	0.07	0.01
	6	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.11	0.02	0.13	0.01	0.10	0.01
	7	*(50,350)*	*CCCES*	*50*	*45200*	*rest*	*10*	*2*	0.11	0.02	0.13	0.01	0.10	0.01
	8	*-*	*RAND*	*-*	*70000*	*-*	*-*	*-*	0.27	0.01	0.27	0.01	0.27	0.01
Triple Simple FM	9	*(6,42)*	*CCES2*	*25*	*75816*	*glob*	*-*	*36*	0.11	0.03	0.13	0.03	0.10	0.03
	10	*(200,1400)*	*CES*	*50*	*70200*	*rest*	*40*	*0*	0.12	0.01	0.13	0.01	0.11	0.01
	11	*(30,210)*	*CCCES*	*50*	*64206*	*rest*	*6*	*3*	0.12	0.02	0.15	0.01	0.11	0.01
	12	*-*	*RAND*	*-*	*70000*	*-*	*-*	*-*	0.28	0.02	0.29	0.01	0.23	0.02

Table 7.17: Acoustic target time-varying matching results

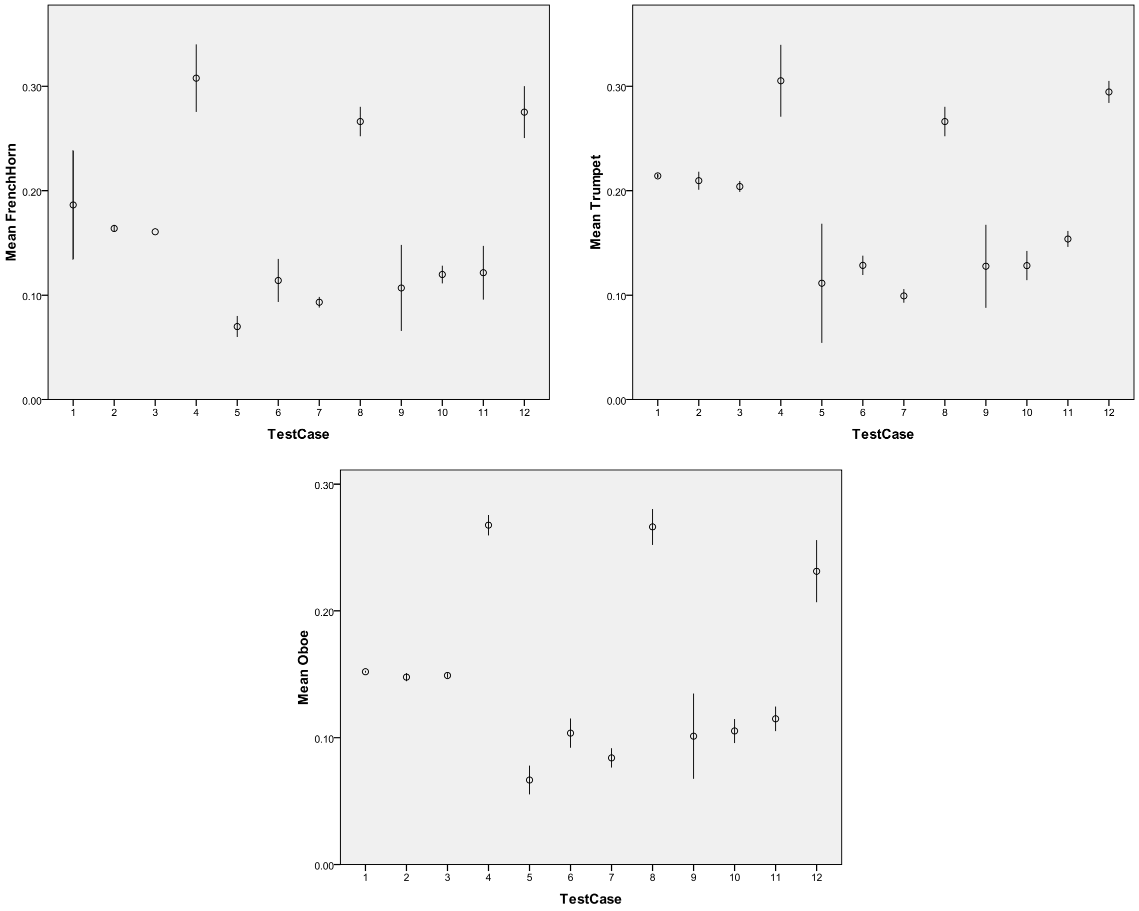


Figure 7.23: Mean and 95% confidence intervals for error when matching acoustic time-varying sounds

When matching with the single simple FM model, the CCCES consistently produced the smallest mean error for all three target sounds in less fitness evaluations and the CCES2 consistently produced the largest error. A post hock comparison of means analysis by ANOVA only detected a statistically significant difference at the 0.05 level between the CCCES and CCES2 on the results of the oboe sound, with no significant difference between any of the other algorithms. This finding suggests that all three EA-based algorithms produce approximately the same error results when matching with the single simple FM model. It is presumed that all EAs converged to the limitations of the matching synthesiser, and in order to produce better results, a more elaborate synthesiser would be required.

Indeed, matches produced with the double FM model result in smaller errors. In contrast to the single simple FM model, on this larger synthesis model the CCES2 consistently produced the smallest error results and the CES consistently produced the largest error. A post hock ANOVA again confirmed this difference to be statistically significant at the 0.05 level for all target sounds. The difference between the CCES2 and CCCES was only significant when matching the French horn; however, the difference was not significant for the other sounds due to the large variance in the CCES2 results.

The reduction in error observed between the matches produced on the single and double simple FM models does not extend further when matches were performed using the triple FM model, where results indicate marginally less accurate matches (although not statistically significant). Again the CCES2 produces the smallest mean error and this time the CCCES produces the largest, however there are no statistically significant differences between the algorithms. It is suggested that the CCES2 advantage is, in part, due to the additional number of fitness evaluations performed by this algorithm, and the faster convergence rates noted earlier in this thesis.

The larger triple FM model is obviously capable of producing more exact matches than the smaller models, but the EAs are unable to exploit this advantage in these tests. This is particularly prevalent in the results produced by the CCCES, and is due to the maintenance of fixed population or scaled subpopulation sizes while the search space dimensionality is increased. This hypothesis can be tested by repeating all three algorithms with population sizes scaled in proportion to the problem dimensionality and by doubling the number generations for which each algorithm runs to allow each algorithm to adequately converge. The results from the average of five runs are tabulated below when matching the oboe sound with the triple simple FM model with all three EAs and the RAND algorithm for comparison; population sizes are indicated with the results below.

	Algorithmic Parameters								***Oboe Target Tone***	
Matching Model	*(strat)*	*algo*	*recomb*	*gens*	*evals*	*sel*	*clusts*	*subpops*	mean error	σ error
Triple Simple FM	*(10,70)*	*CCES2*	*Discrete*	*50*	*252360*	*global*	*-*	*36*	0.07	0.01
	(300,2100)	*CES*	*Discrete*	*100*	*210300*	*restricted*	*60*	*0*	0.08	0.01
	(40,280)	*CCCES*	*Discrete*	*100*	*238744*	*restricted*	*8*	*3*	0.08	0.00
	-	*RAND*	*-*	*-*	*230000*	*-*	*-*	*-*	0.23	0.01

Table 7.18: Oboe matching results

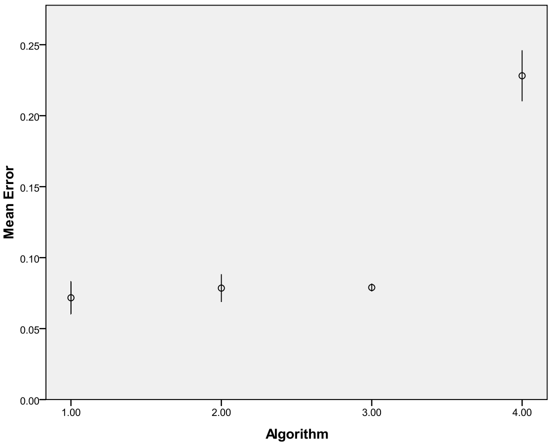
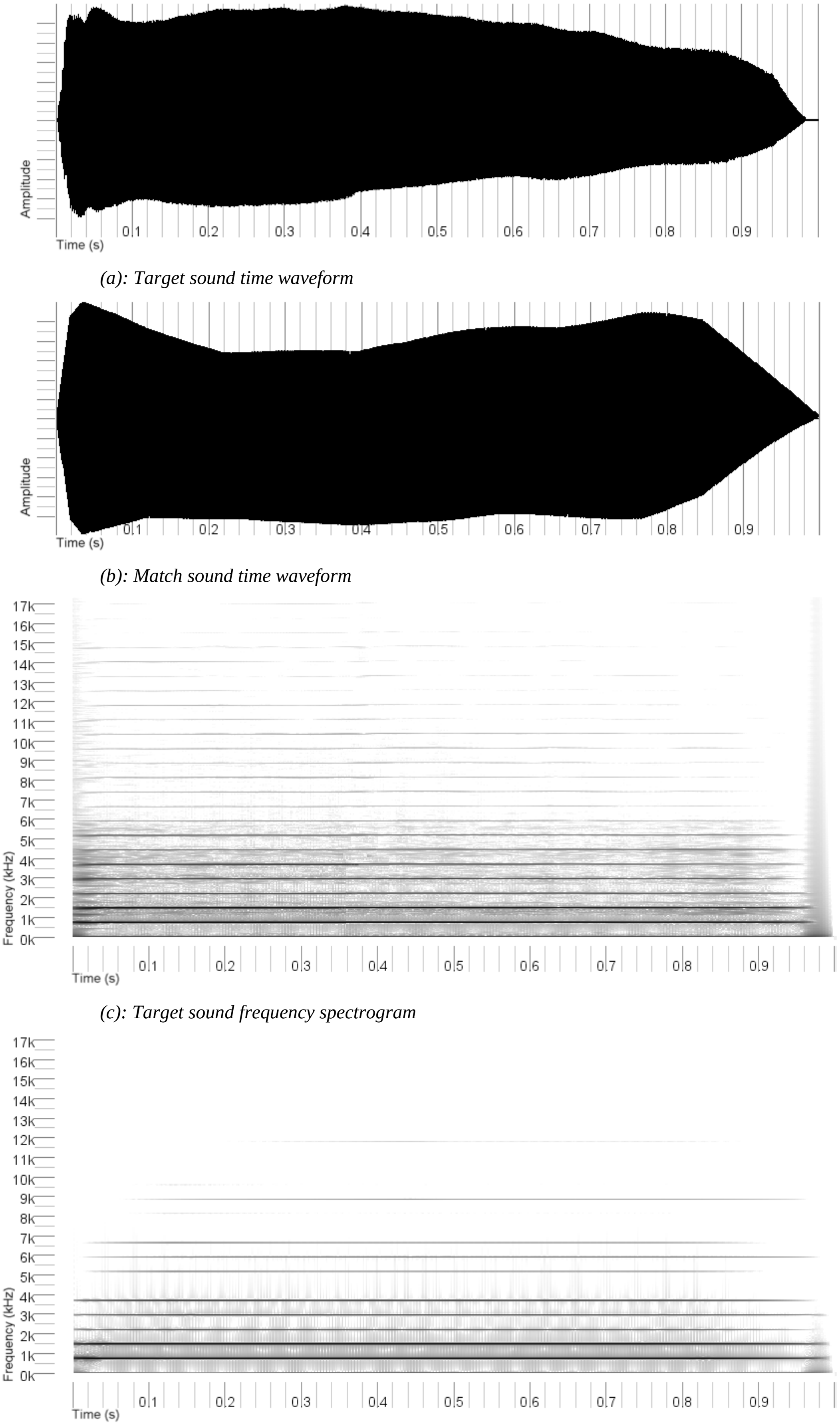


Figure 7.24: Mean and 95% confidence intervals for error when matching an acoustic time-varying oboe sound

These results confirm that optimisation with larger population sizes leads to an improved match.

7.7.2.1 Time Waveform and Frequency Spectrogram Plots with Non-Contrived Targets

The accuracy of the non-contrived time-varying matches may be compared visually by observing time waveform and frequency spectrograms of a specific target sound and its corresponding match. Figure 7.25 plots the F#5 oboe target sound, with a CCCES match using the triple FM synthesis model. This particular example achieves a relative spectral error of 0.078. The general amplitude envelopes of the time waveform plots, while noticeably different, indicate similar characteristics. For example, both sounds have reached full magnitude at 0.04 seconds, and fade to silence at similar rates. However the shape of the sustain tones appear quite different. The real oboe sound exhibits a steady fall in amplitude from approximately 0.4 seconds, while the matched sound amplitude development is different, rising until 0.8 seconds.



(a): Target sound time waveform

(b): Match sound time waveform

(c): Target sound frequency spectrogram

(d): Match sound frequency spectrogram

Figure 7.25: Oboe target sound and time-varying triple simple FM match evolved by CCCES

The frequency spectrograms are plotted on a log amplitude scale to enable the higher frequency, low magnitude harmonics to be compared. Both the target and matched sounds show good correlation in the lower frequency components, with harmonics well matched until the fifth harmonic. The seventh to ninth harmonics are also present, as are some higher harmonics. However, there are some partials absent from the matched sound.

In terms of time variation, the target spectrogram indicates two regions at which the spectrum is shown to be changing. Firstly, the initial onset of the target sounds (0.0 - 0.02 seconds) features many additional partials before settling into a harmonic tone. Secondly, the high frequency partials fade from the target prior to the lower frequency partials. This temporal variation is only partially represented in the matched sound. The exact variation of each partial is better represented in the three-dimensional spectrogram shown in figure 7.26.

To understand why certain features of the target sound are matched and others are overlooked, it is important to establish how each element of the synthesiser contributes to the final oboe match. Figures 7.27 - 7.29 provide output plots from each of the three parallel simple FM elements in isolation. Figure 7.27 plots the time-domain waveforms, figure 7.28 plots the frequency spectrograms, and figure 7.29 plots the long-term average spectrum of the simple FM elements that combine to form the final match.

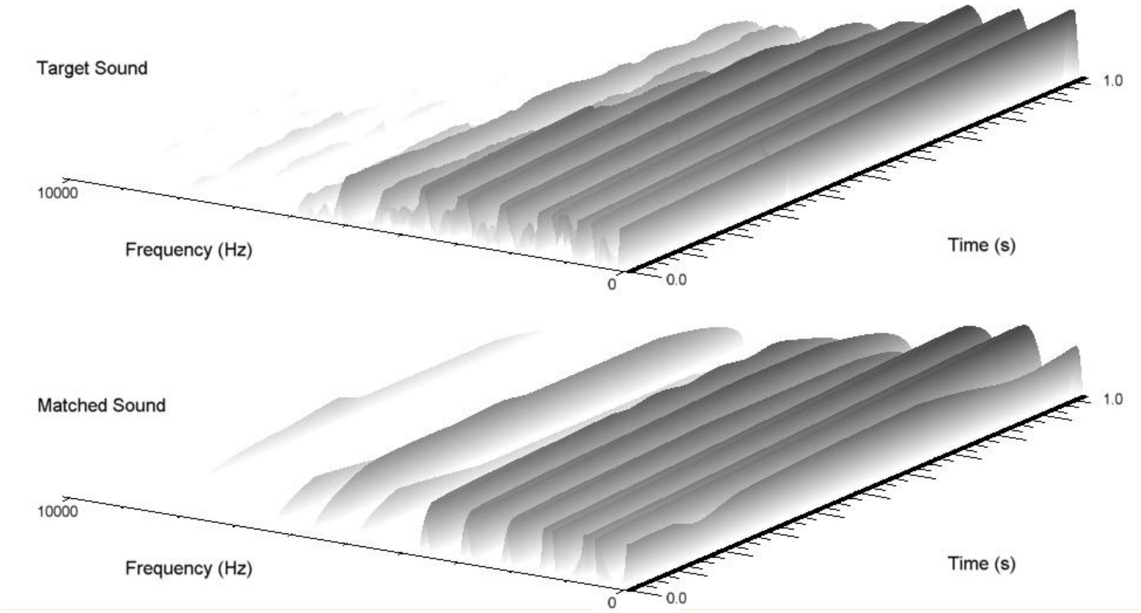


Figure 7.26: 3D- Spectrogram of oboe target sound and CCCES evolved match

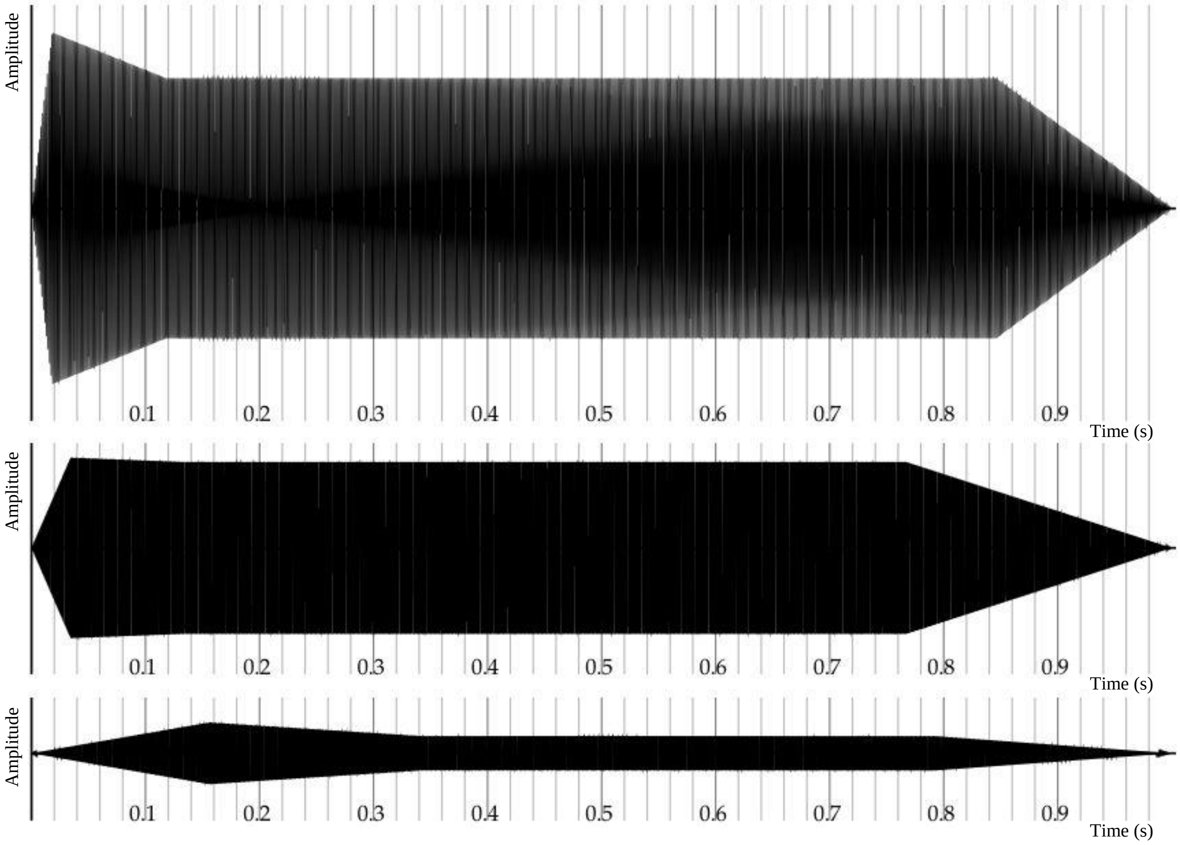


Figure 7.27: FM oboe simulation, isolated time domain waveform produced by simple FM element one (top), element two (middle) and element three (bottom)

Figure 7.27 illustrates the relative contribution that each simple FM element makes to the composite oboe match. In terms of peak amplitude, each element contributes to a decreasing extent. Element one has the most amplitude and also exhibits the fastest attack and release periods. Element three, on the other hand, has the least amplitude and has comparatively long attack, decay and release periods.

Figure 7.28 illustrates that the frequency-bands produced by the three elements are significantly interleaved. Element one positions a large proportion of its partials at the lower end of the spectrum. Element two adds to these lower frequency components, while element three positions its partials farther apart where they extend into the higher frequencies. These relationships are also represented in figure 7.29, which provides the long term average spectrum of each element. The carrier frequencies of each element are centred at the first, second and fourth harmonics of the original oboe sound. The side frequencies produced by element one reflect around 0Hz and coincide with the odd harmonics of the target sound. The second element generates the second harmonic of the target sound, and the first pair of side frequencies contributes to the fifth and, by reflection, the first harmonics. Element three then fills the reaming gaps, providing the fourth, eighth and 12th harmonics.

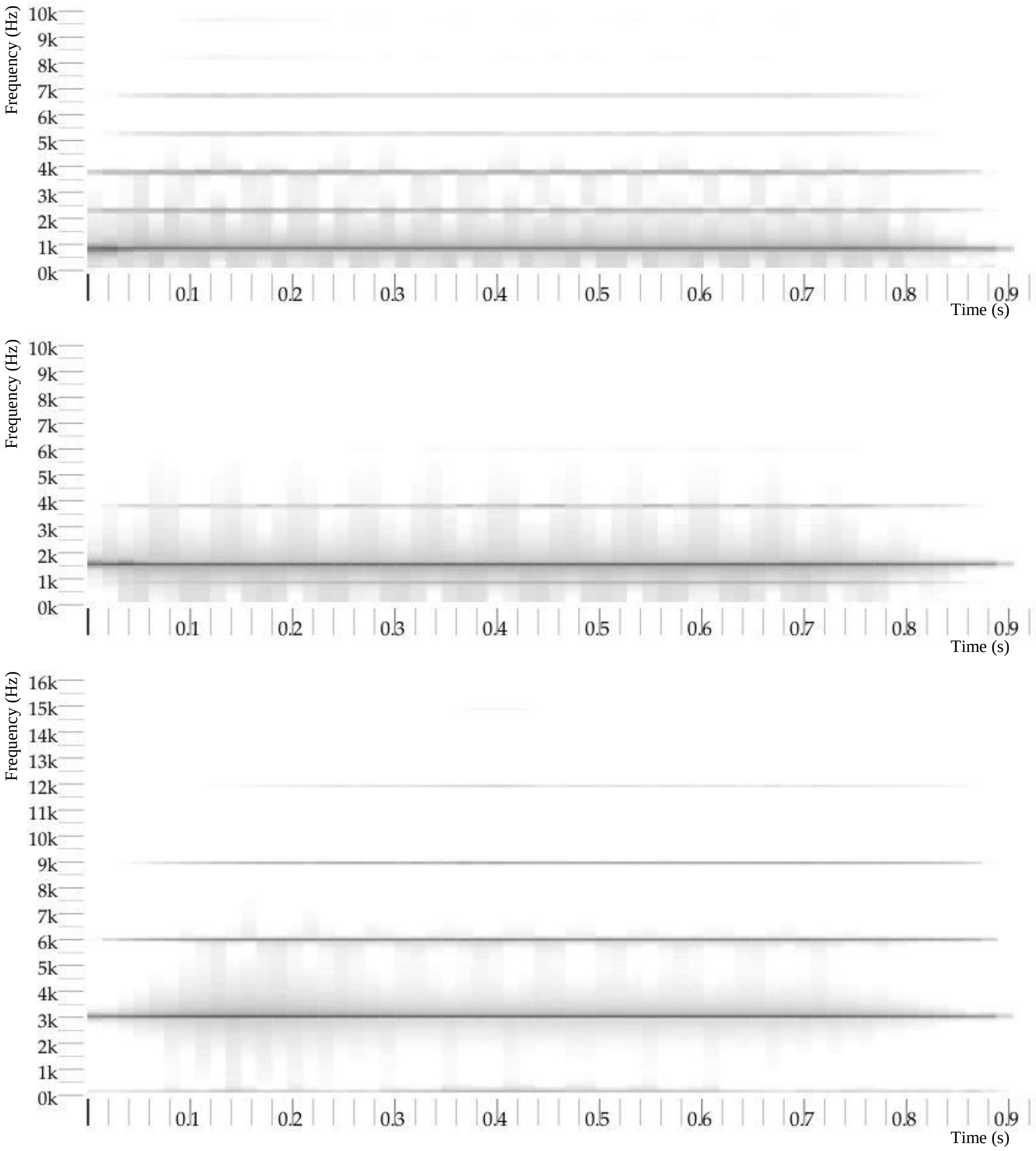


Figure 7.28: FM oboe simulation, isolated frequency spectrograms produced by simple FM element one (top), element two (middle) and element three (bottom)

Each element plays an essential role in the matching process, filling gaps in the frequency domain left by the other elements. As was noted in the experimentation with static sounds, fitness is computed from the spectral difference between target tones and their match candidates, partials with the highest magnitude are therefore prioritised by the matching algorithm. The composite long-term average frequency spectrum for the entire triple FM synthesis model is provided in figure 7.30 above the equivalent plot for the original oboe target sound.

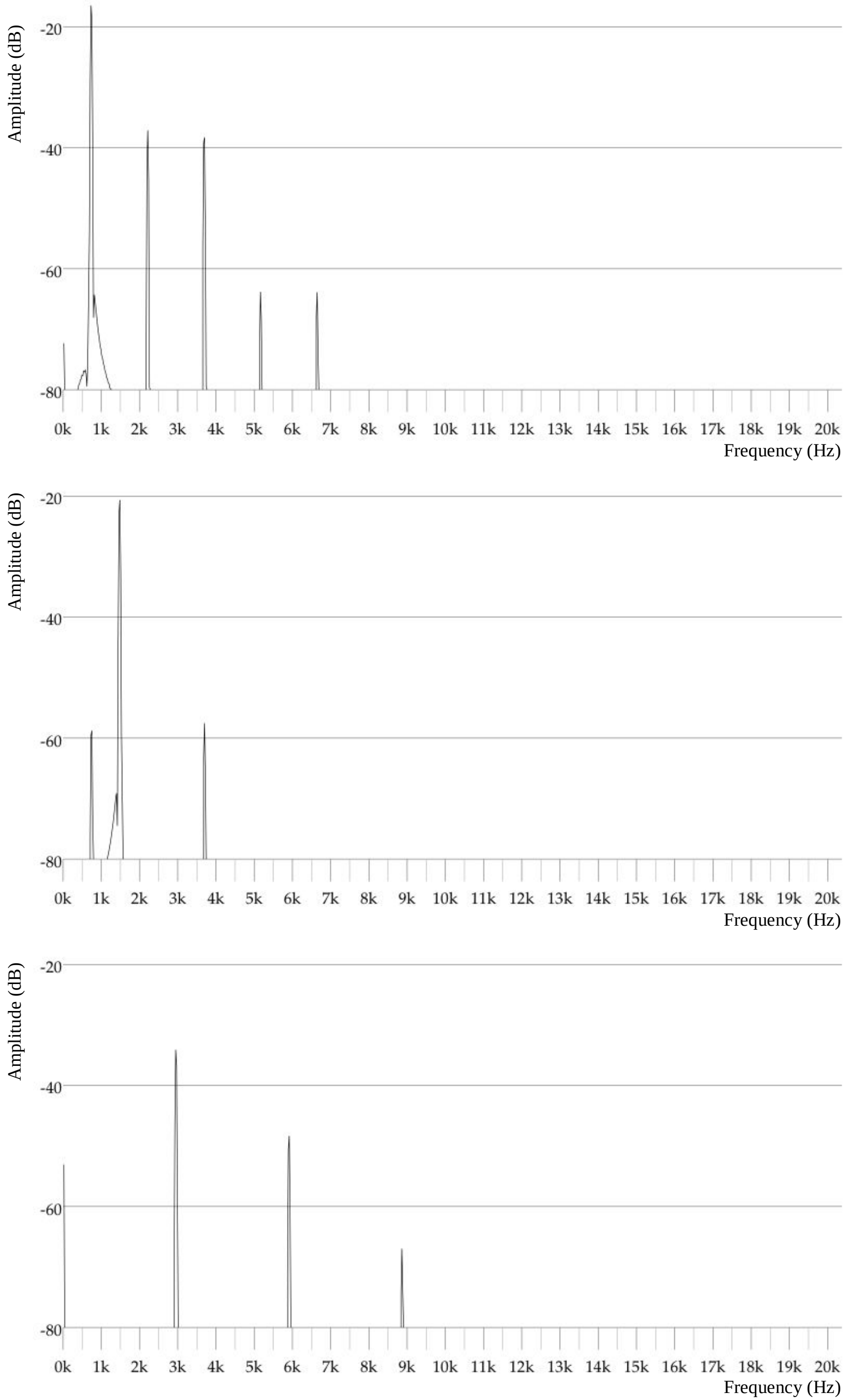


Figure 7.29: FM oboe simulation, isolated long-term average frequency spectrum produced by simple FM element one (top), element two (middle) and element three (bottom)

Many of the low frequency high amplitude harmonics are present and quite well matched in amplitude. However, the sixth harmonic is missing. With the decomposition of the matching synthesiser, it is possible to see why the sixth harmonic is unmatched. The carrier and modulator frequencies of the three elements have evolved such that it is not possible to position a partial at approximately 4.4kHz. Each simple FM element is already matching between one and three harmonics of greater amplitude, and therefore higher priority, than the fifth harmonic. This also goes some way to explain why the more complex elements in the onset of the target sound are overlooked in the match.

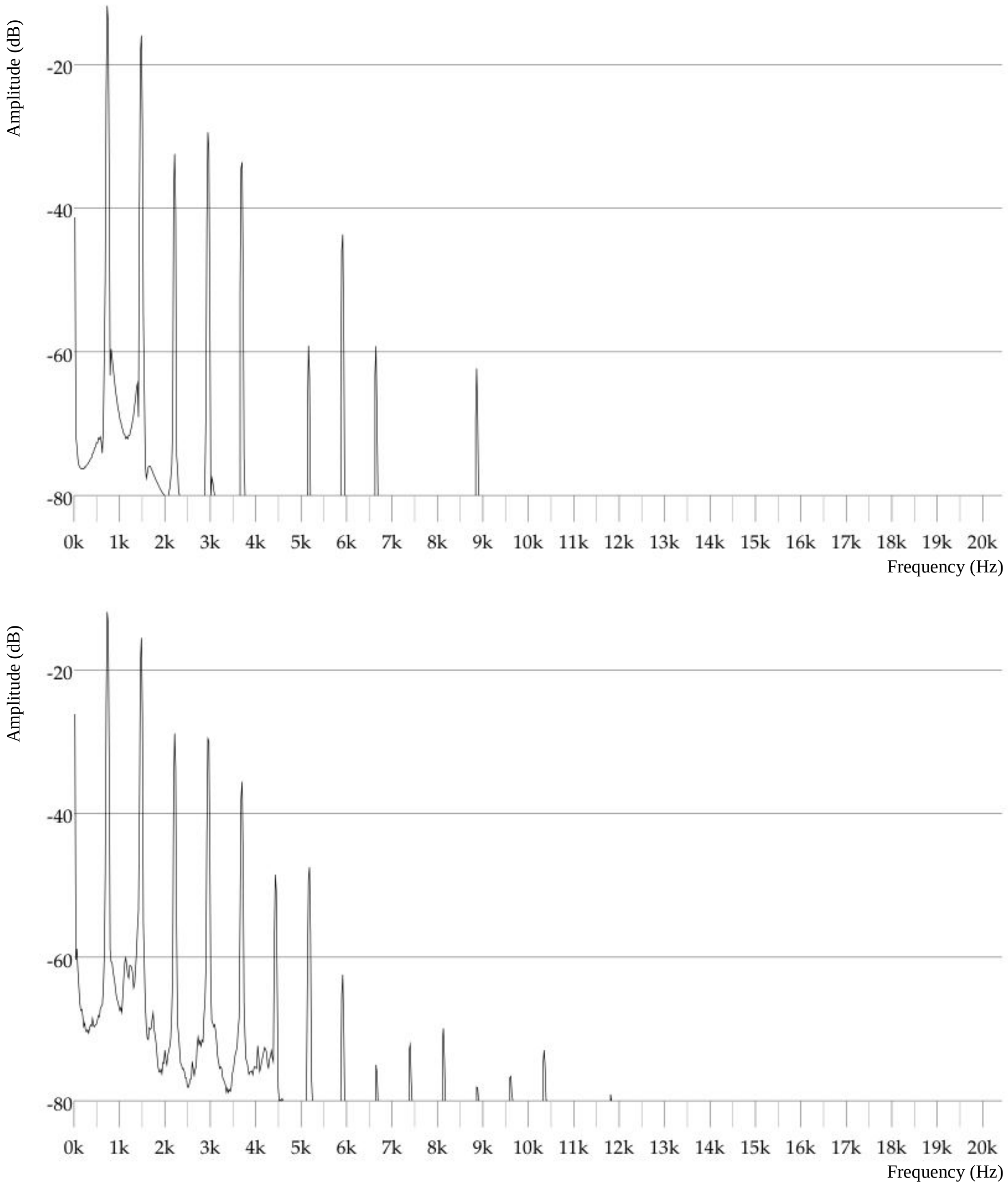


Figure 7.30: Long-term average frequency spectrum for triple simple FM oboe simulation (top) and original oboe sound (bottom)

Even the largest of the three FM synthesis models tested here is only capable of generating three overlapping bands of frequency partials. Matching the high frequency partials in the oboe target sound would require a more capable synthesis model.

7.8 Summary of this Chapter

In this chapter, details of the evolutionary FM matching system were introduced along with a contrived testing method that enables the capabilities of different optimisation algorithms to be compared in application to the parameter estimation problem. Experimentation was then presented in which six ES-based algorithms were applied to optimise the parameters of three FM synthesis models. Two experimental sections were included which explored two types of target sound with differing temporal characteristics: time invariant static tones and time variant dynamic sounds.

Experimentation began with the static target tones, matching initially contrived tones produced by the matching synthesiser, and then non-contrived tones produced by additive-subtractive synthesis and real acoustic instruments. Results from the contrived experimentation identified that the speciation EAs were the most consistent at retrieving contrived target sounds, when compared to the less sophisticated EAs. The CES and CCCES produce consistently more accurate matches for the three tested FM models. With the double and triple parallel FM models the CCCES was able to outperform all other algorithms in approximately 35% fewer fitness evaluations. The CES and CCCES systems were then used to match non-contrived additive-subtractive and acoustic instrument tones. The results of which confirmed that the CCCES is able to outperform the CES in fewer fitness evaluations in this domain. In the experiments with non-contrived target tones, it was demonstrated that the average relative spectrum error is reduced by increasing the complexity of the matching synthesiser, even when the algorithmic settings remain constant.

Performance of the EA-based systems was subsequently assessed when matching time-varying sounds, again matching initially contrived targets and then acoustic target sounds. The contrived matching experimentation highlighted some differing trends to the similar experiments with static tones. For example, the single solution produced by the CCES2 and ES was often found to be comparable with the solutions produced by the CCCES, which was again found to be the most accurate, robust and reliable algorithm. The improved performance of the less sophisticated algorithms suggested that the introduction of time-variation to the target sound improves the tractability of the search space. Matching was then performed with time-varying acoustic instrument tones, using the CCES2, CCCES and CES. Interestingly CCES2 was found to perform favourably with the CCCES and occasionally produced the smallest error, although the matches produced by the tested algorithms were not statistically different. Errors were again shown to be reduced when matching was performed by the larger synthesis models.

In the last section, match deficiencies were explored by decomposing an oboe simulation produced by the CCCES into the contributions of each synthesis element. It was shown that the amplitude and frequency characteristics of the acoustic instrument sound exhibit more complex variations than the FM synthesis model is capable of reproducing.

Chapter 8

Listening Tests

Despite the graphical analysis of the sound matches performed in chapter seven, neither the time and frequency plots, nor the computed relative spectrum error can confirm that the matches achieved are perceptually accurate. The ultimate judges of perceptual quality are, of course, human listeners. This chapter documents two perceptual listening tests, to support the empirical work performed earlier in this thesis.

8.1 Introduction

It is well understood that sound timbre can be related to the frequency spectrum of musical sounds (Roads, 1996). For this reason the spectrum error has been employed as an indicator of sound match accuracy in a range of studies; see for example Wehn (1998), Garcia (2000), Horner *et al* (1993a and 1993b), Horner (1998), Riionheimo and Välimäki (2003), McDermot *et al* (2005) and Mitchell and Pipe (2006). However, it has been noted that spectral difference is not always an exact indicator of perceptual difference (Yang, 2001), (Wun and Horner, 2001). In chapter seven, it was shown in the oboe example that certain partials may be neglected in the matching process in favour of those partials that contribute more to the overall spectrum match. However, it may be the case that some harmonics are perceptually more critical than others, and should not be considered with equal priority. An extensive examination of these perceptual cues is well beyond the scope of this thesis, however, it is important to confirm that the FM experimentation presented here has perceptual validity, and to establish the limits of the techniques that have been developed.

Test one is designed to measure the correlation between the relative spectrum error metric, and perceptual discrimination of human listeners on a set of FM synthesis tones. Test two is designed to obtain feedback on the effectiveness of the FM synthesis matching technique in application to acoustic instrument sounds.

8.2 Listening Panel and Test Conditions

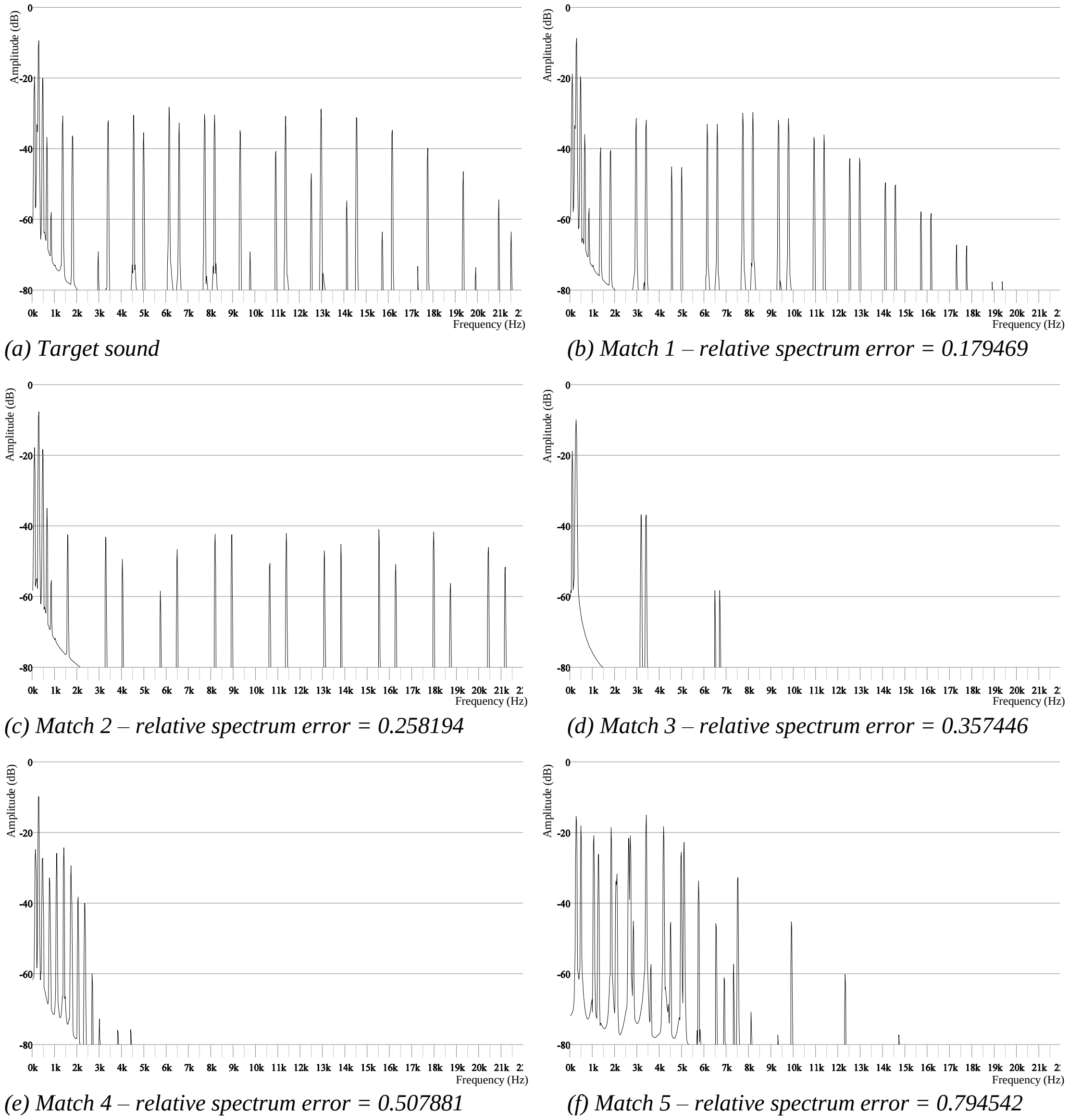
The listening panel for both experiments consisted of six subjects all of whom were familiar with a wide range of synthesis techniques, including FM synthesis. A brief background is provided for each participant below:

- ***Subject one*** is an experienced keyboardist who has studied music theory and composed/produced a variety of electronic and acoustic musical styles.
- ***Subject two*** is an experienced guitarist, keyboardist and band member who has released multiple electronic compositions.
- ***Subject three*** is a senior lecturer in music systems, plays a variety of instruments, and is an experienced researcher in the analysis of sound timbre.
- ***Subject four*** is an experienced keyboardist and band member, who has released multiple electronic compositions.
- ***Subject five*** is a senior lecturer in information systems as well as an experienced guitarist, singer and song writer.
- ***Subject six*** is a senior lecturer in music systems, with over 25 years experience in music production and recording.

Tests were performed in a quiet environment using Beyerdynamic DT100 headphones connected to the headphone output of a MacBook Pro computer. Sound samples were stored on hard disk in monaural 44.1kHz 16-bit integer format, and played back by listeners using a dedicated interface built using Max/MSP. Each test lasted for approximately 10-20 minutes. Participants were provided with a set of written instructions prior to each test and were provided with an opportunity to ask questions.

8.3 Listening Test One – Similarity Ranking

Listening test one was a tone similarity survey in which subjects were required to rank a set of static FM tones by their perceived similarity to a given target tone. Target tones were created by synthesising five sets of randomly generated parameters for the static tone double parallel simple FM model, depicted in figure 7.1b. As multiple matches were required of varying degrees of relative spectral error, each target tone was matched by the CES, and five matches were chosen from the final set of clustered solutions. Matches were chosen to be at least 0.05 from one another (a full breakdown of the relative errors for each match and the test results are provided in appendix one). An example target spectrum with five selected matches are plotted in figure 8.1, in order of increasing relative spectrum error. Subjects were required to rank the matched tones by their perceived similarity to the target tone.



(a) Target sound

(b) Match 1 – relative spectrum error = 0.179469

(c) Match 2 – relative spectrum error = 0.258194

(d) Match 3 – relative spectrum error = 0.357446

(e) Match 4 – relative spectrum error = 0.507881

(f) Match 5 – relative spectrum error = 0.794542

Figure 8.1: Example target sound and five matches with increasing relative spectral error

8.3.1 Test Interface and Instructions

Playback and rank order was controlled from a simple Max/MSP patch designed especially for the task. Tones could be played in any order and for any duration required by the subject without time restraints. The degree to which each match resembled the target was specified by clicking a button numbered one to five from a mutually exclusive vertically aligned *radio group*. The interface was configured such that none of the sounds could be ranked at the same level of similarity, forcing subjects to differentiate between tones. The interface for listening test one is depicted in figure 8.2.

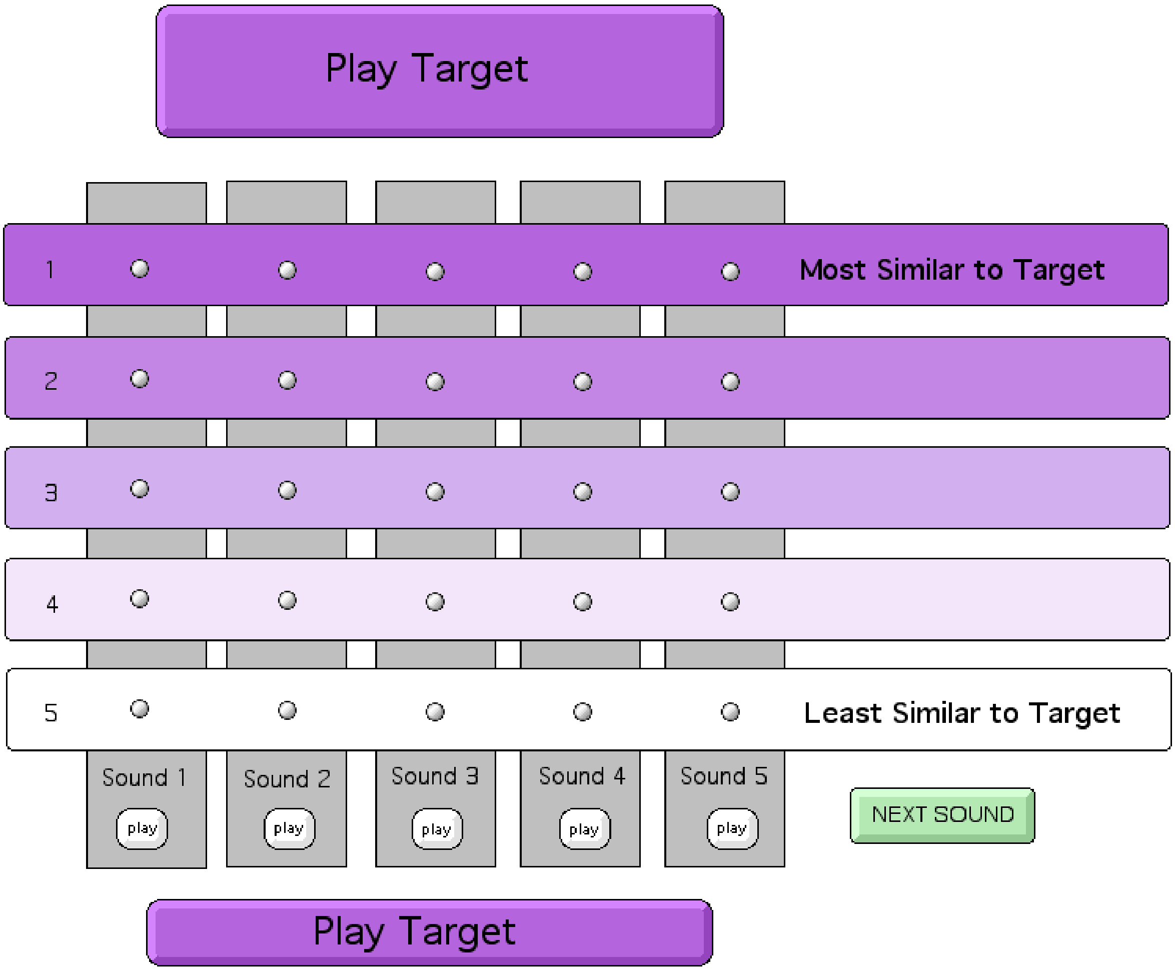
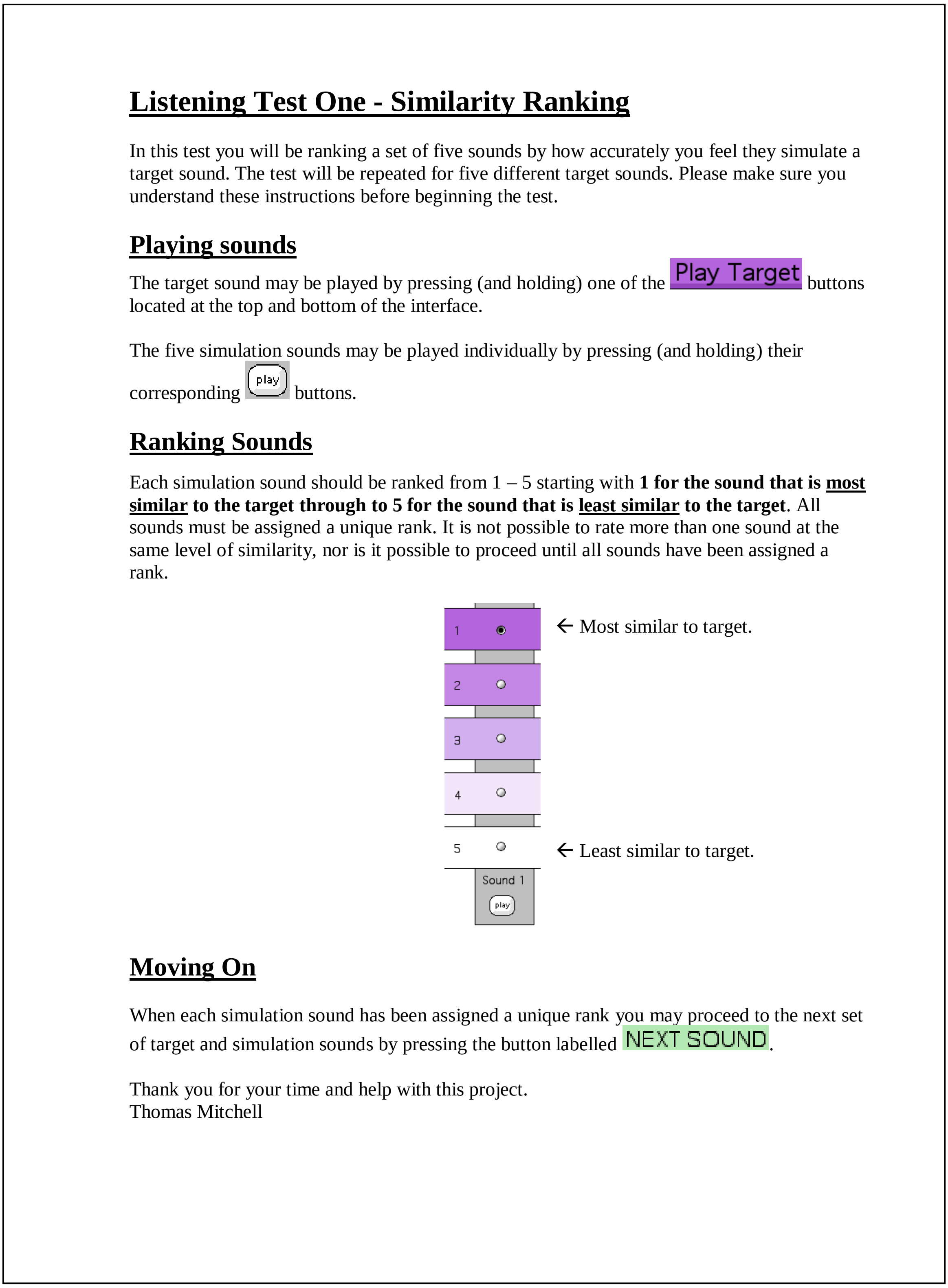


Figure 8.2: Listening test one interface

Matches one to five were ordered randomly with respect to their error, and in a different order for each of the five target tones. Each subject was provided with a set of instructions, given the opportunity to ask questions, and was not disturbed until the test was complete. At the end of the test, results were automatically written to a text file for subsequent analysis. The instruction sheet for test one is provided in figure 8.3.



Listening Test One - Similarity Ranking

In this test you will be ranking a set of five sounds by how accurately you feel they simulate a target sound. The test will be repeated for five different target sounds. Please make sure you understand these instructions before beginning the test.

Playing sounds

The target sound may be played by pressing (and holding) one of the Play Target buttons located at the top and bottom of the interface.

The five simulation sounds may be played individually by pressing (and holding) their corresponding play buttons.

Ranking Sounds

Each simulation sound should be ranked from 1 – 5 starting with **1 for the sound that is <u>most similar</u> to the target through to 5 for the sound that is <u>least similar</u> to the target**. All sounds must be assigned a unique rank. It is not possible to rate more than one sound at the same level of similarity, nor is it possible to proceed until all sounds have been assigned a rank.

1 ● ← Most similar to target.
2 ○
3 ○
4 ○
5 ○ ← Least similar to target.
Sound 1
play

Moving On

When each simulation sound has been assigned a unique rank you may proceed to the next set of target and simulation sounds by pressing the button labelled NEXT SOUND.

Thank you for your time and help with this project.
Thomas Mitchell

Figure 8.3: Listening test one instruction sheet

8.3.2 Results and Discussion

Each target tone was accompanied by five tones of varying match accuracy, which were ranked in order of target sound similarity by the relative spectrum error metric and each test subject. The results shown in table 8.1 provide a breakdown of the correlation (Spearman's rho) between metric ranking, and test subject ranking where ** indicates a result that is significant at the 0.01 (two-tailed) and * indicates a result that is significant at the 0.05 level. The ranks provided by subjects four and six correlate exactly with the metric for target tones one and two. The last column shows the internal consistency reliability of all participants (Cronbach's alpha).

	Subject 1	*Subject 2*	*Subject 3*	*Subject 4*	*Subject 5*	*Subject 6*	*Reliability*
Target Tone 1	0.9*	0.9*	0.8	1.0**	0.9*	1.0**	0.979
Target Tone 2	0.7	1.0**	1.0**	1.0**	1.0**	1.0**	0.982
Target Tone 3	0.9*	0.9*	0.9*	0.9*	0.9*	0.9*	1.0
Target Tone 4	1.0**	0.9*	0.9*	0.9*	1.0**	0.9*	0.983
Target Tone 5	1.0**	1.0**	1.0**	0.9*	1.0**	1.0**	0.994

Table 8.1: Listening test one correlation and reliability results.

These results show a strong positive linear association between the perceived and analytical ranking, supporting the use of the relative spectrum error metric as an indicator of perceived similarity. Matched tones were ranked in exactly the same order by both the test subjects and the analytical measure almost 50% of the time. However, there is clearly some variation both in correlation with the metric and internally between participants (appendix one), suggesting that the analytical measure differs somewhat from the perceptual criteria of the test subjects. The variations between participants suggest that perceived differences between the matched tones are prioritised differently by different subjects.

The results for tone three were ranked similarly by subjects yet different from the metric. In this particular tone, matches ranked second and third by the metric were consistently ranked third and second by the test subjects. The spectrum plots for theses two sounds and the target are provided in figure 8.4.

The target illustrates a strong band of midrange partials between 2 and 3kHz with evenly distributed harmonics at intervals of approximately 2.45kHz. Match one also exhibits a large band of midrange partials, however only very few of the high frequency partials coincide with those of the target. Match two on the other hand, matches the higher frequency harmonics well but the midrange partials are only partially represented. In terms of relative spectrum error, match one is ranked highest as a large proportion of the spectral energy is contained within the midrange band. However, to the test subjects, this midrange correlation was not sufficient to distinguish match one as the superior simulation. Any differences between the midrange partials combined with the high frequency differences clearly tips the perceptual scales in favour of match two. It is clear from these experiments that the process of auditory perceptual discrimination is more complex than the unbiased view of relative spectral error. However despite these minor discrepancies and the small sample size, the results provide overwhelming evidence that the relative spectrum error acts as a good indicator of perceptual similarity.

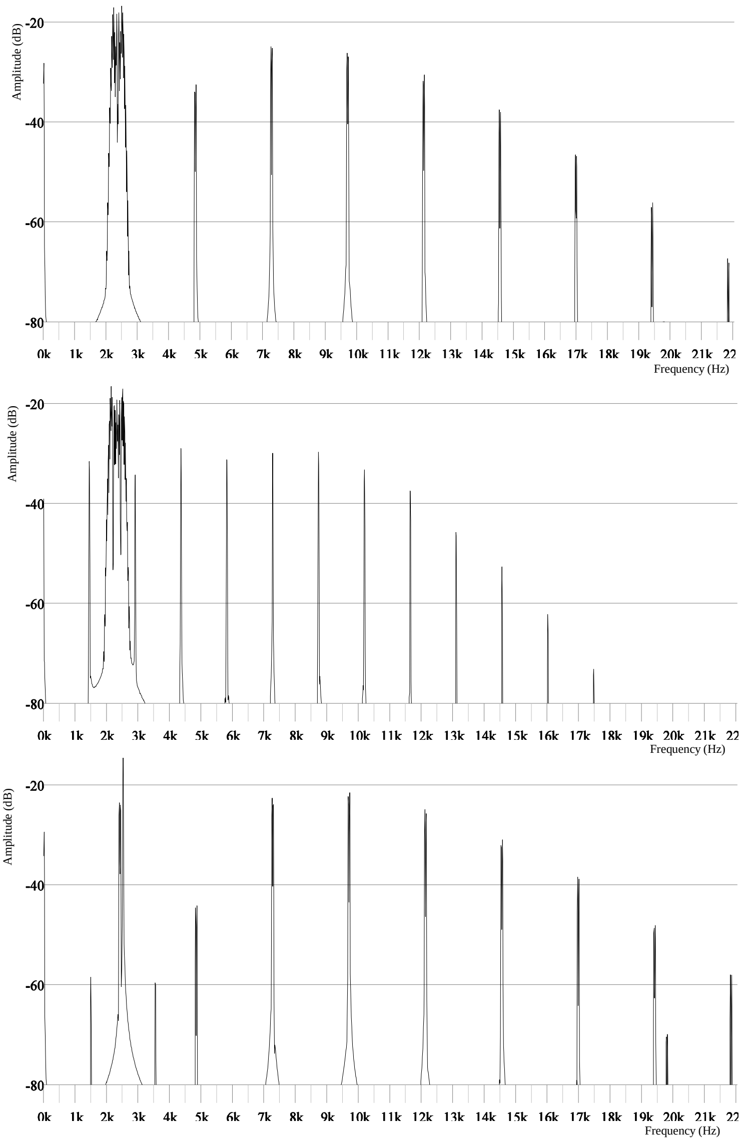


Figure 8.4: Tone three target (top), match one(middle) analytically ranked second perceptually ranked third, match two (bottom) analytically ranked third perceptually ranked second.

8.4 Listening Test Two – General Sound Simulations

Listening test two was a discussion-based qualitative analysis of the subjects' responses to a set of time-varying sound matches performed by the FM matching systems developed in this thesis. A set of four acoustic instrument sounds were matched using the time-varying triple parallel simple FM model (figure 7.1c), with parameters optimised by the CCCES. Algorithmic parameters were identical to those employed in section 7.7.2 to produce the results shown in table 7.16. All target sounds were produced by Opolko and Wapnick (1989), and included recordings of the piano, trumpet, violin and cymbal. All subjects have prior experience of FM synthesis, and were aware of its limitations.

8.4.1 Test Interface and Instructions

Playback of each target sound and match was controlled by a simple Max/MSP patch designed specifically for the task. Subjects were able to playback the sounds at any point during the discussion. The interface for listening test two is shown in figure 8.5.

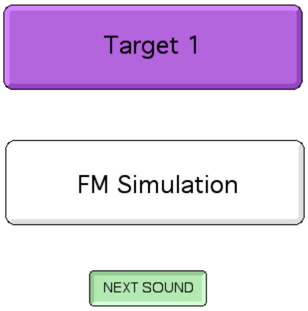
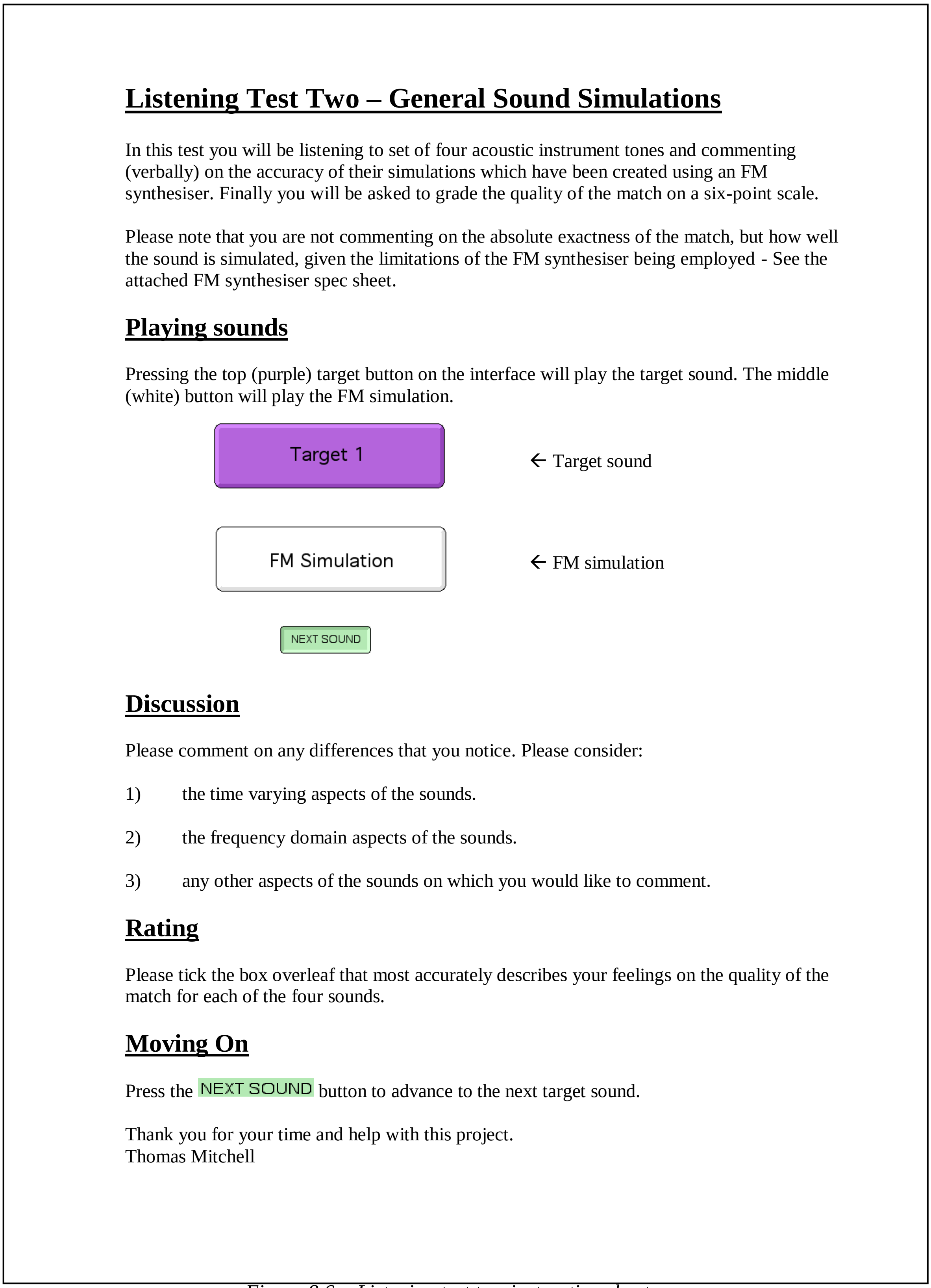


Figure 8.5: Listening test two interface

Subjects were provided with a set of instructions and given an opportunity to ask questions prior to the test. Subsequent to the discussion of each match, subjects were instructed to provide a general 'feel' mark for the quality of the match, with an awareness of the underlying synthesis model, according to their own subjective criteria on a six-pioint semantic differential scale with the opposing statements *Good* and *Bad* (Brace, 2004). The discussion was recorded using an Edirol R-09 audio recorder. Instruction sheets for test two are provided in figures 8.6a, b and c.



Listening Test Two – General Sound Simulations

In this test you will be listening to set of four acoustic instrument tones and commenting (verbally) on the accuracy of their simulations which have been created using an FM synthesiser. Finally you will be asked to grade the quality of the match on a six-point scale.

Please note that you are not commenting on the absolute exactness of the match, but how well the sound is simulated, given the limitations of the FM synthesiser being employed - See the attached FM synthesiser spec sheet.

Playing sounds

Pressing the top (purple) target button on the interface will play the target sound. The middle (white) button will play the FM simulation.

← Target sound

← FM simulation

Discussion

Please comment on any differences that you notice. Please consider:

1) the time varying aspects of the sounds.

2) the frequency domain aspects of the sounds.

3) any other aspects of the sounds on which you would like to comment.

Rating

Please tick the box overleaf that most accurately describes your feelings on the quality of the match for each of the four sounds.

Moving On

Press the NEXT SOUND button to advance to the next target sound.

Thank you for your time and help with this project.
Thomas Mitchell

Figure 8.6a: Listening test two instruction sheet one

FM Synthesiser Specification

The specification of the FM synthesiser, which has been used to simulate the target sounds, is as follows.

Triple Simple FM Model

The synthesis model is constructed from three simple-FM arrangements which are connected in parallel, as illustrated in figure 1.

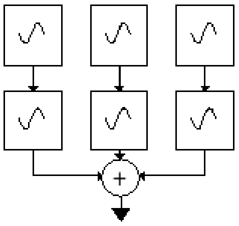


Figure 1: Parallel triple FM model

Simple FM Synthesis

Each of the three simple FM arrangements is composed from a two sinusoidal oscillators, which are connected such that the instantaneous amplitude of the one oscillator (modulator) varies the frequency of the other oscillator (carrier). This configuration is shown in figure 2.

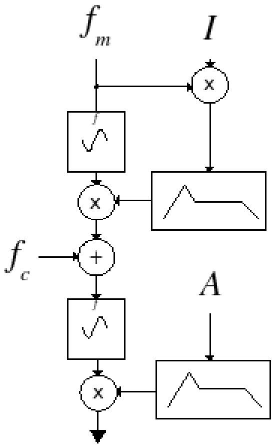
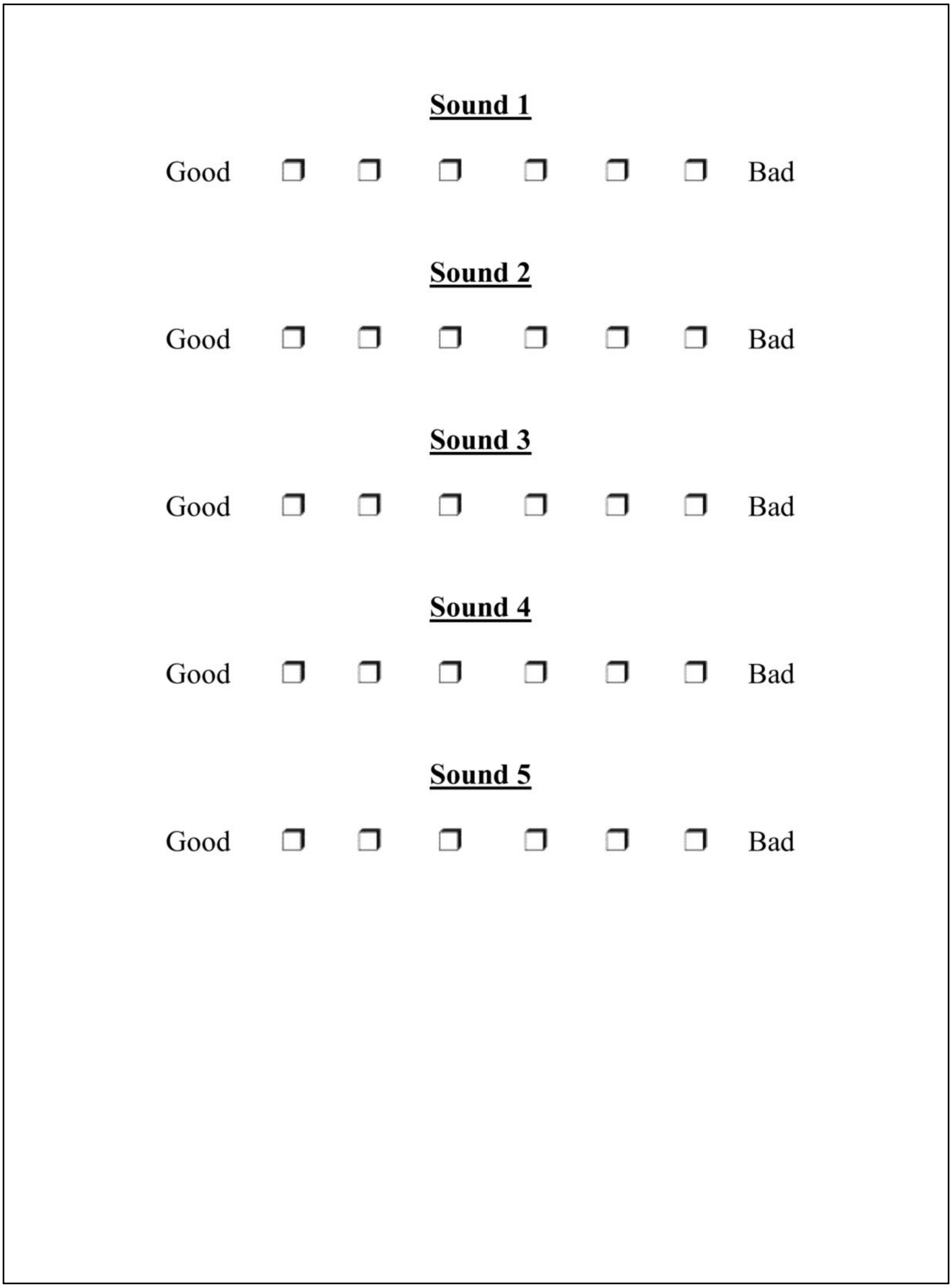


Figure 2: Simple FM block diagram

A simple ADSR envelope generator varies both the carrier amplitude and modulation index. If you have any further questions regarding this synthesis model please ask the interviewer.

Figure 8.6b: Listening test two instruction sheet two



Sound 1

Good □ □ □ □ □ □ Bad

Sound 2

Good □ □ □ □ □ □ Bad

Sound 3

Good □ □ □ □ □ □ Bad

Sound 4

Good □ □ □ □ □ □ Bad

Sound 5

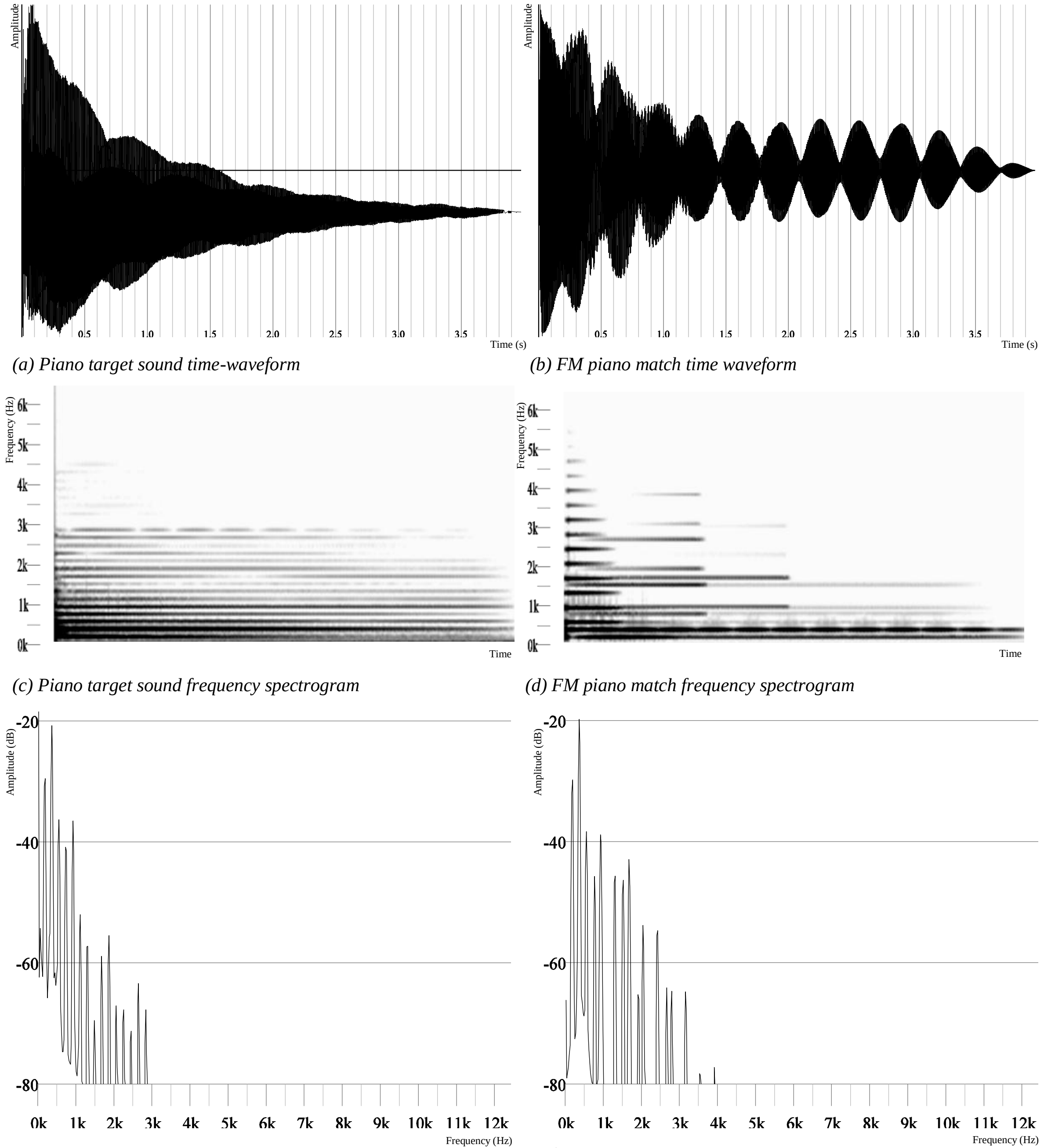
Good □ □ □ □ □ □ Bad

Figure 8.6c: Listening test two instruction sheet three

8.4.2 Results and Discussion

8.4.2.1 Piano

The time domain waveforms, frequency spectrograms, and long-term average spectrum of the target piano sound at 185Hz (F#3) and the evolved triple simple FM match are provided in figure 8.7. The match was evolved by the CCES and achieved a relative spectrum error of 0.26547.



(a) Piano target sound time-waveform

(b) FM piano match time waveform

(c) Piano target sound frequency spectrogram

(d) FM piano match frequency spectrogram

(e) Piano target sound long-term average spectrum

(f) FM piano match long-term average spectrum

Figure 8.7: Piano target and matched sound time and frequency plots

The FM sound produced by the matching system in this experiment was noted by several participants to contain characteristics of the target piano sound. Subjects one, two, three and five, for example, noted that the certain aspects of pitch, amplitude envelope, timbre and character, were reproduced. These observations can be confirmed in figure 8.7. The overall amplitude envelope of the target and matched sounds show clear similarities, with maximum amplitude occurring at the onset of the sound. Each of the three simple FM elements within the matching synthesiser is unable to produce non-linear amplitude transitions independently; however, a reasonable approximation is achieved in the composite output. The distribution and amplitude of the lower harmonics are also well represented in the match. However some partials are missing and others are over-emphasised in the match. Subject three also identified a subtle modulation in the target sound due to the stiffness of the piano strings, and was impressed that the matching system had reproduced this variation. However subject three did add that this variation was 'over the top' and 'too fast in the FM sound.' This observation is confirmed in the time waveforms as a somewhat periodic fluctuation in the amplitude envelope, which is faster and larger in magnitude in the matched sound.

All subjects could identify clear differences between the sounds, stating that the FM sound could be easily distinguished as a synthesiser. The match was betrayed by two significant time-varying features. Subjects one, three, four, five and six noted that the attack segment of the matched sound contained a 'frequency sweep' (subject one) that was not present in the target. The beginning of the match was identified to be 'brassy' by subjects three, four and six, where higher frequency partials swept into the sound over a longer duration in the target. The second major difference was in the movement of the higher frequency harmonics. These artefacts are visible in the spectrogram plots as two distinct groups of harmonics that end abruptly approximately one and two seconds into the match. These harmonics were described by test subjects as 'a sound playing backwards' (subject three), 'sweeping tones' (subject one), 'FM noise' (subject two), 'extra frequencies' (subject five) and 'rogue frequencies' (subject six). Subject four identified that these harmonics were in the target sound, but not as important as other components that were missing.

It is suggested that these 'rogue harmonics' are caused by the under-sampling of the target sound in the time domain. Certain harmonics gradually rise and fall in the target sound, which can be observed in the spectrogram at the 15th harmonic. As only 10 uniformly spaced frames are matched, the system has no way of knowing how partials transit between frames, and consequently provides its own transition. In this case the high frequency harmonics recede unnaturally compared with the target sound.

Results from the semantic differential ratings are provided in table 8.2 where a rating of one is classed as *good* and a rating of six, *bad*.

Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Average
2	3	3	3	3	2	2.6

Table 8.2: Piano semantic differential results

8.4.2.2 Trumpet

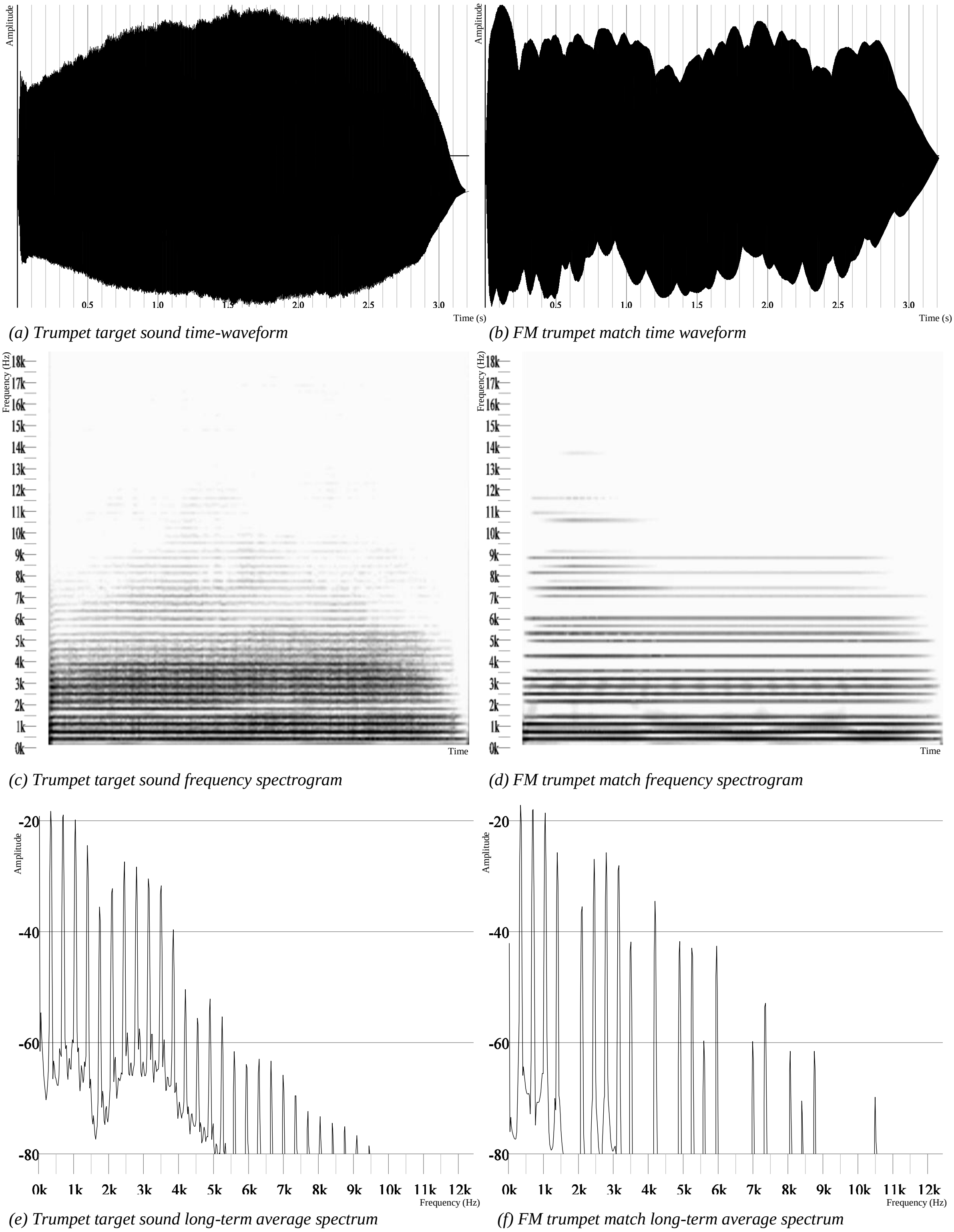
The time-domain waveforms, frequency spectrograms, and long-term average spectrum of the target trumpet sound at 349.23Hz (F4) and the evolved triple simple FM match are provided in figure 8.8. The match was evolved using the CCCES and achieved a relative spectrum error of 0.105399.

The FM sound produced by the matching system elicited a positive reaction from all participants. The match was described as 'really similar' (Subject one), 'quite good' (subjects two and five) and 'excellent' (subject six). However, subject three remarked that, given the design of the matching synthesiser, the match was 'not as good as it possibly could be'. All participants noted that the overall pitch, timbre and amplitude envelope was reproduced in the target and many of the time varying attributes were also matched. Subject three commented on the spectral development of the match as follows:

> *"The character in the sustain portion does actually develop in a slightly similar way to the original. If you listen to the original brass sound, there is actually a slight modulation in character as it progresses and you can hear that in the FM one."*

Subject three also expressed how the match exhibits certain characteristics of a wind instrument at the onset of the sound. Adding that the elements of breath, identifying the target as a brass sound, are not represented in the match.

All participants remarked upon a harmonic imbalance in the higher frequencies in the match that differentiated it from the target. For example, subject two described the match as 'slightly too bright,' with 'an extra dominant pitch'. Subject three described a 'strong octave' in the match as though the 'second harmonic is standing up too much'. Subject four described an upper harmonic ‘that makes it sound metallic in the simulation’. These observations can be related to the spectral analysis in figure 8.8. The long-term average spectrum, for example, shows that the partials above 4kHz have much larger amplitudes in the match than in the target. Furthermore, many of the high frequency partials are missing in the matched sound.



(a) Trumpet target sound time-waveform

(b) FM trumpet match time waveform

(c) Trumpet target sound frequency spectrogram

(d) FM trumpet match frequency spectrogram

(e) Trumpet target sound long-term average spectrum

(f) FM trumpet match long-term average spectrum

Figure 8.8: Trumpet target and matched sound time and frequency plots

Results from the semantic differential ratings are shown in table 8.3.

Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Average
1	2	2	2	2	2	1.833

Table 8.3: Trumpet semantic differential results

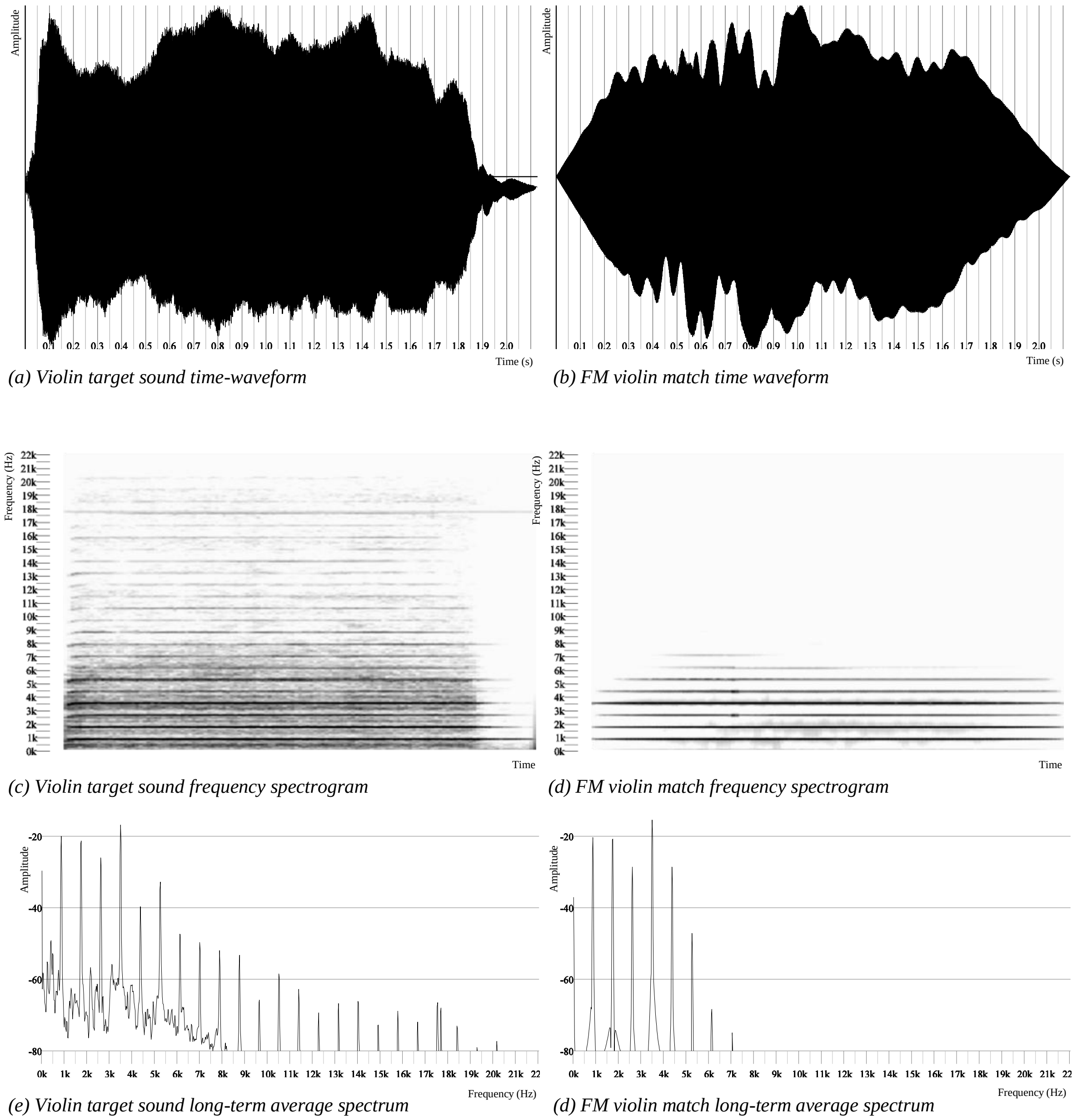
Given the quantity of harmonics, and the shape of the spectral envelope, and the relative simplicity of the triple FM synthesis model, many of the harmonics are well represented in the match.

8.4.2.3 Violin

The time-domain waveforms, frequency spectrograms, and long-term average spectrum of the target violin sound at 880Hz (A5) and the evolved triple simple FM match are provided in figure 8.9. The match was evolved using the CCES and achieved a relative spectrum error of 0.163790.

Several participants commented that the FM sound produced by the matching system accurately simulated certain characteristics of the target violin sound. For example, subjects one, two, three, four and five remarked on how aspects of the pitch, amplitude envelope and timbre of the synthesised sound matched the original violin sound. This correlation is evidenced in the long-term average spectrum provided in figure 8.9. The harmonics from the fundamental to the eighth harmonic are synthesised in the match, with the fourth harmonic emphasised in both sounds.

Subject four perceived a longer attack time in the simulation than in the target which can be confirmed in the time-waveform plot of figure 8.9. Subjects one and three observed ‘small pitch fluctuations’ and a ‘subtle twisting in the character’ of the target sound, which was also present in the match. Furthermore, subject three also commented that the system ‘would have difficulty doing any better than that, given the complexity of the model’. However, all participants noticed an absence of noise and high frequency content in the match, which can be observed in figure 8.9c. Remarks included, ‘the simulation seems duller... it’s missing the scraping of the bow’ (subject one), ‘the simulation isn’t quite harsh enough’ (subject two), ‘there are slightly less high frequencies in the simulation than in the target... the target has got a scrape, that isn’t in the simulation’ (subject three), ‘there should be a bit more high frequency content’ (subject four) and ‘no bowing sound really on the simulation but that would be quite hard to get I would think’ (subject five).



(a) Violin target sound time-waveform

(b) FM violin match time waveform

(c) Violin target sound frequency spectrogram

(d) FM violin match frequency spectrogram

(e) Violin target sound long-term average spectrum

(d) FM violin match long-term average spectrum

Figure 8.9: Violin target and matched sound time and frequency plots

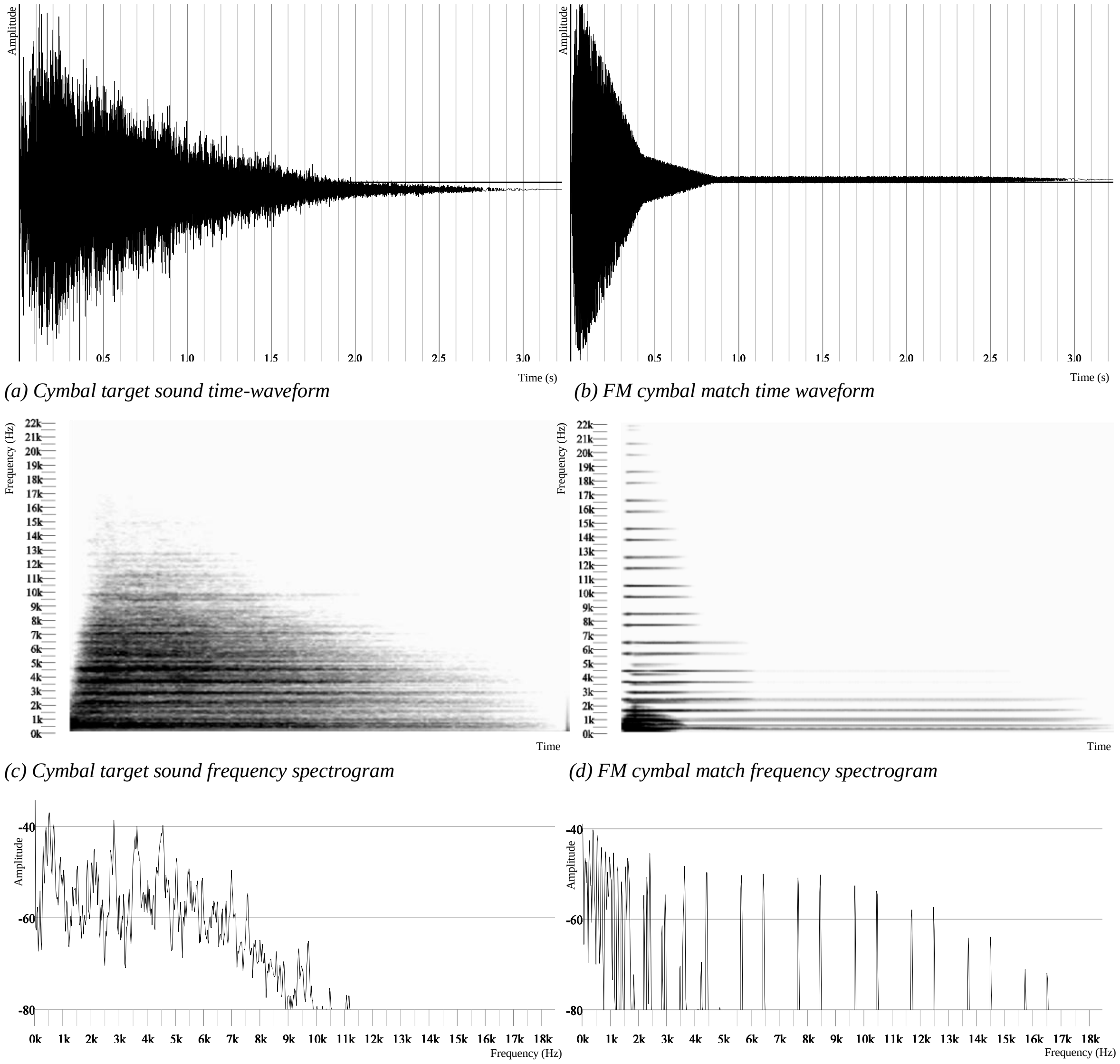
The matching system had placed focus on the low-frequency high amplitude components of the frequency spectrum and ignored the noise produced by the bow entirely. Some participants agreed that the absence of the bowing noise made the target sound ‘better’ or ‘purer’ than the target (subjects two and three), while others found the missing characteristics to be an essential component of the target violin (subjects one, four and six). Consequently, the values on the semantic differential scale varied significantly due to the polarised viewpoints of the test subjects.

Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Average
3	1	1	5	1	4	2.5

Table 8.4: Violin semantic differential results

8.4.2.4 Cymbal

The time-domain waveforms, frequency spectrograms, and long-term average spectrum of the target cymbal sound and the evolved triple simple FM match are provided in figure 8.10. The match was evolved using the CES and achieved a relative spectrum error of 0.20456.



(a) Cymbal target sound time-waveform

(b) FM cymbal match time waveform

(c) Cymbal target sound frequency spectrogram

(d) FM cymbal match frequency spectrogram

(e) Cymbal target sound long-term average spectrum

(d) FM cymbal match long-term average spectrum

Figure 8.10: Cymbal target and matched sound time and frequency plots

The cymbal tone produced a less positive response from the participants, with general comments including: ‘they are obviously not the same’ (subject two), ‘I don’t think there’s anything in that FM simulation that was like a cymbal’ (subject six) ‘[the match does not] sound particularly like the target’ (subject five). The plots in figure 8.10 indicate a significant amount of complexity in the target sound, with inharmonic partials distributed throughout the frequency spectrum. The complexity of the target sound far exceeds the capabilities of the triple simple FM matching synthesiser, a restriction noted by subjects three, four and six.

Many participants compared the target and match by their amplitude functions. In figure 8.10, the target sound follows an exponential decay with the amplitude decreasing faster at the beginning than at the end. The matched sound amplitude envelope is clearly shaped by three linear sections that approximately fit the shape of the target. Subjects two and five remarked that the decay time on the matched sound was too short, and subject six noted that the sustain period was too long.

Subject three described the spectral development of the target sound to be ‘noisy inharmonic’, whereas the FM sound was ‘constant inharmonic’. This effect was also observed by subject six, who commented that the simulation had discernable pitch that the target did not. The spectrogram plots shown in figure 8.10 indicate that target sound does exhibit inharmonic peaks, but the energy of the partials is distributed across all frequencies. However, the match clearly targets certain frequencies and not others. Given that the FM parameters only enable the variation of partial amplitudes at fixed frequencies, this characteristic is endemic to the synthesis model; broadband noise, as featured in the target sound, exceeds the capabilities of the matching synthesiser.

Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Subject 6	Average
5	5	5	2	5	6	4.667

Table 8.5: Cymbal semantic differential results

8.5 Chapter Summary and Conclusions

Two experiments have examined the relationships between the results produced by the matching system developed in previous chapters and the judgements of perceived similarity by expert listeners. In the first test, a group of static FM tones were ranked by the relative spectrum error metric and by six human listeners. The results indicated significant positive correlation between the subjective evaluations of the test participants and the automatic measure of error employed by the matching system. Some variation between the two suggested that the metric could not be assumed as a direct measure of perceptual similarity. The relative spectrum error measure presents an unbiased comparison in the frequency-domain. However, the human auditory system uses a more sophisticated form of analysis. A better matching system might be achieved by incorporating bias toward spectral features which are important to the human auditory system. This may include considerations for frequency masking and equal loudness curves (Zwicker and Fastl, 1999), as considered by Riionheimo and Välimäki (2003), the balance and significance of harmonics and inharmonics (So and Horner, 2004), critical bands, noise bands and spectral envelopes, for example. Due to an unbiased view of similarity, the system may currently be misled by partials of large magnitude but limited perceptual significance.

The second test gathered feedback on the performance of the system when matching acoustic target sounds. The test produced a generally positive reaction from participants and enabled system limitations to be recognised. Match inaccuracies can be attributed to different types of limitations in the system. For example, the inability of the system to match the exponential decay and noisy components of the cymbal sound can be attributed to the limitations of the matching synthesiser. It is asking a great deal of the simple parallel FM synthesiser to produce a high quality recreation of sounds such as the cymbal considered in 8.4.2.4. However, there were test cases in which subjects noted that a better match might have been possible with the matching synthesiser. Under these circumstances, it is clearly the capabilities of the matching technique that limits the quality of the match. For example, the piano match exhibited unnatural characteristics as a result of the limited temporal representation of the target sound. With target variation only captured by 10 uniformly spaced sample-points, transitions that occur between these points will be interpolated by the matching synthesiser and may not accurately match the development of the target sound. Furthermore, rapidly changing target sounds will not be sufficiently characterised to ensure an accurate match, which may also contribute to the poor results with the cymbal sound and the missing transients in the piano, violin and trumpet sounds.

However, there is an additional factor in the matching technique, and that is the ability of the underlying optimiser to effectively control the parameters of the matching synthesiser. It is this last factor on which a significant proportion of the work presented on this thesis has been focussed. These three factors correspond directly with the components of the synthesis matching problem identified in section 6.3.5: the synthesis model, the similarity metric and the characteristics of the target sound. Future development in each of these components will be central to future work in synthesis parameter estimation.

Chapter 9

Conclusions and Further Work

Within this thesis, the central focus has been on the application of evolutionary computation to assist in the process of sound matching with the highly complex and non-linear Frequency Modulation (FM) synthesis technique (Chowning, 1973). Consequently, work from both of these fields have been reviewed, explored and built upon.

9.1 Thesis Summary and Conclusions

In chapter one, the context, motivation, objectives and research question for this work were introduced in order to set the scene for the following chapters. A general background to the field of evolutionary computation was provided in chapter two, in which the principal EAs, intended for static objective function optimisation were introduced. Emphasis was placed on the ES, the framework within which the algorithms developed in later chapters were built. Issues relating the preconvergence issues of evolutionary algorithms in multimodal search spaces were reviewed in chapter three, along with the efforts which have been made to prevent suboptimal convergence. The concepts of species and niche were introduced and the factors that limit their concurrent maintenance in traditional EAs were summarised.

The architecture for a novel niching EA was introduced in chapter four, which extended the capabilities of the FCES algorithm in light of the work reviewed in chapter three. This algorithm was named the Clustering Evolution Strategy (CES), which was comparatively examined in application to a series of well known real-valued test functions. The first set of experiments in section 4.3.1 set out to determine the ability of the CES to locate the global optimum when applied to the Multimodal, Langerman and Maximum of Two Quadratics functions. Of all tested multimembered algorithms, the CES was shown to be the most robust to change in the search environment.

In the second set of experiments, the niching capabilities of the CES were examined, by recording the number of optima maintained in comparison with the closely related FCES developed by Sullivan (2001). Results were produced using well known performance measures, and the CES was again found to consistently locate significantly more optima than the Fuzzy Clustering alternative. However, although the FCES exhibited significantly improved performance when adopting the proposed *restricted cluster selection* of the CES, with identical selection operators, the CES still demonstrated superior performance. This performance improvement was attributed to differences in clustering and recombination mechanisms.

In section 4.3.3, the CES was assessed in application to high-dimensional search space problem domains. Performance was compared with a multi-start hill climber algorithm, which was composed of multiple $(1 + 1)$ ESs which would have matched the CES performance in the earlier experimentation. The CES was demonstrated to consistently outperform its competitor in the higher-dimensional multimodal problems, in which the number of search space peaks significantly outnumbered the search points sampled at each generation.

In chapter five, the principles of niching were applied to the CCEA architecture to enable the location and concurrent maintenance of multiple search space solutions at distinct optima. A general model for the Niching Cooperative Coevolutionary Algorithm (NCCEA) was introduced with a corresponding collaboration technique to enable species to form within each subpopulation. The intention was to preserve the benefits of parameter optimisation with CCES, while limiting the effects of *relative overgeneralisation* (Wiegand, 2004). Thereafter, an instance of the NCCEA model, named the Clustering Cooperative Coevolution Strategy (CCCES), adopting the CES as the underlying niching algorithm, was implemented and tested within several multimodal search space environments. It was shown that the new operators, which encourage the development of subpopulation species, were also found to reduce the effects of *relative overgeneralisation*, and increase the likelihood of ideal (optimal) collaboration. This result was demonstrated in application to the MTQ function, which is known to exacerbate this pathology (Wiegand, 2004). Interestingly, the CCCES with six subpopulation species was found to outperform all multimembered algorithms, including those tested throughout chapter four. This performance improvement was attributed to the partitioning of the coevolving subpopulations into separate species, and the subsequent diverse collaboration procedure that provides a more methodical technique for selecting multiple collaborators than the simple stochastic methods adopted in previous studies (Wiegand, 2001).

A review of FM synthesis and related work was provided in chapter six, forming a real-world test environment within which to examine the performance of the developed niching algorithms. The application of EC to the dynamic-sound FM synthesis model itself constitutes a major contribution, and the ultimate motivation for this work. A contrived sound matching method was introduced that enables the matching technique be isolated from the synthesiser limitations by factoring out sounds that the synthesiser is incapable of matching. This enables the effectiveness of each optimisation technique to be quantified. The experiments reported in chapter seven of this thesis concern the application of a variety of EAs to the problem of sound matching with FM synthesis. Both of the novel algorithms proposed in this thesis, CES and CCCES, were found to consistently deliver multiple potential high-quality solutions, as well as the closest matches to both contrived and non-contrived targets, with the CCCES shown to be the most robust for navigating the domain of static tones and time-varying sounds. The simple CCES was also found to yield strong results, but only when matching time-varying sounds.

It has been established that FM provides a difficult search domain, but, when the EA techniques are applied to a parallel implementation of the simple FM arrangement, it is possible to retrieve synthesis parameters that produce tones that match 75-95% of the spectra of the real acoustic instrument tones that were tested. For this purpose, the CCCES was found to be the most robust algorithm when matching both static tones and time-varying sounds.

The experimental work in chapter eight involved two listening tests with a panel of six subjects experienced with FM synthesis. The first test confirmed the significant positive relationship between spectral similarity (established by relative spectrum error), and perception. However, variation in the results confirmed that the relative spectrum error cannot be assumed as an exact measure of perceptual similarity. Human listeners use a more complex analysis technique when ranking sound similarity, which is not truly reflected by the relative spectrum error metric. The second test explained the effectiveness of the most elaborate synthesis model in matching a selection of musical-instrument sounds. The qualitative feedback was, in general, positive and enabled limitations in the system to be highlighted.

9.2 Contributions

The work documented in this thesis provides several contributions to the knowledge base of evolutionary computation and sound synthesis and specifically the field of unsupervised sound matching that intersects these two fields.

- In chapter four, a niching evolutionary algorithm was presented incorporating *k*-means cluster analysis into the evolutionary cycle of a conventional evolution strategy to preserve population diversity and enable solutions at multiple distinct optima to be optimised. The algorithm was named the clustering evolution strategy (CES) (Mitchell and Creasey, 2007).
- In chapter five a general model for a niching cooperative coevolutionary algorithm NCCEA was introduced that enables the baseline CCEA architecture to concurrently optimise and maintain multiple solutions at distinct optima.
- An instance of the NCCEA was also presented in chapter five which includes the CES within the architecture of the cooperative coevolutionary algorithm. The resulting algorithm was named the clustering cooperative coevolution strategy (CCCES).The CCCES was then demonstrated to optimise multiple distinct optima while preserving the convergence characteristics of the standard architecture.
- In chapter six, a windowed relative spectrum error measure was developed which addressed some of the difficulties associated with comparing sounds using conventional spectrum error measures (Mitchell and Pipe, 2005).
- In chapter seven, the contrived testing method was developed that enabled the matching method to be viewed and tested without interference from the synthesiser limitations. This testing method therefore enables effectiveness of each optimisation technique to be quantified and compared (Mitchell and Creasey, 2007).

- Also included in chapter seven, was a comparative study between standard EAs and the presented algorithms (CES and CCCES). The experiments revealed the capabilities of each algorithm when used to optimise the parameters of six frequency modulation synthesisers when matching both static and dynamic sounds (Mitchell and Sullivan, 2005), (Mitchell and Pipe, 2006) and (Mitchell and Creasey, 2007). Unlike previous studies, the synthesis models employed here were standard with unsimplified continuous parameters.

The EAs proposed within this thesis contribute to the field of evolutionary computation. Specifically, it is hoped that these contributions will assist practitioners to find multiple high-fitness solutions to complex, real-world problem domains. The application work detailed in chapters six and seven contributes to the field of sound synthesis, using EAs to derive synthesis parameters that reproduce given target sounds. The contrived matching method may be easily implemented in future studies to enable EAs to be compared quantitatively in application to related matching problems. In addition to the novel synthesis application and algorithmic contributions of this thesis, it is hoped that the contrived sound matching test method will serve as a useful tool in future matching developments. Furthermore, the method may also be used to estimate the relative difficulty in matching sounds with different synthesis types.

The proposed CES algorithm implements a speciation procedure by incorporating *MinMax* initialised *k*-means cluster analysis within the generational model of the ES. The resulting algorithm has been shown to produce particularly robust results when applied to complex, multimodal, multidimensional optimisation problems. The principles of the algorithms precursor, FCES, have been expanded into a novel niching algorithm that is able to preserve multiple solutions located at diverse regions of the search space. This diversity is preserved with the introduction of the new operators: *restricted cluster selection*, *hard intermediate recombination*, and *hard centroid recombination,* which enable clusters of solutions to evolve in isolation.

The CCCES provides a new cooperative coevolutionary development that facilitates the concurrent optimisation of multiple search space solutions at distinct optima. This was achieved with the introduction of a subpopulation niching algorithm, combined with a collaborative procedure that encourages the maintenance of diverse solutions. The diverse collaboration method maximises the amount of information extracted from the interaction space between coevolving subpopulations to promote increased performance when applied to multimodal function optimisation problems. The CCCES was found to consistently locate the optimum on the MTQ function, which was designed specifically to expose a weakness within the cooperative coevolutionary framework. The CCCES has been found to be particularly robust throughout all of the presented experiments, while exhibiting very fast rates of progress. The performance advantage is attained through the *dynamic linkage* procedure which samples the interaction space to determine optimal linkage between subpopulation species, such that each offspring need only participate in one collaboration when evaluated for fitness. This produces a cooperative coevolutionary algorithm, tuned for optimal collaboration, which has strong potential for optimisation in other real-world applications.

9.3 Future Work

This work presents an exploration of evolutionary computation applied to automatic sound matching with FM synthesis. As a complete system for matching sounds, some further development would have to be completed before the system could be used in a musical context, possibly in conjunction with an alternative user interface. At present, the computational implication of performing matches with dynamic sounds presents a major drawback. For example, in the final set of experiments in chapter seven, each generation of the CES requires the synthesis and analysis of exactly 2100 one second waveforms per generation. Each waveform requires the computation of six amplitude-weighted sinusoids and envelope generators. A single match required approximately 20 minutes computation using a Pentium 4 2.4GHz computer. However, as computational optimisation was not a priority in this work, there are numerous optimisations that could be applied to this model. Some ideas might be gleaned from Winduratna (1998), in which the Bessel function coefficients were encoded into the matching program, such that the spectra of the candidate sound could be synthesised directly into the frequency domain, circumventing the need for synthesis and time-frequency analysis. Additionally, within the same work, matches were performed using one simple FM configuration at a time, matching remaining partials with additional simple FM networks. Progressive matching techniques may prove more advantageous than evolving the entire synthesis space at once. However, this approach would require the model to be specialised, using domain-specific knowledge to enhance system performance; an approach which has been largely avoided in this research to keep the system as general as possible.

A further avenue for future work is the development of synthesis interfaces that use the matching algorithm to probe the underlying synthesis space. Some example interfaces were reviewed in chapter six, which could be employed to build a spectral profile for the proposed system to match. With sufficient optimisation, commands like *more brassy, brighter* and *less harmonic* may easily become available through the matching system.

Through analysis of the results in this thesis, the inability to effectively match target sounds may be attributed to the limitations of the underlying synthesiser and the effectiveness of the matching technique. The contrived matching method enables the capabilities of the matching synthesiser to be isolated from the matching technique. By focusing on sounds which can be achieved by the matching synthesiser, the effectiveness of different matching techniques can be examined. Once an effective sound navigation technique is confirmed, the next limitation in matching generalised sounds is the capabilities of the matching synthesiser. The accuracy of the evolved match is circumscribed by the capabilities of the synthesiser. It has been demonstrated that, with fixed population size, more accurate simulations can be evolved when the complexity of the synthesiser is increased. There are a number of ways the FM synthesiser could be further enhanced to improve the quality of matches. For example, the model could be enlarged with additional parallel simple FM elements, oscillators could be improved to include feedback and non-sinusoidal waveforms and time-variation could be enhanced with more sophisticated envelope generators. Furthermore, due to the generalised approach adopted here, in which target sounds are matched via an intermediate spectral representation without any FM-specific knowledge, the system could be easily modified to optimise the parameters of alternative synthesis techniques. For example, the FM synthesiser could be replaced by a physical modelling, waveshaping, granular or subtractive synthesiser, with only minor modifications to the rest of the system.

Closely related to the system limitations is the chosen representation of the target sound. In this study, the relative spectrum error is taken at 10 uniformly distributed time intervals. However, as noted in the listening tests in chapter eight, this approach may not be sufficient to characterise the changes in rapidly changing target sounds. Previous research has attempted to improve results by employing the full Short-Time Fourier Analysis representation (Riionheimo and Välimäki, 2003) or by biasing the distributions of time intervals such that more samples are positioned at the beginning of the sound (Horner, 1998). The former of these techniques may provide unnecessarily precise target sound representation when target sounds develop very slowly, while the latter presupposes that target sounds change fastest at the beginning, which may not be the case. A more 'intelligent' solution is to analyse the target sounds and place samples at regions of rapid change, to ensure that targets are represented to the required accuracy.

Different analysis techniques and similarity measures may also yield results that correlate more accurately with the perceptual distinctions of human listeners. As demonstrated in chapter eight, and also noted by Hon and Horner (2001), a small relative spectrum error between the target and candidate sound is not always a reliable indicator of perceived similarity. Better matches may be achieved when the similarity measure considers other factors of sound timbre that may include harmonics, spectral and amplitude envelope, inharmonics, noise components, equal loudness contours, masking and critical-bands. The intermediate representation of the target sound may also be derived by alternative analysis procedures to the Fast Fourier Transform; for example, the constant-Q transform (Brown, 1991), waveguide analysis (Tzanetakis *et al*, 2001) or Wigner Distribution (Preis and Georgopoulos, 1999) may result in a more tractable problem domain, and/or more accurate matching. Certainly, a comparative analysis between many of these different methods, and their corresponding affect on the complexity of the parameter-sound space mapping, would make a useful contribution to both the synthesis matching and content-based analysis research areas.

There is also a significant amount of development work to be completed with the proposed Niching Cooperative Coevolutionary Algorithm (NCCEA) architecture, specifically the CCCES, developed in this thesis. Further analysis could establish exactly what advantages this architecture might offer over the traditional CCEA architecture. Wiegand and Sarma (2004), in their analysis of spatially embedded CCEAs, attribute the observed performance increase over the baseline CCEA, to the model's ability to retain diversity within subpopulations, thus maintaining symmetrical rates of evolutionary change in the coevolving subpopulations. While this might also be the case for the CCCES, it is likely that the dynamic linking procedure introduces an additional factor. For example, Bucci and Pollack (2005) propose that traditional CCEA 'misbehaviour' is often caused by a lack of *informativeness* in the population. The maintenance of diversity by clustering subpopulations may then serve to retain information which, in turn, increases the likelihood of ideal collaboration. Further examination of the cooperative coevolution strategy could also be pursued in this context, as the rate of evolutionary change is intimately connected with the self-adaptive mutation operator, it may be that evolutionary balance, and/or *informativeness* are better retained using some alternative self-adaptation method.

The author hopes that this work goes some way to explore the potential of automatic self-programming synthesisers and can imagine a time when the synthesiser interface is entirely detached from underlying scientific processes of sound generation. The user may then use their preferred sound specification or navigation interface to control any known synthesiser type.

References

Affenzeller, M., Wagner, S. and Winkler, S. (2005) GA-Selection Revisited from an ES-Driven Point of View, *In:* Mira, J. and Álvarez, J.R. Eds. *First International Work-Conference on the Interplay Between Natural and Artificial Computation*, IWINAC 2005, Las Palmas, Canary Islands, Spain, June 15-18, 2005. Lecture Notes in Computer Science 3562 Springer, pp 262-271.

Anon (1987) DX7II *FD/D Digital Programmable Algorithm Synthesiser Owners Manual.* Yamaha Corporation.

Ashley, R. (1986) A Knowledge-Based Approach to Assistance in Timbral Design, *In: Proceedings of the 1986 International Computer Music Conference*, Royal Conservatory, Den Haag, Netherlands, pp 11-16.

Bäck, T. (1996) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford, UK: Oxford University Press.

Bäck, T., Eiben, A. E. and van der Vaart, N. A. L. (2000) An Empirical Study on GAs "Without Parameters". *In:* Schoenauer, M., Deb, K, Rudolph, G., Yao, X., Lutton, E., Merelo Guervós, J.J, Schwefel, H.-P. Eds. *Parallel Problem Solving from Nature - PPSN VI*, 6th International Conference, Paris, France, September 18-20, Lecture Notes in Computer Science, Springer, pp 315-324.

Bäck, T., Fogel, D. and Michalewicz, Z. (1997a) *Handbook of Evolutionary Computation*, UK: Oxford University Press.

Bäck, T., Hammel, U., and Schwefel, H.-P. (1997b) Evolutionary Computation: Comments on the History and Current State, *IEEE Transactions on Evolutionary Computation*, 1(1), pp 3-17.

Bäck, T., Rudolph, G., Schwefel H.-P. (1993) Evolutionary Programming and Evolution Strategies: Similarities and Differences. *In: Proceedings of the second Annual Conference on Evolutionary Programming*, pages 11--22, La Jolla, CA, USA, Evolutionary Programming Society, pp 11-22.

Bäck, T., and Schutz, M. (1996) Intelligent Mutation Rate Control in Canonical Genetic Algorithms, *In International Symposium on Methodologies for Intelligent Systems*, pp 158-167.

Bäck, T. and Schwefel, H. (1993) An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*. 1(1), pp 1-23.

Baker, J. E. (1985) Adaptive selection methods for genetic algorithms. *In* Grefenstette, J.J. Ed. *Proceedings of an International Conference on Genetic Algorithms and their Application*, Hillsdale, New Jersey, USA. Lawrence Erlbaum Associates, pp 101--111.

Beauchamp, J.W., and Horner, A. (2003) Error Metrics for Predicting Discrimination of Original and Spectrally Altered Musical Instrument Sounds. *The Journal of the Acoustical Society of America.* 114(4). pp 2325-2325.

Bersini, H., Doringo, M., Langerman, S., Seront, G., Gambardella, L. (Organising Committee) (1996) Results of the first international contest on evolutionary optimization (1st ICEO). *In: Proceedings of the 1996 IEEE International Conference on Evolutionary Computation.* pp 611-615.

Beyer, H.-G. (1995). Toward a theory of evolution strategies: On the benefit of sex – the ($\mu/\mu+\lambda$)-theory. *Evolutionary Computation*. 3(3). pp 311-347.

Beyer, H.-G. (2001) *The Theory of Evolution Strategies*, Springer.

Beyer, H. -G. and Schwefel, H. P. (2002) Evolution Strategies - A Comprehensive Introduction. *Natural Computing*. 1(1), pp 3-52.

Bezdek, J.C. (1973) *Fuzzy Mathematics in Pattern Classication*. Ph.D. thesis, Cornell University.

Bezdek J. C. (1981) *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA.

Bezdek, J.C. and Pal, S.K. (1992) *Fuzzy Models for Pattern Recognition: Methods That Search for Structures in Data*. New York: IEEE Press.

Bonham C. R. and Parmee I. C. (2004) Developments of the Cluster Oriented Genetic Algorithm (COGA), *Journal of Engineering Optimisation*, 36 (2); Taylor & Francis, 2004; pp 249-279

Brace, I. (2004) *Questionnaire Design: How to Plan, Structure and Write Survey Material for Effective Market Research*, London: Kogan Page.

Bristow-Johnson, R., (1996) Wavetable Synthesis 101, A Fundamental Perspective, *In: Proceedings of the 101st AES Convention*, Preprint 4400.

Brown, J.C., (1991). Calculation of a Constant Q Spectral Transform. *Journal of the Acoustical Society of America*, 89(1), pp 425-434.

Bucci, A. and Pollack, J. B. (2005) Identifying Global Optima in Cooperative Coevolution, *In: GECCO'05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation,* Washington D.C. USA, pp 539-544.

Bucci, A. (2007) Personal Communication.

Bull, L. (1997) Evolutionary Computing in Multi-Agent Environments: Partners. *In*: *Proceedings of the Seventh International Conference on Genetic Algorithms.* Morgan Kaufmann. pp 370-377.

Bull, L. (2001) On Coevolutionary Genetic Algorithms, *Soft Computing*, 3 (5). pp 201-207.

Burred, J. J., Lerch, A., and FitzGerald, D. (2004) Hierarchical Automatic Audio Signal Classification, *Journal of the Audio Engineering Society*, 52 (7/8), pp 724-739.

Cartlidge, J., and Bullock, S. (2003) Caring versus Sharing: How to Maintain Engagement and Diversity in Coevolving Populations, *In: Proceedings of the 7th European Conference on Artificial Life*, September 14th-17th, Dortmund, Germany, pp 299-308

Chowning, J. M. (1973) The synthesis of complex audio spectra by means of frequency modulation, *Journal of the Audio Engineering Society*. 21(7), pp 526-534.

Chowning J. M. and Bristow D. (1986) *FM Theory and Applications by Musicians for Musician*s, Yamaha Music Foundation.

Cilibrasi, R., de Vitanyi, P., Wolf, R. (2004) Algorithmic Clustering of Music. *In: Proceedings of the Fourth International Conference on Web Delivering of Music*, pp 110-117.

Cobb, H. G, (1990) An Investigation into the Use of Hypermutation as an Adaptive Operator in Genetic Algorithms Having Continuous, Time-Dependent Nonstationary Environments, 6760 (NLR Memorandum), Washington, D.C.

Cooley, J. W., and J. W. Tukey, (1965) An algorithm for the machine calculation of complex Fourier series, *Mathematics of Computation*, 19(90), pp 297-301

Dahlstedt, P. (2001) Creating and Exploring the Huge Space Called Sound: Interactive Evolution as a Composition Tool. *In: Proceedings of Music Without Walls?, Music Without Instruments? Conference*. De Montfort University, Leicester, UK.

Damavandi, N., and Safavi-Naeini , S. (2003) A Global Optimization Algorithm Based on Combined Evolutionary Programming/Cluster Analysis, *In*: *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, IEEE-CCECE, Montreal, Quebec, Canada, May, Volume 2 pp 1123- 1126.

Darwen , P. and Yao, X., (1996), Automatic Modularisation by Speciation, *In: Proceedings of the Third IEEE International Conference on Evolutionary Computation (ICEC'96)*. Nagoya, Japan. 20-22 May, pp.88-93.

Darwin, C. (1859) *The Origin of Species: By Means of Natural Selection or, The Preservation of Favoured Races in the Struggle for Life*, London: John Murray.

Davidor, Y, (1991) Epistasis Variance: A viewpoint on GA-hardness. *In:* Rawlins, G.J.E., *Foundations of Genetic Algorithms (FOGA),* California: Morgan Kaufmann, pp 23-35.

Dawkins, R, (1986) *The Blind Watchmaker,* London: Penguin.

De Jong, K.A. (1975) *An Analysis of the Behavior of a class of Genetic Adaptive Systems*. PhD thesis, University of Michigan.

De Jong, K. A. (1993) Genetic Algorithms are NOT Function Optimizers. *In:* Whitley, L. D., *Foundations of Genetic Algorithms 2 (FOGA2)*, California: Morgan Kaufmann, pp 5-18.

Deb, K. (2001) *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester: John Wiley and Sons.

Deb, K., and Beyer, H-G. (1999) *Self-adaptive Genetic Algorithms with Simulated Binary Crossover*, Technical report no. CI-61/99, University of Dortmund, Department of Computer Science/LS11.

Deb, Ka. and Goldberg, D. E., (1989) An Investigation of Niche and Species Formation in Genetic Function Optimization. *In: Proceedings of the 3rd International Conference on Genetic Algorithms*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., pp 42-50

Delprat, N. (1997) Global Frequency Modulation Laws Extraction from the Gabor Transform of a Signal: A First study of the Interacting Component Case, *In: IEEE Transactions of Speech and Audio Processing*, 5 (1), pp 64-71.

Downie, J. S. (2003) Music Information Retrieval, *Review of Information Science and Technology*, 37, Information Today Books, pp 295-340.

Downie, J. S. (2004) Introduction and Overview: A Sample of Music Information Retrieval Approaches, *Journal of the American Society for Information Science and Technology*, 55(12), pp 1033-1036.

Eiben, A. E., Bäck, T. (1997) An Empirical Investigation of Multiparent Recombination Operators in Evolution Strategies. *Evolutionary Computation*, 5(3), pp 347-365.

Eshelman, L. J. (1990) The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. *In: Foundations of Genetic Algorithms (FOGA).* pp 265-283

Etherington, R. and Punch, B. (1994), SeaWave: A system for musical timbre description. *Computer Music Journal*, 18(1), pp. 30—39.

Ficici, S. (2004) *Solution Concepts in Coevolutionary Algorithms,* PhD Thesis. Brandeis University

Fogel, D. B. (1992) *Evolving Artificial Life*. Ph.D Thesis, University of California, San Diego.

Fogel, L., Owens, A., and Walsh, M. (1966) *Artificial Intelligence Through Simulated Evolution,* New York: John Wiley Sons.

Gan, J. and Warwick, K. A. (1998*) Genetic Algorithm with Dynamic Niche Clustering for Multimodal Function Optimisation*, (Internal Report No. 98-001) Cybernetics Dept, Reading University.

Gan, J., and Warwick, K. (2001) Dynamic Niche Clustering: A Fuzzy Variable Radius Niching Technique for Multimodal Optimisation in Gas, *In: Proceedings of the 2001 Congress on Evolutionary Computation*, pp 215-222.

Garcia, R. (2000) Towards the Automatic Generation of Sound Synthesis Techniques: Preparatory Steps, *In: Proceedings of the 109th Convention of the Audio Engineering Society*, Los Angeles, CA, September 22 -25, 2000. Preprint 5186.

Garcia, R. (2001) Growing Sound Synthesizers using Evolutionary Methods. *In:* Bilotta, E., Miranda, E. R., Pantano, P. and Todd, P. M., editors, *Proceedings of ALMMA 2002 Workshop on Artificial Models for Musical Applications*, Editoriale Bios, Cosenza, Italy, pp 99-107,.

Garcia, R. (2002) Automatic Design of Sound Synthesis Techniques by means of Genetic Programming. *In: Proceedings of the 113th Convention of the Audio Engineering Society*. Los Angeles, CA, October 5-8. Preprint 5654.

Goldberg, D. E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning* Addison-Wesley Pub. Co.

Goldberg, D.E. and Deb, K (1991) A comparative Analysis of Selection Schemes Used in Genetic Algorithms, *In: The Proceedings of the First Workshop on Foundations of Genetic Algorithms*, California:Morgan Kaufmann, pp.69-93.

Goldberg, D. E. and Richardson, J. (1987) Genetic Algorithms with Sharing for Multimodal Function Optimization, *In: Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, Cambridge, Massachusetts, United States, Lawrence Erlbaum Associates, Inc. Mahwah, NJ, USA, pp 41-49.

Goldberg, D. E., and Wang, L. (1997). Adaptive Niching via Coevolutionary Sharing. *In: Genetic Algorithms and Evolution Strategy in Engineering and Computer Science*, Chapter 2, pp. 21—38, UK: John Wiley and Sons Ltd, pp 21-38.

Gonzalez., T. F. (1985) Clustering to Minimize the Maximum Intercluster Distance, *Theoretical Computer Science*, 38(2-3), pp 293–306.

Gordon, V.S. and Whitley, D. (1993) Serial and Parallel Genetic Algorithms as Function Optimizers, *In:* Forrest, S. Ed. *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, CA. Morgan-Kaufmann, pp.177-183.

Gorges-Schleuter, M. (1989) ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategy, *In: The Proceedings of the Third International Conference on Genetic Algorithms*, George Mason University, United States, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, pp.422-427.

Grefenstette, J. (1986) Optimization of Control Parameters for Genetic Algorithms, *In: IEEE Transactions on Systems, Man and Cybernetics*, 16(1), pp 122-128.

Grey, J. M. (1978) Timbre Discrimination in Musical Patterns. *Journal of the Acoustical Society of America*. 64, pp 467-472.

Hanagandi, V. and Nikolaou, M. (1998) A Hybrid Approach to Global Optimization using a Clustering Algorithm in a Genetic Search Framework, *Computers and Chemical Engineering*, 22(12), pp 1913-1925.

Harik, G. (1994) *Finding Multiple Solutions in Problems Of Bounded Difficulty*. IlliGal report no. 94002, University of Illinois, The Illinois Genetic Algorithms Laboratory, Urbana-Champaign.

Harik, G. R. (1995) Finding Multimodal Solutions Using Restricted Tournament Selection, *In: The Proceedings of the Sixth International Conference on Genetic Algorithms*, San Francisco, CA publisher Morgan Kaufmann, pp 24-31.

Harris, F.J (1978) On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform, *Proceedings of the IEEE*, 66(1), pp. 51-83.

He, J., Lan, M., Tan, C. L., Sung, S. Y. and Low, H. B. (2004) Initialization of Clusters Refinement Algorithms: A Review and Comparative Study, *In: Proceedings of the International Joint Conference on Neural Networks*.

Herdy, M. (1992) Reproductive Isolation as Strategy Parameter in Hierarchically Organized Evolution Strategies, *In: Parallel Problem Solving from Nature 2 (PPSN-II), Brussels, Belgium*, pp 207-217.

Herrera, F., Lozano, M. and Verdegay, J. L. (1998) Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*. 12 (4), pp 265-319.

Hillis, W. D. (1990) Co-evolving Parasites Improve Simulated Evolution as an Optimization Procedure. *In: CNLS'89: Proceedings of the Ninth Annual International Conference of the Center for Nonlinear Studies on Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks on Emergent computation*. Los Alamos, pp 228—234.

Himmelblau, D.M (1972) *Applied Nonlinear Programming*. New York: McGraw-Hill.

Hocaoglu, C. and Sanderson, A. C. (1997) Multimodal Function Optimization Using Minimal Representation Size Clustering and Its Application to Planning Multipaths. *Evolutionary Computation* 5(1). pp 81-104.

Holland, J. (1975) *Adaptation in Natural and Artificial Systems*, University Press, Ann Arbor, Michigan.

Horner, A. (1993) *Spectral Matching of Musical Instrument Tones*. PhD Thesis, University of Illinois Computer Science.

Horner, A. (1995) Wavetable Matching Synthesis of Dynamic Instruments with Genetic Algorithms. *Journal of the Audio Engineering Society*, 43(11), pp 916-931.

Horner, A. (1997) A Comparison of Wavetable and FM Parameter Spaces. *Computer Music Journal*, 21(4), pp 55-85.

Horner, A. (1998) Nested Modulator and Feedback FM Matching of Instrument Tones. *In: IEEE Transactions on Speech and Audio Processing*, 6(6), pp 398-409.

Horner, A. (2003) Auto-Programmable FM and Wavetable Synthesizers. *Contemporary Music Review,* 22(3), pp 21-29.

Horner, A., Beauchamp, J.W. and SO, R.H.Y. (2006) Error Metrics to Predict Discrimination of Original and Spectrally Altered Musical Instrument Sounds. *Journal of the Audio Engineering Society*. 54(3). pp 140-156.

Horner, A., Beauchamp, J. and Haken, L., (1993a) Machine Tongues XVI: Genetic Algorithms and Their Application to FM Matching Synthesis, Computer Music Journal, 17(4), pp 17-29.

Horner, A., Beauchamp, J. and Haken, L. (1993b) Methods for Multiple Wavetable Synthesis of Musical Instrument Tones, *Journal of the Audio Engineering Society*, 41(5), pp 336-356.

Howard, D. M. and Angus, J. 2000 *Acoustics and Psychoacoustics*. 2nd. Butterworth-Heinemann.

Huband, S., Hingston, P., While L. and Barone, L. (2003) An Evolution Strategy with Probabilistic Mutation for Multi-Objective Optimisation, *In: Proceedings of the IEEE Congress on Evolutionary Computation*. Canberra, December, pp 2284-2291.

Husbands, P (1994) Distributed Coevolutionary Genetic Algorithms for Multi-Criteria and Multi-Constraint Optimisation. *In: Evolutionary Computing, AISB Workshop*, pp 150-165.

Husbands, P. and Mill, F. (1991) Simulated Co-Evolution as the Mechanism for Emergent Planning and Scheduling. *In: Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, pages 264--270.

Hutter, M. (2002) Fitness Uniform Selection to Preserve Genetic Diversity. *In:* X. Yao, *Proceedings of the 2002 Congress on Evolutionary Computation* (CEC-2002), Washington D.C, USA, May 2002. IEEE, pp 783–788.

Ifeachor, E. C. and Jervis, B. W. (2002) *Digital signal processing : a practical approach*, 2nd ed., Harlow: Prentice Hall.

Iorio, A. and Li, X. (2002) Parameter Control Within a Cooperative Coevolutionary Genetic Algorithm, *In: Parallel Problem Solving From Nature VII (PPSN'02)*. Lecture Notes in Computer Science, Vol 2439. Springer Verlag. pp 247—256.

Jansen, T., and Wiegand, P. (2003) Exploring the Explorative Advantige of the Cooperative Coevolutionary (1+1) EA, *In: Proceedings of the 2003 Genetic and Evolutionary Computation Conference,* pp 310-321.

Jansen, T., and Wiegand, P. (2004) The Cooperative Coevolutionary (1+1) EA. *Evolutionary Computation*. MIT Press. 12(4). pp.405-434.

Johnson. C. G. (1999) Exploring the Sound-Space of Synthesis Algorithms Using Interactive Genetic Algorithms. *Proceedings of the AISB Symposium on Musical Creativity*, Edinburgh, pp 20-27.

Johnson, C. G. (2003) Exploring sound-space with interactive genetic algorithms. *Leonardo* vol 36 (1), MIT Press, pp51-54.

Johnson, C. G. and Gounaropoulos, A. (2006) Timbre Interfaces Using Adjectives and adverbs. *In: Proceedings of New Interfaces for Musical Expression NIME*. pp 101-102.

Justice , J. H. (1979) Analytic Signal Processing in Music Computation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-27(6), pp 670-684.

Karjalainen, M., Valimaki, V. and Tolonen, T. (1998) Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond. *Computer Music Journal* 22(3), pp.17-32.

Katsavounidis I. C. C., Kuo J., Zhang, Z. (1994) A New Initialisation Technique for Generalised Lloyd Iteration. *IEEE Signal Processing Letters*, 1(10), pp 144 – 146.

Kaufman, L and Rousseeuw, P.J. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley.

Koza J. R., 1992, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge: MIT Press.

Krishna, K. and Murty, M. Narasimha (1999) Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics.* 29(3). pp 433-439.

Krishnakumar, K. S., 1989, Micro Genetic Algorithms for Stationary and Nonstationary Function Optimization. *In: Proceedings of the SPIE Intelligent Control and Adaptive Systems Conference.*

Kronland-Martinet, R., Gillemain, P. and Ystad, S. (2001) From Sound Modelling to Analysis-Synthesis of Sounds, *Proceedings of Workshop on Current Research Directions in Computer Music*, Barcelona, Pompeu Fabra University, pp. 217-224.

LeBrun, M. (1977) A Derivation of the Spectrum of FM with a Complex Modulating Wave. *Computer Music Journal*. 1(4). pp 51-52

Legg, S., and Hutter, M., and Kumar, A. (2004) Tournament versus Fitness Uniform Selection. *In: Proceedings of the 2004 Congress on Evolutionary Computation (CEC 2004)*. pp 2144-2151.

Lim, S. M. and B. T. G. Tan. (1999) Performance of the Genetic Annealing Algorithm in DFM Synthesis of Dynamic Musical Sound Samples. *Journal of the Audio Engineering Society*. 47(5). pp 339-354.

Lin C.Y., Liou J.Y. and Yang Y.J. (1998) Hybrid Multimodal Optimization with Clustering Genetic Strategies. *Engineering Optimization*, 30, pp.263-80.

Lin, C.-Y. and Yang, Y.-J. (1998) Cluster Identification Techniques in Genetic Algorithms for Multimodal Optimization. *Computer–Aided Civil and Infrastructure Engineering*, 13(1), pp 53-62.

Lin, C.Y. and Wu, W.H. (2002) Niche Identification Techniques in Multimodal Genetic Search with Sharing Scheme. *Advances in Engineering Software*, 33(11-12), pp 779-791.

Lloyd. S. (1982) Least Square Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), pp 129-137.

Lohmann, R., (1991) Application of Evolution Strategy in Parallel Populations, *In: PPSN I: Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, Springer-Verlag, pp 198-208.

MacQueen, J.B. (1967) Some Methods for the Classification and Analysis of Multivariate Observations. *In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. pp 281-297.

Mahfoud, S. W. (1992) Crowding and Preselection Revisited, *In: Parallel Problem Solving from Nature PPSN2*, publisher North-Holland, Amsterdam, pp 27-36.

Mahfoud, S. W. (1995) *Niching Methods for Genetic Algorithms*. PhD Thesis. Urbana, IL, USA.

Mandelis, J. (2002) Adaptive Hyperinstruments: Applying Evolutionary Techniques to Sound Synthesis and Performance. *In: Proceedings of the 2004 Conference on New Interfaces for Musical Expression*, pp 190-191.

Mandelis, J. (2003a) Genophone: Evolving Sounds and Integral Performance Parameter Mappings, *In: Applications of Evolutionary Computing, EvoWorkshops*, pp 535-546.

Mandelis J. and Husbands, P. (2003b) Musical Interaction with Artificial Life Forms: Sound Synthesis and Performance Mappings. *In: Contemporary Music Review*, 22(3), pp.69-77.

Mandelis J. and Husbands, P. (2004) Don't just play it, grow it!: breeding sound synthesis and performance mappings. In: *NIME'04: Proceedings of the 2004 Conference on New Interfaces for Musical Expression* , Hamamatsu, Shizuoka, Japan, National University of Singapore pp 47-50.

Manderick, B. and Spiessens, P., (1989) Fine-Grained Parallel Genetic Algorithms. *In: Proceedings of the 3rd International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, pp 428-433.

Massey, H. (1986) *The Complete DX7*, Amsco Press.

Mathias, K. E., Schaffer, J. D., Eshelman, L. J. and Mani, M. (1998) The Effects of Control Parameters and Restarts on Search Stagnation in Evolutionary Programming. *In PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature publisher*. Springer-Verlag, pp 398-407.

Mauldin, M. L. (1984) Maintaining Diversity in Genetic Search. *In: Proceedings of the National Conference on Artificial Intelligence*, pp 247-250.

McDermott, J., Griffith, N. J. L. and O'Neill, M., (2005) Toward User-Directed Evolution of Sound Synthesis Parameters. *In: EvoWorkshops*, pp 517-526.

McDermott, J., Griffith, N. J.L., and O'Neill, M. (2006) Timbral, Perceptual, and Statistical Attributes for Synthesized Sound. *In: Proceedings of the International Computer Music Conference (ICMC)*, New Orleans, November.

Miller, B. L. and Shaw, M. J. (1996) Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization. *In: Proceedings of the International Conference on Evolutionary Computation*, pp 786-791.

Miller, B. L., and Goldberg, D. E. (1995) Genetic Algorithms, Tournament Selection, and the Effects of Noise. *Complex Systems,* 9, pp 193-212.

Miranda, E. R. (1998) Striking the right note with ARTIST: an AI-based synthesiser, *In:* Chemillier, M. and Pachet, F. Eds. *Informatique Musicale*. Paris (France): Editions Hermes.

Miranda, E. R. (2002) *Computer Sound Design: Synthesis Techniques and Programming*. Second Edition, Focal Press.

Mitchel, T.J. and Creasey, D.P. (2007) Evolutionary Sound Matching: A Test Methodology and Comparative Study. *Proceedings of The Sixth International Conference on Machine Learning and Applications* (ICMLA'07), pp 229-234.

Mitchell, T. and Pipe, A. G. (2006) A Comparison of Evolution Strategy-based Methods for Frequency Modulated Musical Tone Timbre Matching. *In: Proceedings of the 7th International Conference on Adaptive Computing in Design and Manufacture, (ACDM),* Bristol, pp. 167-175.

Mitchell, T. J. and Pipe, A. G. (2005) Convergence Synthesis of Dynamic Frequency Modulation Tones Using an Evolution Strategy, *In: EvoWorkshops*. pp 533-538.

Mitchell, T. J. and Sullivan, C. W. (2005) Frequency Modulation Tone Matching Using a Fuzzy Clustering Evolution. *In: Proceedings of the 118th Convention of the Audio Engineering Society*. Barcelona, Preprint 6366.

Mtihlenbein, H., and Schlierkamp-Voosen, D. (1993) Predictive Models for the Breeder Genetic Algorithm I: Continuous Parameter Optimization. *Evolutionary Computation,* 1(1), pp25-49.

Mühlenbein, H. (1992) Asynchronous Parallel Search by the Parallel Genetic Algorithm. *In: Third IEEE Symposium on Parallel and Distributed Products*, pp 526-533.

Nicol, C. and Brewster, S. A. and Gray, P. D. (2004) Designing Sound: Towards a System for Designing Audio Interfaces using Timbre Spaces. *In: Proceedings of the 10th Meeting of the International Conference on Auditory Display*. Sydney, Australia, July 6-9, pp 1–5.

Opolko, F. and Warpnick J. (1989) *McGill University Master Samples (MUMS)*, 11 CD-ROM set, Faculty of Music, McGill University, Montreal, Canada.

Ostermeier,A., Gawelczyk, A. and Hansen, N. (1994) A Derandomised Approach to Self-Adaptation of Evolution Strategies. *Evolutionary Computation*. 2(4), pp 369-380.

Panait, L., Wiegand, R. P. and Luke, S. (2003) Improving Coevolutionary Search for Optimal Multiagent Behaviors. *In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-2003),* pp 653-658.

Panait, L. and Wiegand, R. P. and Luke, S. (2004) A Sensitivity Analysis of a Cooperative Coevolutionary Algorithm Biased for Optimization. *In GECCO'04: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp 573-584.

Parkin, D. T. (1979) An *Introduction to Evolutionary Genetics, A Series of Student Texts in Contemporary Biology*. Baltimore, University Park Press.

Payne, R. (1987) A Microcomputer Based Analysis/Resynthesis Scheme for Processing Sampled Sounds Using FM. *In: Proceedings of the International Computer Music Conference, San Francisco*. CA. pp. 282 289.

Pétrowski, A. (1996) A Clearing Procedure as a Niching Method for Genetic Algorithms. *In: Proceedings of the International Conference on Evolutionary Computation*, pp 798-803.

Pelleg, D. and Moore, A. (2000) X-means: Extending K-means with Efficient Estimation of the Number of Clusters. *In: Proceedings of the Seventeenth International Conference on Machine Learning,* pp 727-734.

Peña, A. and Lozano, J. and Larran, J. (1999) An Empirical Comparison of Four Initialization Methods for the *K*-means Algorithm. *Pattern Recognition Letters*, 20, pp 1027-1040.

Popovici, E. and De Jong, K. (2004) Understanding Competitive Co-Evolutionary Dynamics Via Fitness Landscapes. *In:* Luke,S. Ed. *Artificial Multiagent Learning Workshop at the AAAI Fall Symposia - AAAIFS-04.*

Popovici, E. and De Jong, K. (2005) A Dynamical Systems Analysis of Collaboration Methods in Cooperative Coevolution, *In: Proceedings of the AAAI 2005 Fall Symposium on Coevolutionary and Coadaptive Systems.*

Popovici, E. and De Jong, K. (2005b) Understanding Cooperative Co-evolutionary Dynamics via Simple Fitness Landscapes, *In GECCO'05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp 507-514.

Potter, M. A., Meeden, L. and Schultz, A. C. (2001) Heterogeneity in the Coevolved Behaviors of Mobile Robots: The Emergence of Specialists, *In: Proceedings of the 17th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann, pp 1337-1343.

Potter, M. A. (1997) *The Design and Analysis of a Computational Model of Cooperative Coevolution*, Ph.D. Thesis, George Mason University, Fairfax, VA.

Potter, M. A., and De Jong, K. A. (1994) A Cooperative Coevolutionary Approach to Function Optimization. *In PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature*. pp 249-257.

Potter, M. A., and De Jong, K. A. (1998) The Coevolution of Antibodies for Concept Learning. *In: proceeding of the Parallel Problem Solving from Nature*, pp. 530-539.

Potter, M. A., and De Jong, K. A. (2000) Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation*. 8(1), pp 1-29.

Preis, D. and Georgopoulos, V. C. (1999) Wigner Distribution Representation and Analysis of Audio Signals: An Illustrated Tutorial Review, *Journal of the Audio Engineering Society*. 47(12) pp. 1043-1053.

Preuss, M., Schönemann, L. and Emmerich, M. (2005) Counteracting Genetic Drift and Disruptive Recombination in $(\mu \overset{+}{,} \lambda)$-EA on multimodal fitness landscapes. *In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2005).* Washington D.C, New York: ACM Press, pp.865-872.

Rechenberg, I. (1965) *Cybernetic Solution Path of an Experimental Problem*, Techhnical report RAE Translation 1122. Farnborough, Hants.

Rechenberg, I. 1973. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Frommann-Holzboog.

Riionheimo, J. and Välimäki, V. (2003) Parameter Estimation of a Plucked String Synthesis Model Using a Genetic Algorithm with Perceptual Fitness Calculation. *In: EURASIP Journal on Applied Signal Processing*. pp.791-805.

Risberg, J. S., 1980, Non-Linear Estimation of FM Synthesis Parameters. *In: Proceedings of the 67th Convention of the Audio Engineering Society.* New York. October, Preprint 1685.

Roads, C., (1996) *The Computer Music Tutorial*. The MIT Press.

Roberts, M. E. and Claridge, E. (2004), Cooperative Coevolution of Image Feature Construction and Object Detection, *In: Parallel Problem Solving from Nature*. pp 902-911.

Rosin, C. D. and Belew, R. K. (1995) Methods for Competitive Co-evolution: Finding Opponents Worth Beating. *In: Proceedings of the Sixth International Conference on Genetic Algorithms*, pp 373-380.

Rosin, C. D. and Belew, R. K. (1997) New Methods for Competitive Coevolution. *In: Evolutionary Computation*. 5 (1). pp 1-29.

Rothlauf, F.; Branke, J.; Cagnoni, S.; Corne, D.W.; Drechsler, R.; Jin, Y.; Machado, P.; Marchiori, E.; Romero, J.; Smith, G.D.; Squillero, G. (Eds.), (2005) *Applications of Evolutionary Computing, EvoWorkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoSTOC*, Lausanne, Switzerland, March 30 - April 1, 2005, Proceedings. Lecture Notes in Computer Science 3449 Springer.

Rowe, J.E. and Hidović, D. (2004) An Evolution Strategy Using a Continuous Version of the Gray-code Neighbourhood Distribution, *In:* K. Deb et al. (Eds.). *Genetic and Evolutionary Computation—GECCO*. Springer. Heidelberg, pp. 725-736.

Rudolph, G (1991) Global Optimization by Means of Distributed Evolution Strategies. *In: Proceedings of First Conference on Parallel Problem Solving from Nature PPSNI.* Springer, pp 209-213.

Schaffer, J. D., Caruana, R., Eshelman, L. J., and Das, R. (1989) A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. *In: Proceedings of the 3rd international Conference on Genetic Algorithms*, pp 51-60.

Schönemann, L., Emmerich, M., and Preuss, M. (2004) On the Extinction of Evolutionary Algorithms Sub-populations on Multimodal Landscapes. *Informatica (Sl), Special Issue on Bioinspired Optimization.*

Schottstaedt, B. (1977) The Simulation of Natural Instrument Tones Using Frequency Modulation with a Complex Modulating Wave, *Computer Music Journal*, 1(4), pp 46-50.

Schwefel, H. -P. (1981) *Numerical Optimisation of Computer Models*, New York: John Wiley and Sons, Inc.

Schwefel, H.-P. (1987) Collective Phenomena in Evolutionary Systems, in Problems of Constancy and Change. *In: The Complementarity of Systems Approaches to Complexity, Proceedings of the 31st Annual* Meeting, Volume 2, pp 1025-1033.

Schwefel, H.-P. (1995) *Evolution and Optimum Seeking*, New York: Wiley Interscience, series Sixth-Generation Computer Technology.

Schwefel, H.-P. and Rudolph G. (1995) Contemporary Evolution Strategies. *In: Proceedings of the Third European Conference on Advances in Artificial Life*, Lecture Notes in Artificial Intelligence, Springer, Berlin, pp 893-907.

Schwefel, H.-P. and Bäck, T. (1997) Artificial Evolution: How and Why?, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*. John Wiley, Chichester, New York. pp. 1-19.

Shir O. M. and Bäck, T. (2005) Niching in Evolution Strategies, *In: GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*. pp 915 – 916.

Smith, J., and Fogarty, T. C. (1996) Self Adaptation of Mutation Rates in a Steady State Genetic Algorithm. *In: International Conference on Evolutionary Computation*, pp 318-323.

Smith, R. and Gray, B., (1994) Co-adaptive Genetic Algorithms, an Example in Othello Strategy. *In: Proceedings of the Florida Artifical Intelligence Symposium.*

Smith, S. W. (2003) *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. London: Newnes.

Snarey, M., Terrett, N. K., Willett, P. and Wilton, D. J. (1997) Comparison of Algorithms for Dissimilarity-Based Compound Selection. *In: Journal of Molecular Graphics and Modelling*, 15(6), New York: Elsevier Science Inc, pp 372-385.

So, K.F. and Horner, A. (2004) Wavetable Matching of Pitched Inharmonic Instrument Tones, *Journal of the Audio Engineering Society*, 52(5), pp 516-529.

Sprave, J. (1994) Linear Neighbourhood Evolution Strategy. *In: Proceedings of the Third Annual Conference on Evolutionary Programming (EP '94)*. pp 42 -51.

Streichert, F., Stein, G, Ulmer, H., and Zell, A. (2003) A Clustering Based Niching EA for Multimodal Search Spaces. *In: Proceedings of the International Conference Evolution Artificielle*, Lecture Notes in Computer Science 2936, pp 293-304.

Sullivan, J. C. W., (2001) *An Evolutionary Computing Approach to Motor Learning with an Application to Robot Manipulators*. PhD Thesis, University of the West of England, Bristol.

Syswerda, G. (1989). Uniform Crossover in Genetic Algorithms. *In:* Schaffer, H. Ed. *Proceedings of the 3rd International Conference on Genetic Algorithms, San* Mateo: Morgan Kaufmann, pp 2-9.

Tan, B. T. G. and S. M. Lim, (1996) Automated Parameter Optimisation for Double Frequency Modulation Synthesis Using the Genetic Annealing Algorithm. *In: Journal of Audio Engineering Society*, 44, (1/2) pp 3-15.

Tan, B. T. G., Gan, S. L., Lim, S. M. and Tang, S. H. (1994) Real-Time Implementation of Double Frequency Modulation (DFM) Synthesis. *Journal of the Audio Engineering Society*. 42(11). pp 918-926.

Tanese, R., (1989) Distributed Genetic Algorithms. *In: Proceedings of the 3rd International Conference on Genetic Algorithms*. pp 434-439.

Törn, A. (1977) Cluster Analysis Using Seed Points and Density-Determined Hyperspheres as an Aid to Global Optimization. *In: IEEE Transactions on Systems, Man and Cybernetics*, 7, pp 610-616.

Törn, A.A. (1986) Clustering Methods in Global Optimization, *In: Preeprints of the Second IFAC Symposium on Stochastic Control*, Vilnius, 19-23 May, Part 2, pp138-143.

Tsutsui, S. and Fujimoto, Y. (1993), Forking Genetic Algorithm with Blocking and Shrinking Modes (fGA*). In: Proceedings of the 5th International Conference on Genetic Algorithms*. pp 206-215.

Typke, R., and Wiering, F., and Veltkamp, R. C. (2005) A Survey of Music Information Retrieval Systems. *In: ISMIR 2005: Proceedings of the 6th International Conference on Music Information Retrieval*. London, UK, 11-15 September, pp 153-160.

Tzanetakis, G., Essl, G., Cook, P. (2001) Audio Analysis using the Discrete Wavelet Transform. *In: Proceedings of WSES International Conference on Acoustics and Music: Theory and Applications*, Skiathos, Greece.

Ursem, R. K., (1999) Multinational Evolutionary Algorithms. *In: Proceedings of the Congress on Evolutionary Computation*. 3(6-9), IEEE Press, pp 1633-1640.

Vertegaal, R. and Bonis, E. (1994) ISEE: An Intuitive Sound Editing Environment. *Computer Music Journal* 18(2), pp, 21-29.

Wehn. K. (1998) Using Ideas from Natural Selection to Evolve Synthesized Sounds. *In: Proceedings of the Digital Audio Effects DAFX98*. Barcelona, pp 159-167.

Whitley, L. D. (1991) Fundamental Principles of Deception in Genetic Search. *In: Foundations of Genetic Algorithms*, pp 221-241.

Whitley, D. (1994) A Genetic Algorithm Tutorial. *Statistics and Computing*, 4, pp 65-85.

Whitley, D., Rana, S., and Heckendorn, R. (1999) Island Model genetic Algorithms and Linearly Separable Problems. *Journal of Computing and Information Technology,* 7 (1) pp 33 – 47.

Whitley, L. D. and Kauth, J. (1988) GENITOR: A Different Genetic Algorithm. *In: Proceedings of the Rocky Mountain Conference on Artificial Intelligence,* pp.118-130.

Whitley, D. and Starkweather, T. (1990) GENITOR II.: A Distributed Genetic Algorithm. *In: Journal of Experimental and Theoretical Artificial Intelligence.* 2(3). pp 189-214.

Wiegand, R. P., (2004), *An Analysis of Cooperative Coevolutionary Algorithms*. Ph.D. Thesis, George Mason University, Fairfax, VA.

Wiegand, R. P., 2005, Robustness in Compositional Coevolution. *In Proceedings of the AAAI 2005 Fall Symposium on Coevolutionary and Coadaptive Systems*. AAAI Press.

Wiegand, R., M. Potter, D. Sofge, and W. Spears (2006) A Generalized Graph-Based Method for Engineering Swarm Solutions to Multiagent Problems. *In: Parallel Problem Solving from Nature*, pp 741-750.

Wiegand, R. P., Liles, W. C., and De Jong, K. A. (2001) An Empirical Analysis of Collaboration Methods in Cooperative Coevolutionary Algorithms. *In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001),* pp 1235-1242.

Wiegand, R. P., Liles, W. C., and De Jong, K. A. (2002) Analyzing Cooperative Coevolution with Evolution Game Theory. *In: Proceedings of the the 2002 International Congress On Evolutionary Computation,* IEEE Press, 2, pp 1600-1605.

Wiegand, R. P., and Sarma, J. (2004) Spatial Embedding and Loss of Gradient in Cooperative Coevolutionary Algorithms, *In: Parallel Problem Solving from Nature (PPSN VIII),* Lecture Notes in Computer Science, pp 912-921.

Winduratna, B. (1998), FM Analysis/Synthesis Based Audio Coding. *In: Proceedings of the 104th convention of the Audio Engineering Society*, Amsterdam, Preprint 4746.

Wolpert, David H. and Macready, William G. (1997), No Free Lunch Theorems for Optimization. *In: IEEE Transactions on Evolutionary Computation*. 1(1). pp 67-82.

Wright, A.H. (1991). Genetic Algorithms for Real Parameter Optimization. *In:* Rawlins, G.E. Ed. *Foundations of Genetic Algorithms. 3*. Morgan Kauffmann. pp 205-218.

Wright, S., (1969), *Evolution and the Genetics of Populations, IV. Variability Within and Among Natural Populations*. University of Chicago Press, Chicago.

Wun, C.W. and Horner, A., (2001), Perceptual Wavetable Matching Synthesis of Musical Instrument Tones. *Journal of the Audio Engineering Society*. 49(4), pp 250-262.

Xie, X.L. and Beni, G., (1991). A Validity Measure for Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 13(8). pp 841-847.

Yang, C. (2001) *Music Database Retrieval Based on Spectral Similarity*, Stanford University Database Group Technical Report.

Yang, J.M., and Kao, C.Y. (2000) Integrating Adaptive Mutations and Family Competition Into Genetic Algorithms as Function Optimizer. *Soft Computing*, 4(2). pp 89-102.

Yao, X., and Liu, Y. (1997) Fast Evolution Strategies. *Evolutionary Programming VI*, Berlin: Springer, pp 151-161.

Yee-King, M. (2000), *AudioServer - Collaborative, Interactive Genetic Algorithm for Sound Design*. MSc Thesis, University of Sussex, Brighton.

Yin, X. and Germay, N. (1993) A Fast Genetic Algorithm with Sharing Scheme Using Cluster Analysis Methods in Multimodal Function Optimization. *In: Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*. Berlin: Springer-Verlag, pp. 450-457.

Zadeh,L.A. (1965) Fuzzy Sets. *Information and Control*, 8, pp 338-353.

Zhang, H., Ho, T.B., and Lin, M.S. (2004) An Evolutionary K-Means Algorithm for Clustering Time Series Data. *In Proceedings of the Third International Conference on Machine Learning and Cybernetics*, IEEE Press, pp.1282-1287.

Zolzer, U. (1997), *Digital Audio Signal Processing*, Chichester : Wiley.

Zwicker, E. and Fastl, H. (1999) *Psychoacoustics. Facts and Models - Second Updated Edition*. Berlin: Springer.

Appendix 1 - Listening Test 1 – Results

This appendix provides the full rests to the listening test described in section 8.3. Listed is the relative spectrum error of each tone, the analytical ranking results (performed by relative spectrum error) and the ranking results of each test subject.

Target Sound 1

Relative Spectrum Error	Ranking						
	Analytical	Subject: 1	Subject: 2	Subject: 3	Subject: 4	Subject: 5	Subject: 6
0.000578	1	1	1	1	1	1	1
0.30887	2	2	2	3	2	3	2
0.505066	3	3	3	2	3	2	3
0.815398	4	5	5	5	4	4	4
0.915851	5	4	4	4	5	5	5

Target Sound 2

Relative Spectrum Error	Ranking						
	Analytical	Subject: 1	Subject: 2	Subject: 3	Subject: 4	Subject: 5	Subject: 6
0.179469	1	1	1	1	1	1	1
0.258194	2	2	2	2	2	2	2
0.357446	3	5	3	3	3	3	3
0.507881	4	3	4	4	4	4	4
0.794542	5	4	5	5	5	5	5

Target Sound 3

Relative Spectrum Error	Ranking						
	Analytical	Subject: 1	Subject: 2	Subject: 3	Subject: 4	Subject: 5	Subject: 6
0.148244	1	1	1	1	1	1	1
0.27958	2	3	3	3	3	3	3
0.444212	3	2	2	2	2	2	2
0.668059	4	4	4	4	4	4	4
0.851414	5	5	5	5	5	5	5

Target Sound 4

Relative Spectrum Error	Ranking						
	Analytical	Subject: 1	Subject: 2	Subject: 3	Subject: 4	Subject: 5	Subject: 6
0.103175	1	1	1	1	1	1	1
0.200043	2	2	2	2	3	2	2
0.529532	3	3	3	3	2	3	3
0.621401	4	4	5	5	4	4	5
0.728357	5	5	4	4	5	5	4

Target Sound 5

Relative Spectrum Error	Ranking						
	Analytical	Subject: 1	Subject: 2	Subject: 3	Subject: 4	Subject: 5	Subject: 6
0.000578	1	1	1	1	1	1	1
0.30887	2	2	2	2	2	2	2
0.505066	3	3	3	3	3	3	3
0.815398	4	4	4	4	5	4	4
0.915851	5	5	5	5	4	5	5