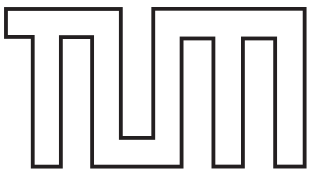

Technische Universität München
Chair of Media Technology
Prof. Dr.-Ing. Eckehard Steinbach

Master Thesis

Automatic Audio Encoder Tuning Using Evolutionary Algorithms

Author:	Sebastian Wozny
Matriculation Number:	03609721
Address:	Karlsbader Weg 12
	86415 Mering
Supervising Professor:	Prof. Dr.-Ing. Eckehard Steinbach
Supervisers (Dolby Germany):	Dr. Arijit Biswas
	Michael Schug
Begin:	01.10.2015
End:	21.04.2016

With my signature below, I assert that the work in this thesis has been composed by myself independently and no source materials or aids other than those mentioned in the thesis have been used.

München, April 18, 2016 _____

Place, Date Signature

München, April 18, 2016 _____

Place, Date Signature

Abstract

The aim of this thesis is to investigate the potential of evolutionary algorithms for automatically tuning sophisticated perceptual audio codecs. In particular, the effect of changing the transform lengths for the time frequency analysis and the control of the bit reservoir of the Dolby AC-4 encoder are tested.

In the first part, an overview of the fundamentals of psychoacoustics and optimization techniques is presented. In the implementation part a, decoupled distributed system is described, which implements a process to optimize the perceptual audio quality. In the process, several parameters in the Dolby AC-4 encoder are modified and the effects of the modifications are evaluated. The feedback of objective measurement tools is used to heuristically determine possible better parameter values. After a fixed amount of iterations, a set of optimal parameter values is returned.

A number of experiments that are conducted on the system to evaluate the impact of parameter optimization. The perceptual audio quality is evaluated using objective measurement tools and validated using MUSHRA listening tests with expert listeners as test subjects.

The results of the experiments indicate that perceptual audio quality improvement was achieved. Both the results acquired with objective measurement tools and the subjective listening experience tests indicate that for some signals audible improvement was achieved. In conclusion, the thesis supports, that parts of perceptual audio encoders can be efficiently tuned with genetic algorithms, a particular class of evolutionary algorithms.

Keywords: Perceptual audio codec, evolutionary algorithm, genetic algorithm, audio encoder tuning, PEAQ, DEAP, distributed system

Contents

Chapter 1

Introduction

1.1 Audio Coding

Perceptual audio encoding, leveraging the psychoacoustic effects and properties of the human ear, has revolutionized the audio compression landscape in the last quarter of the century. Starting with the first commercially successful encoding, MPEG-1 Audio Layer III, or MP3, perceptually compressed (lossy) digital audio formats surpassed any other form of digital audio playback media [Bra99]. Even though our audio processing and playback techniques have become more sophisticated, current technology's storage and transmission capacity is limited. The widespread use of mobile streaming devices has spurred the demand for ubiquitous availability of audio everywhere. It motivates research to create ever more sophisticated perceptual audio codecs. Complex perceptual audio codecs today contain hundreds of heuristically determined thresholds, to achieve reasonable audio quality at bit rates low enough, to be suitable for applications on limited transmission bandwidth [SPA06].

The first solutions of audio recording, storage and playback date back to the mid-nineteenth century. In the past phonographs, vinyl plates and audio CDs mark several milestones of audio technology. The next great revolution of audio storage technologies is the discovery of perceptual audio coding. Perceptual audio coding made it possible to overcome the storage challenges that storing uncompressed digital audio collections creates. Audio compression can be divided in lossless and lossy compression, which take advantage of the redundancy and irrelevancy of the information present in the audio respectively. Perceptual audio encoding is a lossy audio compression technique, which achieves higher compression ratios than lossless compression, like MPEG-4 ALS [LMH+05] or Dolby TrueHD, by modeling the audibility of sounds according to the way the human ear processes sensory information. The psychoacoustic model of the ear has been studied for many years [Gre60]. Perceptual algorithms model the range of sounds that will not be perceived by the human ear due to various psychoacoustic effects. This information is commonly referred to as "irrelevancy",

since it cannot be detected, not even by a sensitive listener in a quiet environment [SPA06]. Perceptual coding thereby drastically reduces the amount of irrelevant information in the audio signal. The encoded result of this process, can then be subjected to further lossless compression by means of entropy coding techniques, like Huffman Coding.

By modelling the irrelevant information in the signal, a level of acceptable degradation in comparison to the original uncompressed file is defined. At high bit rates, no degradation is introduced, and the compressed and uncompressed audio sound the same. At lower bit rates, it is not always possible to avoid the introduction of audible quantization noise, which can be perceived by a human listener to detect the degradation as coding artifacts. Coding artifacts affect the listening experience of the listener negatively, and low bit rate coding algorithms strive to minimize these adverse effects.

Substantial progress in the field of low bit rate perceptual encoding has been achieved and codecs like High Efficiency Advanced Audio Coding (HE-AAC) [WKHP03] allow good to excellent audio quality at bit rates as low as 64 kbit/s stereo. Current research attempts to continue reducing bit rates further. At the same time, it is attempted to reduce the amount of artifacts at very low bit rates to a minimum, in order to provide an excellent listening experience for consumers, even when on restricted bandwidth network connections, like cellular networks for smartphones.

1.2 Challenges of Audio Encoder Development

Even at low bit rates, highly efficient codecs like HE-AAC, are approaching the limits of perceptual coding efficiency today. Therefore the further development of newer codecs like the Dolby AC-4 codec [KRW+16], has become increasingly difficult and complex.

The psychoacoustic model of the human ear has been empirically studied, and perceptual algorithms attempt to reduce the coding noise level to inaudible levels. The relationships between many different parameters in a perceptual audio coding algorithm and the effects of changing these parameters on subjective listening experience are poorly understood, and highly non-linear. The underlying error surface of the optimization problem is presumably equally non-linear, and has a multitude of local optima.

In order to test audio quality improvements, audio tuning experts introduce changes to the source code of an audio codec, build the necessary tools from the modified code base and prepare tests to evaluate the quality of the algorithm. Typically the quality of perceptual coding algorithms is evaluated using subjective criteria [Rec03]. The participants in the listening tests often undergo thorough training in order to be able to detect even small differences and degradation. However, formal subjective tests are often expensive or impractical to set up [TTB+00].

It is apparent that the current approach to audio encoder quality tuning and optimization is

very resource and time intensive and that the scarcity of expert listeners is a limiting factor in the development of more efficient perceptual audio coding algorithms. Therefore, an objective measurement method is needed that models the sensory and cognitive processes underlying subjective evaluation by listeners [TTB+00]. Several objective measurement methods have been proposed since 1979 [TTB+00], and in 2001 the objective measure method "Perceptual Evaluation of Audio Quality", in short PEAQ was standardized by the International Telecommunications Union (ITU) as the standard for objective measurement of perceived audio quality.

PEAQ was calibrated using listening tests and an improvement reported by PEAQ has been found to correlate well with audible improvements in perceptual audio quality [TTB+00]. PEAQ gives a measure of the degradation of an excerpt that has been subjected to perceptual audio coding in comparison to the known original excerpt.

Objective measurement methods, while not as accurate as a human expert listener, can provide quick feedback and the process can be easily automated. With the guidance of objective measurement methods, computer programs can perform experiments and indicate possible changes to tune perceptual audio encoders autonomously. To efficiently assist experts, computers must be able to generalize and learn patterns from data. The field concerned with the construction of algorithms that can learn from and make predictions on data is called machine learning [KP98]. The discipline of machine learning includes many different principles and algorithms. The core objective of a learning agent is to generalize its experience [Bis06]. With increasing experience, the generalization capability improves and the predictions of the learner become more accurate. Eventually, performance as defined by the experimenter, is gradually ameliorated. Machine learning can be split into supervised and unsupervised learning. In unsupervised learning, i.e. when training data is unavailable, the quality of a trial solution is often estimated directly as a property from the data set, e.g. the separation of data in a clustering algorithm. In supervised learning, the basis for the machine learning algorithm is an example set of trial-solutions with associated solution-quality values. Starting from this so-called training-set, the task of the algorithm is to generalize.

A machine learning algorithm needs to be able to determine the quality of a proposed problem solution, to be able to judge decisions and make predictions. When it has made a certain prediction, the performance of that trial solution needs to be assessed, so the algorithm gains feedback about whether it has made a good or a bad decision. The availability and quality of objective measurement tools is the key to applying machine learning to audio encoder tuning. With it, an objective function is defined that assigns a real-valued quality metric to trial solutions of the optimization algorithm. Over successive iterations, a good learning algorithm will find optimal input values for the objective function, and reach a global optimum with high probability.

Examples of machine learning algorithms include artificial neural networks, which try to mimic the biological structure of human neurons. Another area of machine learning is concerned with clustering algorithms, which increase the separability of data. This is often

useful to find patterns and extract information in high dimensional problems. The area of particular interest for this work are genetic algorithms (GA) as a subclass of evolutionary algorithms (EA),which base on Darwin's principle of evolution [TMKH96] [WF05].

GAs as a part of machine learning, are search heuristics based on the principles of biological evolution in nature [Hol75] [EHG05]. GAs cope well with the problem of local optima in error surfaces and find global optima with high probability [TMKH96]. Such algorithms have found application in the field of audio coding [Mar06], and the potential application of the GA in signal processing is unfathomably wide [TMKH96].

The performance of a particular input parameter constellation for an optimization problem is measured with an objective function. GAs aim to find a set of input parameters that produce minimal error on the function being optimized. Particular values are chosen for the entire set of genes to create individuals. Like in natural evolution, individuals with good genes are more likely to survive than individuals with bad genes and are therefore more likely to reproduce. The assumption is that if two well performing individuals exchange genes, the resulting combination of genes will be even more likely to survive. At the same time, GAs introduce spontaneous changes to the genes, akin to the phenomenon of mutation in nature. Applying these principles of Darwinistic evolution to an iterative process that spans numerous iterations of procreation, mutation and selection, GAs can learn to traverse an error surface to find a global optimum [TMKH96].

1.3 Goals

The goal of this thesis is to show that a perceptual audio codec can be automatically tuned. The feasibility of an automatic optimization process to become an effective guiding factor in expert decision making should be examined. The process should provide automation of menial tasks, like recompilation of executables and (en)coding, and should evaluate the perceptual audio quality of coded excerpts autonomously. Prior research of [PWO15] and [HZ09a] is applied to the Dolby AC-4 codec, and the thesis aims to verify that the GA is a feasible parameter estimation method in audio encoder tuning.

The evaluation of the results of the optimization process poses several challenges for the experimenter. The perceptual audio quality improvement measured by PEAQ are reported with respect to a particular set of audio files. The set of files used to validate improvements should therefore be separate in order to guarantee that the optimization has general validity and is not merely a special set of parameters performing well on a particular set of audio excerpts. Another important factor is the reproducibility of results, i.e., if an optimization for a certain set of parameters is run multiple times, the same level of improvement should be achieved each time.

A perceptual audio coding algorithm, like the Dolby AC-4 [KRW+16], opens up many possibilities to tune and optimize perceptual quality of audio. If too many parameters are

considered at once, no meaningful optimization can occur, as the algorithm is only able to explore a subset of the parameter space. Therefore, a choice must be made as to which parameters to adjust with the algorithm. The decisions were taken with the intuition and guidance of audio tuning experts who estimated the room for improvement that could be made by adjusting a certain subset of parameters.

Two different areas of optimization were the focus of the work in this thesis. The choice of transformations that should be used by the encoder to analyze time-frequency behavior, commonly referred to as block- or transform-switching, was the first investigation point of the thesis and yielded promising results. The second attempt was to optimize the behavior that control the bit reservoir of the codec.

Unlike prior research described in Sections 3.4 and 3.3, who examined a simple low-delay ADPCM-based codec, the subject of this thesis is the Dolby AC-4 codec. It is a sophisticated perceptual audio codec, with hundreds of adjustable parameters. The current state of the Dolby AC-4 codec is the result of years of research, and many of the parameters have been found empirically by audio tuning experts [KRW+16]. The aim of this thesis is therefore also to find out whether an automatic parameter optimization method can outperform experts.

Another goal of the thesis is to use a validation set, in addition to the training set of audio excerpts. When the result of an optimization algorithm is evaluated on the same set as it was trained on, the improvement for that particular set is measured. In order to prevent this overfitting on a particular set of files occurred, the results of the optimization should be cross validated on another set of excerpts. Improvement of PEAQ scores for the validation set indicates that the encoder does not only improve the perceptual audio quality for the training set, but for general classes of audio. Additionally, the parameter constellations that showed the greatest improvement measured with PEAQ, the perceptual audio quality improvement was also tested in formal listening tests as described in [Rec03] with expert listeners .

As mentioned above, comparing an original and an excerpt that has been subjected to perceptual audio coding is a process that involves several computationally intensive steps. The first challenge that affects the performance of the system, is the amount of processing power that is required to encode, decode, post-process and compare the original with the processed excerpts. For an excerpt length of twenty seconds, this action may well require time on the scale of tens of seconds, which highlights the importance of designing a massively parallel system that can process data concurrently in order to reach acceptable run time for the algorithm.

Operability, flexibility, and maintenance were the key goals in the design of the system designed in the course of this thesis. The system's purpose is to guide audio tuning experts in their decision making. These experts do not necessarily have the background in software engineering to operate an intricately complex system in order to run a few experiments. Rather than being concerned with the technical details of the system, the operator should

be able to focus on the research aspect of the project .

The evaluation of results, progress and performance plays an integral role in the research process of this work. The insight provided by graphical process analysis tools was essential to discover the relationship between changes and improvements made to the system and the resulting effects on performance and correctness of the system. In order to prepare the results of the optimization for interpretation by audio tuning experts, data visualization is indispensable. Additionally, statistical analysis was carried out to compare different objective listener tools to increase the confidence in the perceptual improvement reported by these tools.

1.4 Contributions & Overview

As a result of this thesis, a three component system, consisting of "Publisher", "Broker" and "Client" was built. The separation in three components makes the system more maintainable, because different aspects can be modified without breaking the rest of the system. The Publisher is most often updated, while the Client only needs to be updated rarely, and the Broker is static. The part of the system is responsible for implementing the machine learning and optimization algorithm, in other words the core of the intelligence of the system, is referred to as the "Publisher", described in Section 4.2.1. The Publisher generates different trial solutions to the problem. The trial solutions (individuals) are composed of different input parameters (genes). An objective function value is assigned to each individual by the system, by carrying out a distributed process.

The bulk of the computational work is done by a component referred to as the "Client", described in Section 4.2.2. The Client is a collection of different tools that realize the encoding, decoding, post-processing and objective measure capabilities of the system. The parallel nature of the system is most apparent in that, the Client is distributed on a network of computers all performing the same algorithm on different parts of the data produced by the Publisher. In order to evaluate a particular individual, a Client makes the appropriate adjustments in the audio encoder and processes the file. After decoding the resulting bit-stream, a new wave file is obtained. Before the coded file can be compared to the original, it must be time-aligned and low-pass filtered, to make the comparison with PEAQ possible. The Client measures the degradation over a set of excerpts with PEAQ, and reports back a measure for the average degradation in the set. The Publisher waits for the entire series of individuals to be evaluated before proceeding to applying genetic operations in order to generate the next set of individuals. This process is repeated until no substantial improvement is achieved in multiple successive generations. The result of one experiment is a set of optimal values for the genes of the individual.

The communication and persistence capabilities of the system are encapsulated in a component referred to as the "Broker", described in Section 4.2.3. The responsibility of the

broker is to facilitate communication between the other two components of the system.

In the system, different languages are used for the implementation. Numerous external tools are used which have to be coordinated. Scripting languages, e.g.: Python, offer a great advantage for constructing a distributed system, because they allow the operator to move quickly from idea to prototype, and enable the operator to easily modify the system. At the same time, computationally demanding parts of the process can be delegated to tools written in languages with higher computational throughput, such as C. The Hyper Text Transport Protocol (HTTP) is used in the system to establish communication between the distributed parts. The communication of the parts within the system follows a simple, fixed protocol that defines an interface for the different components. The Broker guarantees the robustness of the optimization process with regards to structural and operational failure, e.g: loss of electrical power, failures of the network, and recovery from such failures. At the same time, it persists data to a database system for later retrieval and evaluation.

A great advantage arising from the decoupled design into three parts is that the different parts can be modified, updated and replaced independently of each other as long as the communication interface is respected. In particular the concrete choice of a machine learning algorithm for the Publisher is not predetermined, but can be substituted at any time with ease. Another beneficial consequence is that the system is robust to failure of individual parts. Failure of random elements of a distributed computer cluster is very common due to various reasons like maintenance of the machines, network communication errors or power failures. With the persistence and recovery capabilities of the system, such failures do not have a large impact on the overall progress of optimization; rather the effects are limited in time and gravity.

Several experiments were conducted to investigate, whether a sophisticated perceptual audio codec can be tuned by a GA. In Section 5.1 the experiments and results concerning window-switching are described. The modification of bit reservoir control is described in Section 5.2.

The first result that can be interpreted by the experimenter, is the score reported by PEAQ. PEAQ forms the first indicator for the experimenter for the perceptual audio quality of files. The results obtained with PEAQ are described in Sections 5.1.1, 5.1.3 and 5.2.1. The ground truth for perceptual quality improvement however, is the confirmation by expert listeners in formal listening tests. Only when good correlation is found between the quality improvement reported by PEAQ and the results of formal listening tests, it can be assumed that the objective listener scores bear relevance to perceptual quality improvement. In Sections 5.1.1, 5.1.2 and 5.1.3, the results for listening test experiments are described.

As a result of the thesis, parameter constellations for the AC-4 encoder were found, which in some cases show clear improvement in perceptual audio quality over the default values of the encoder. The quality of the optimization based on PEAQ scores of the training set of audio files is cross validated, by running the evaluation on a validation set of au-

dio files, the ATSC3.0 and the MPEG set described in Section 4.3. PEAQ shows clear improvement of audio quality for some excerpts and the improvements were confirmed in listening tests. Especially the transform-switching behavior showed room for improvement through optimization. From the experiments conducted, it is seen that in many cases the perceptual audio quality estimate of PEAQ correlates well with the results of subjective listening tests. The estimates of PEAQ appear to be adequate enough for the purposes of audio encoder tuning.

Chapter 2

Theoretical Foundation

2.1 Psychoacoustics

Perceptual coding is largely based on the concept of psychoacoustics [Fle40][ZRRE65][Sch70] [Hel72][Gre61]. Psychoacoustics characterizes human auditory perception and the time-frequency analysis capabilities of the inner ear. Superior compression is achieved by perceptual audio codecs, by exploiting the phenomenon that irrelevant information in an audio signal is not detectable, even by a well-trained or sensitive listener.

Irrelevant information is recognized when analyzing a signal by abiding by several psychoacoustic principles. Combining these psychoacoustic notions with basic properties of signal quantization has also led to the theory of perceptual entropy [Joh88], a quantitative estimate of the fundamental limit of transparent audio signal compression. The following sections give a brief overview over the fundamentals of psychoacoustic principles, including absolute hearing thresholds, critical band frequency analysis, simultaneous masking and the spread of masking along the basilar membrane [SPA06].

Absolute Hearing Thresholds

The absolute threshold of hearing characterizes the amount of energy needed in a pure tone such that it can be detected by a listener in a noiseless environment. The absolute threshold is typically expressed in terms of dB Sound Pressure Level (SPL). In 1940 Fletcher [Fle40] reported test results for a group of listeners that were generated in a National Institutes of Health (NIH) study of typical American hearing acuity and quantified the frequency dependence of the absolute hearing threshold. [SPA06]. For a young subject with good hearing in a quiet environment the threshold is well approximated by the non linear function in Equation 2.1 [SPA06].

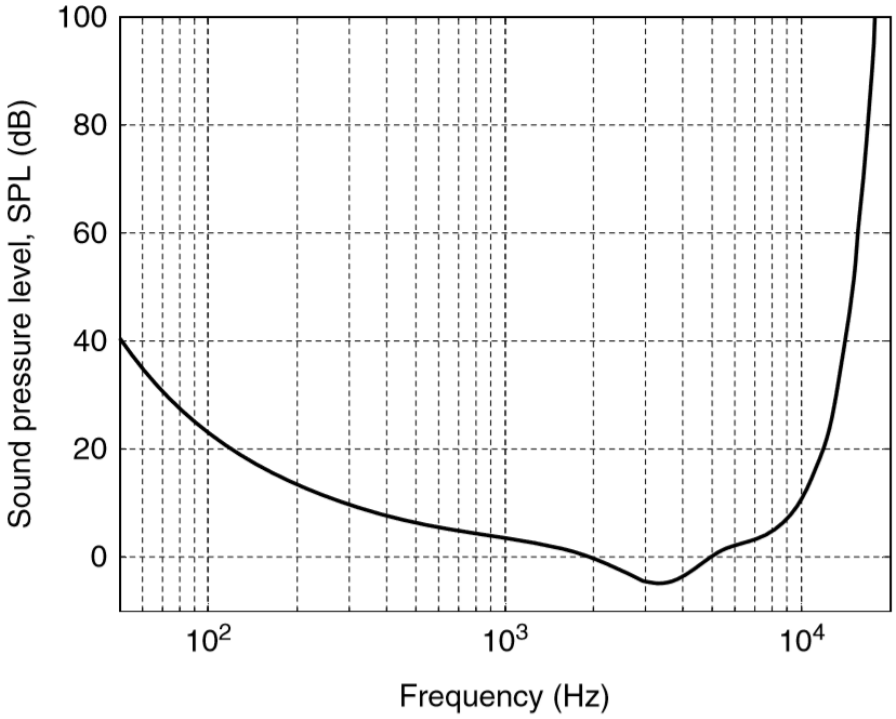

Figure 2.1: Absolute hearing threshold in quiet, as described in [SPN2006].

$$T_q(f) = 3.64(f/1000)^{0.8} - 6.5e^{-0.6(f/10003.3)^2} + 10^{-3}(f/1000)^4(dbSPL) \tag{2.1}$$

T_q is the hearing threshold in a quiet environment for a listener in a quiet environment given in db SPL as a function of frequency f. T_q is depicted in Figure 2.1. In signal processing and coding, Equation 2.1 could be simplistically viewed as the maximum allowable energy level for noise as a result of perceptual coding in the frequency domain, so distortions can not be noticed by listeners.

Critical Bands

Taking into consideration the absolute threshold of hearing is the first step to exploiting the irrelevant information in a signal to achieve superior signal compression compared to entropy coding. It can be used to shape the coding distortion spectrum but it is of limited use in perceptual audio coding when used in isolation. Other techniques build on the concept of irrelevant information in more complex ways and improve signal compression further.

Equation 2.1 describes the absolute hearing thresholds, irrespective of time and other signals present. By examining the relationships between different stimuli present at a given time and the effect on the audibility of noise, it is possible to adjust the detection threshold for spectrally complex quantization noise. Because of the time-varying nature of

stimuli, the detection threshold is also a function of time and the input signal. In order to adjust the threshold appropriately it is necessary to examine how the inner ear performs spectral analysis of sound [SPA06].

Inside the cochlea, a frequency to place transformation takes place along the basilar membrane. An incoming sound wave or acoustic stimulus excites the eardrum which passes on the mechanical vibrations to the cochlea. Inside the fluid filled structure containing the basilar membrane, waves excite resonance at frequency dependent membrane locations. Different neural receptors are activated for different locations and therefore different frequencies in an incoming signal [SPA06].

When two tones excite receptors in the same physical location of the cochlea, they interfere with each other. A critical band is the range of audio frequencies, within which a second tone will influence the perception of the first tone by auditory masking.

Masking

When two stimuli overlap in time, it is possible that one stimulus becomes inaudible. This phenomenon is referred to as masking. Anytime the human auditory complex is confronted with more than one sound at the same time, masking of sounds can occur. Dependent of the shape of the magnitude spectrum of the masker and maskee, certain spectral portions of the masked sound (maskee) may be rendered inaudible by the masking stimulus (masker) [SPA06]. Biologically, an explanation of the phenomenon is that a strong stimulus can excite the basilar membrane at a critical band location sufficiently, to prevent detection of the weaker stimulus.

Spreading

Simultaneous masking effects are not band-limited to a single critical band. The presence of a stimulus in one critical band also leads to predicTable interband masking effects. The change of detection thresholds of stimuli in neighboring critical bands is called the spread of masking [SPA06]. It is often modeled in perceptual coding algorithms as an approximately triangular spreading function with slopes of approximately +25 dB/Bark on the lower frequency side and -10 dB/Bark on the higher frequency side.

2.2 Time-Frequency Analysis and Transform Switching

Block based coding schemes like the Dolby AC-4 codec [KRW+16] tend to be plagued by pre-echoes [SPA06]. Several strategies have been proposed and successfully applied, in the

effort to mitigate this effect. Two of the most widespread techniques are the bit reservoir and window switching.

Transform Switching

Analysis of an audio signal is performed in the time-frequency domain. A tradeoff must be made by perceptual audio codecs, between good time-resolution and good frequency-resolution. The shape of the masking threshold in the time-frequency domain depends

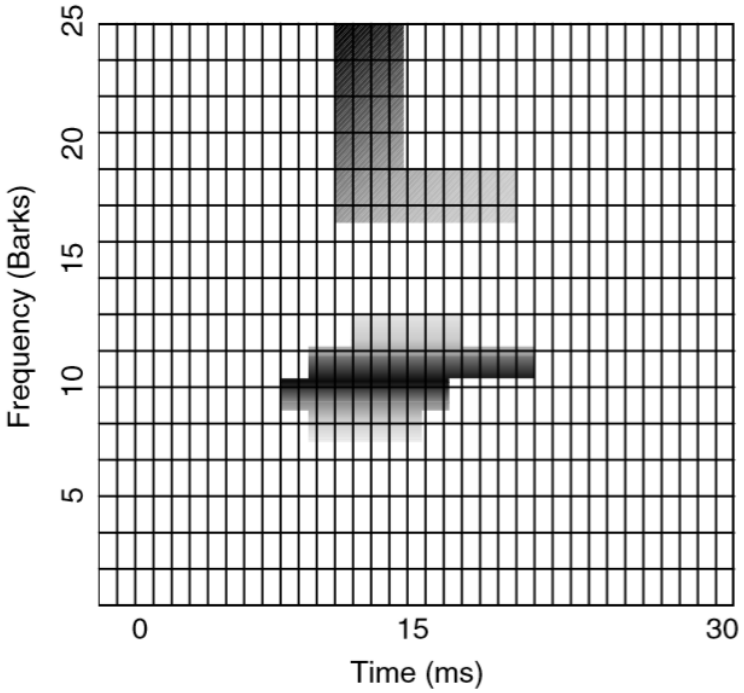

Figure 2.2: Masking thresholds in the time-frequency plane for castanets (after [PJ95]).

heavily of the characteristics of the signal. Signals with sharp attacks and transients, like castanets, mask a broad frequency range in a short window of time, as depicted in Figure 2.2. Tonal signals, like trumpets or piccolos, on the other hand, mask a narrow frequency range over an extended period, as depicted in Figure 2.3. Choosing adequate transform lengths that match the characteristics of the input signal is an important factor for the efficiency of perceptual audio coding. An inappropriate choice of transform length can often result in pre-echos and other perceptible artifacts, or low coding gain and therefore high bitrates [SPA06].

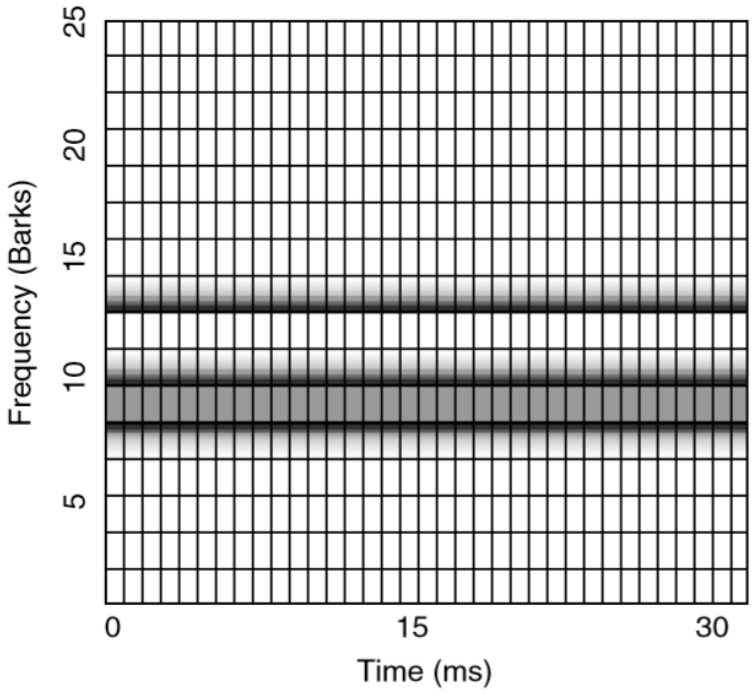

Figure 2.3: Masking thresholds in the time-frequency plane for piccolo (after [PJ95]).

Bitreservoir Control

Another technique to compensate pre-echo, is the bit reservoir. It is a technique that satisfies the greater bit demand associated with transients, by storing surplus bits during periods of low demand. Even though most coding algorithms use a fixed bit rate, the instantaneous bit rates are time varying. The reservoir is based on the idea of storing bits that are unneeded during passages which require few bits, and spend bits from the reservoir on passages with peak demand. The instantaneous bit rates vary, but result in an fixed average bit rate [SPA06].

Advanced Spectral Extension

Advanced Spectral Extension (A-SPX), is a technique based on spectral band replication (SBR) [DLKK02]. SBR is a strategy to improve an audio or speech codec, especially at low bit rates. It is based on harmonic redundancy in the frequency domain. The codec itself transmits the lower band of the spectrum, while SBR replicates higher frequency content by transposing up harmonics from the lower and mid-frequencies at the decoder. A-SPX performs high frequency reconstruction similar to e.g. MPEG SBR [DLKK02], or DD+ SPX [ACD$^+$04]. The low-band signal is waveform coded and used to reproduce a high-band signal, which is adjusted afterwards utilizing the A-SPX side-information, to equal the properties of the original high-band signal. In comparison to former versions, however,

it permits very flexible interleaving of wave-form coded elements, with the parametrically coded passages [KRW+16].

2.3 Principal Component Analysis

Principal Component Analysis (PCA), is the common name for a strategy using intricate underlying mathematical principles to transform a number of potentially correlated variables into a smaller number of variables, which are called principal components [BS14]. PCA has been referred to as one of the biggest achievements of applied linear algebra, and it is perhaps the most common first step in the analysis of large datasets. Some of the other common applications include: de-noising signals, parameter choice optimization [WEG87], and data compression.

A data set containing many variables can often be interpreted in a space of reduced dimensionality. PCA uses mathematical properties of a dataset to project it onto a lower-dimensional subspace. This way PCA applies a vector space transformation to reduce the dimensionality of data sets. The vectors of the basis of the new vector space are called principal components. In the reduced vector space the user can spot patterns and dependencies in a dataset more easily, than would have been possible without the PCA [BS14].

2.4 Perceptual Evaluation of Audio Quality (PEAQ)

The subjective listening experience of humans listening to excerpts that have been subjected to perceptual coding is typically assessed with formal listening tests using subjective criteria. Because such tests are frequently impractical to set up and expensive a method was created that models the sensory and cognitive processes underlying subjective ratings. The PEAQ algorithm is an algorithm certified by the ITU as the standard for objective measurement of perceived audio quality. The algorithm compares an perceptually coded excerpt to its uncoded pendant. Based on a series of model output variables derived from psychoacoustic principles PEAQ returns a real value between 0 and -5 indicating the perceptibility of distortions and artifacts in the coded excerpt. An ODG value of 0 indicates that no perceptible distortions are present in the excerpt, while lower values signify the presence of increasingly annoying distortions as seen in Table 2.4.

PEAQ includes two ear models, one based on a filter bank and one based on the fast Fourier transform (FFT). From these models, a masked threshold is estimated on a frame-by-frame basis. The model output values are based partly on the concept of masked thresholds and partly on a comparison of internal representations. In addition, it also compares linear spectra not processed by an ear model and yields output values from the comparison. The model outputs the partial loudness of non-linear distortions, the partial loudness of linear

ODG Value	Presence of Distortions
0	Imperceptible
-1	Perceptible, but not annoying
-2	Slightly annoying
-3	Annoying
-4	Very annoying

Table 2.1: Meaning of ODG values in terms of subjective listening experience.

distortions, a noise-to-mask ratio, measures of alterations of temporal envelopes, a measure of harmonics in the difference signal, a probability of artifact detection, and the percentage of excerpt frames containing audible distortions [TTB+00].

An artificial neural network with one hidden layer [RHW85] is used to map selected output variables to a single quality indicator.

PEAQ as recommended and verified per ITU recommendation as the standard for subjective perceptual quality assessment [Rec03], is available in two versions: PEAQ Advanced and PEAQ Basic.

PEAQ Basic

In the basic version of PEAQ, only the FFT-based ear model is used. Both the concept of comparing internal representation and the concept of masked threshold are applied to compare two excerpts. The FFT-based ear model has relatively poor temporal resolution, but the restrictions arising due to this fact are partly compensated by a greater number of model output variables and a better spectral resolution than the Advanced version [TTB+00]. The model-output-variables derived from the ear model measure the loudness of distortions, the amount of linear distortions, the relative frequency of audible artifacts, changes in the temporal envelope, a noise-to-mask ratio, noise detection probability, and the harmonic structure in the difference signal [TTB+00].

PEAQ Advanced

The advanced version of PEAQ utilizes both the FFT-based ear model as well as the filter bank based ear model. The concept of comparing internal representations is applied using the filter bank based ear model, while the masked threshold concept is applied using the FFT based ear model. The model-output-variables derived from the filter bank model determine the loudness of nonlinear distortions, the amount of linear distortions, and disruptions of the temporal envelope. The variables based on the FFT include a noise-to-mask ratio and a measure of harmonic structure in the difference signal [TTB+00]. PEAQ

Advanced and PEAQ Basic scores often differ for the same signal, since the advanced version analyses the signal differently.

2.5 Optimization Algorithms

Mathematical optimization in computer science, economics or management science, is the process of determining the best element from a set of possible alternatives, with regard to some criterion. Often it is the act of minimizing or maximizing a real function by systematically selecting input values from within an allowed range, and computing the function value. In broader terms, optimization also includes finding the best available values of some objective function for a set or constraints or a defined domain [WD16].

There are traditionally two classes of optimization techniques used: calculus-based algorithms and enumerative algorithms. Enumerative methods are based on the principle of potentially discovering all points in the search space over time, and thus finding the optimum value [WD16]. If some objective function that is the subject of optimization is characterized by a differentiable error surface, calculus-based optimization techniques use gradient-directed searching mechanisms to solve the optimization problem. Unfortunately local optima are often obtained for non-convex optimization problems.

In the domain of signal processing, objective functions with many local optima are common, since the signal can be noisy, fuzzy, vague, and discontinuous [TMKH96]. Some major enumerative techniques, like dynamic programming, have the capability of dealing with the local optima problem. A trade-off must be made, however, between its simplicity and robustness and its high computational complexity must be made. Enumerative approaches may also break down on complex problems of moderate size - a situation that is widely known as the “curse of dimensionality” [TMKH96].

2.5.1 Gradient Descent

Gradient descent is a first-order optimization algorithm. To find a local minimum of a function f using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point.

Calculation of the Gradient

The gradient ∇X is the sum of partial differentials in every dimension at some current point X as described in Equation 2.2. Accumulating the partial differentials over all dimensions

x_i produces the gradient in X.

$$\nabla X = \sum_{n=1}^{i} \frac{\partial f}{\partial x_n} \tag{2.2}$$

If the underlying function for the gradient descent method is non-smooth and/or non contiguous, a discrete approximation, given in Equation 2.3 may be used to calculate a discrete partial differentials for some point [BKS08].

$$\frac{\partial f}{\partial x_1} = f(\begin{bmatrix} x_1 + \epsilon \\ x_2 \\ \dots \\ x_i \end{bmatrix}) - f(\begin{bmatrix} x_1 - \epsilon \\ x_2 \\ \dots \\ x_i \end{bmatrix}) \tag{2.3}$$

Equation 2.3 approximates the partial differential in dimension x_1 for some small value ϵ in a non-contiguous objective function f.

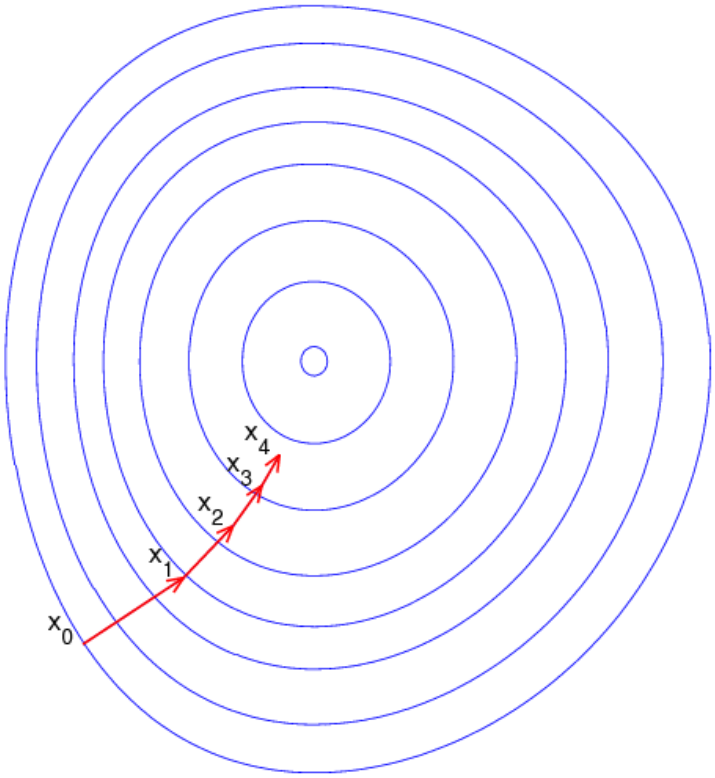

Figure 2.4: A gradient descent in two dimensions.

Algorithm

When the gradient or the approximated gradient is calculated using Equation 2.2 for some iteration $t \in 1, 2, 3, 4, ...t_{max}$ the next point X_{t+1} in the gradient descent is calculated according to the current point X_t and its gradient ∇X_t with Equation 2.4. α is a small value also called step size, which can be adjusted to make larger or smaller jumps by the experimenter.

$$X_{t+1} = X_t + \alpha * \nabla X_t \tag{2.4}$$

This process is repeated until a fixed amount of iterations t_{max} has been reached or no improvement is achieved in several successive iterations. This process is visualized in Figure 2.4.

2.5.2 Evolutionary Algorithms

Evolutionary algorithms (EAs) are heuristics methods relying on statistic search, that apply the principles of natural biological evolution and/or the social dynamics of species. For large scale optimization problems, traditional mathematical optimization approaches may fail. Evolutionary algorithms have been developed to arrive at near-optimum solutions to such optimization problems efficiently [EHG05].

2.5.3 Genetic Algorithms

In 1975, Holland [Hol75] introduced a new approach to optimization. The new approach utilizes a process that mimics evolution as encountered in nature and described by Charles Darwin. This approach is known as the Genetic Algorithm (GA) [Hol75]. GAs function in a similar manner to guided random techniques: simulated annealing [KV$^+$83], evolutionary strategies [Sch81], and evolutionary programming [FOW66].

The GA offers great adaptability and robustness that are unique for signal processing. Additionally its concept and implementation are both relatively simplistic for an optimization technique. It can be used on real-world applications as an optimization tool for designing AI-hybrid systems [TMKH96].

Genetic Algorithm Cycle

GAs iterate through a cycle inspired by the laws of evolution, consisting of the steps of Selection, Genetic Operation and Replacement. Figure 2.5 shows a typical Genetic Algorithm cycle. In a first step, an initial population is generated randomly. Each individual in the population encodes a possible solution to the problem. The fitness values of the initial population are determined by calculating the value of the objective function. The

individuals of this population form the basis for the selection process according to their fitness and the subsequent application of genetic operations. After the application of the

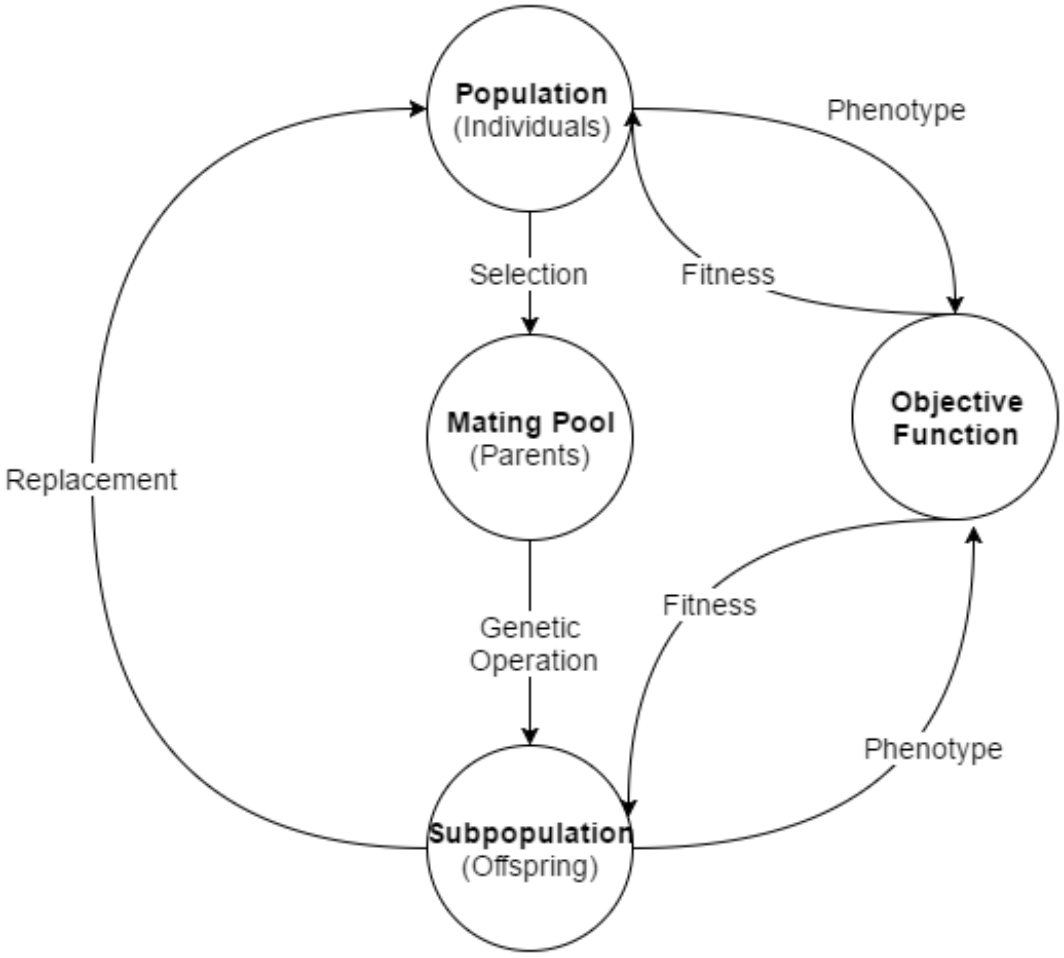

Figure 2.5: Genetic Algorithm Cycle as described in [TMKH96].

genetic operations, the objective function value of the resulting new individuals is calculated in a similar fashion in order to retrieve the fitness of the offspring. According to a certain replacement strategy the individuals of the initial population are partly replaced by their offspring so the size of the population remains constant.

The fundamental postulate of GAs is that by preferring individuals with higher fitness values during replacement and/or selection, the mean fitness of the population will increase with successive iterations of the GA cycle. The cycle is repeated until the fitness of the best individual of a generation, and therefore the quality of the solution of the problem, reaches a desired level. Alternatively a pre-set number of generations can be run [TMKH96].

Data Representation and Encoding Scheme

In order to apply Genetic principles to real-world applications, the input data and possible solutions need to be represented in the genetic domain and encoded appropriately. To increase the performance of the GA, a individual representation that contains problem-specific information is favorable. As described above, the GA produces a multiset of individuals over successive generations. Generally, each individual $x_i(i = 1, 2, ...)$ symbolizes a solution attempt to the optimization problem and is made up of a series of parameters or variables. The parameters making up an individual are referred to as genes [TMKH96].

The representation of the individual genes can be chosen freely and may be chosen to be of binary, real or more exotic forms such as LISP-S-expressions [Koz91]. Traditionally the encoding as a bit-string or using grey-coding have been considered to be the most classic encoding schemes. However, for some optimization problems these representations pose difficulties and create unnatural obstacles to the success of the optimization. Therefore also real number representations have been successfully employed to represent genetic information [JM91].

Genetic Operations

The two major genetic operations inspired by natural evolution are Mating or Crossover and the phenomenon of Mutation.

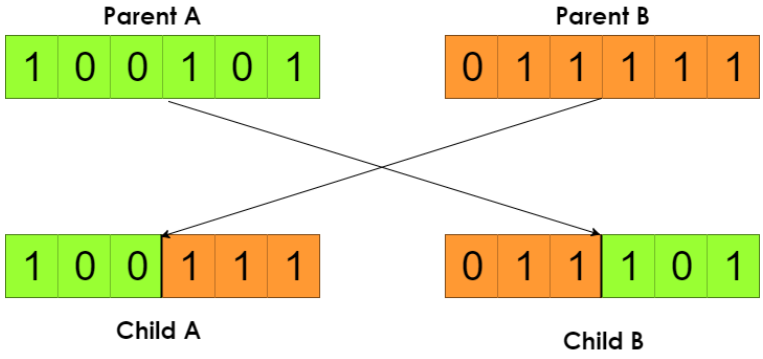

Figure 2.6: Single point crossover operation.

Crossover is an operation that takes two parent individuals and creates offspring by taking some genes from one parent and the rest from the other parent individual. The resulting offspring therefore contain a mix of genes from either parent. In the GA cycle, the rate at which crossover or mating occurs is adjusted with a probability term CXPB (Crossover Probability). GA practitioners often view the crossover operator to be the key component that differentiates the GA from all other optimization methods [TMKH96]. Different possible variations of the crossover operation have been proposed. The most basic variation is

a single-point crossover. A crossover point is chosen at random and the portions of genes of the parent individuals are mixed accordingly as depicted in Figure 2.6.

A more sophisticated version of the crossover operation is called multipoint crossover. Instead of choosing only one crossover point, multiple points are chosen randomly. This is illustrated in Figure 2.7. Both singlepoint and multipoint crossover define points at which the parent individuals can be split. By generalizing multipoint crossover to uniform crossover, the operation is generalized to make every locus a valid crossover point. In uniform crossover a binary string with the same length as the parent individuals indicates from which parent individual the associated gene should be chosen for the descendant. This procedure can easily be implemented by choosing a random binary string.

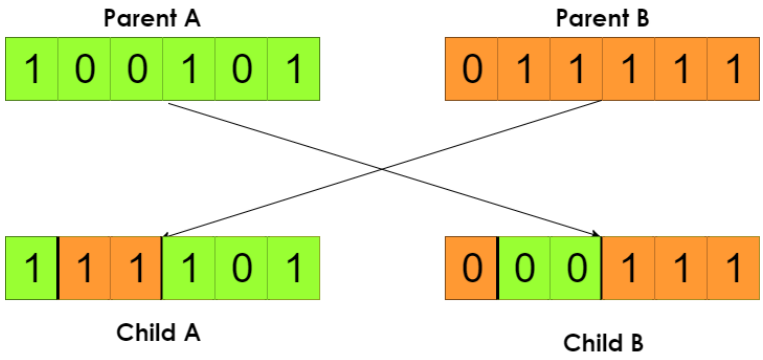

Figure 2.7: Multipoint crossover operation.

Mutation is a genetic operation that modifies individuals in a random fashion, thereby introducing new genetic material to the population that was not present before. It is useful to increase the variety of the genetic material, because even though just as in nature a random mutation often results in defective individuals, the biggest steps in evolution are often only achieved by mutation (e.g: lungs, thumbs,...). The rate at which mutation occurs is adjusted with a probability term MUTPB (Mutation Probability).

Mutation can be carried out on an individual by altering its genes randomly, either by randomly toggling bits in them (for a bit string representation), or by adding random real numbers, sampled either from a uniform probability distribution, or a gaussian probability distribution.

Replacement and Selection

Through the process of mating, the GA generates a new set of individuals that did not exist in the population so far. Inspired by nature, two processes, “replacement” and “selection”, influence the evolution to keep population size constant. This is desired for computational complexity reasons, but also to facilitate the eventual convergence of the algorithm. Only

if the less fit, and therefore less performing, members in the population are taken out of the gene-pool, the fitness of individuals will increase over time.

Replacement strategies generally favor the best performing individuals in some way, this is called the elitism of the algorithm. Common replacement strategies include best-n, single and (multi-stage) tournaments, roulette selection and random selection. In a best-n replacement, the best individuals of a population and its descendants form the next generations, and the remaining weaker individuals are discarded. In a tournament selection, groups of individuals are chosen at random and the best individual from that group survives. The process is repeated until enough survivors are chosen so that the population size remains constant. In contrast to best-n selection, in tournament selection, an individual can be chosen more than once. Due to the random grouping of individuals in tournament selection, no individual is guaranteed to pass on to the next generation, and very good individuals merely have a very high chance of surviving [HNG94]. Roulette selection is similar in this regard, because individuals are chosen as members of a new generation based on a probabilistic selection scheme. The probability that a particular individual will be chosen as a survivor is proportional to its fitness [GD91].

Parallelism

GAs, especially when applied to signal processing problems, may be parallelized in a number of methods to increase the computation speed. The parallelization can be classified as one of several options:

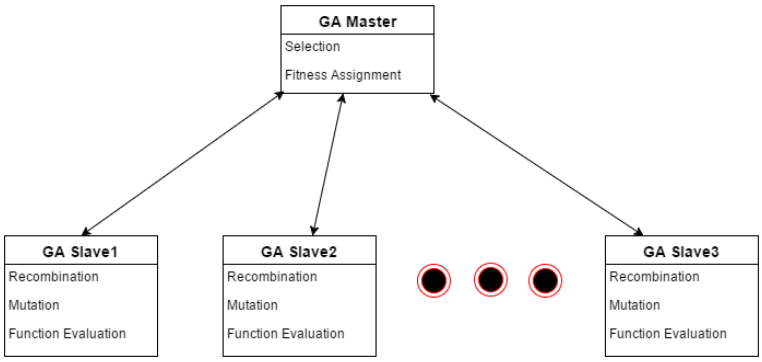

Figure 2.8: Global Parallelization as described in [TMKH96].

Global parallelization (Figure 2.8) treats the entire population as a single breeding, and waits for the fitness of all individuals of a generation to be evaluated. Other parallelization schemes, like migration or diffusion, split the global population into distinct breeding groups, that function like geographically separated breedings in nature, which exchange genetic material only on a few occasions. The advantage of diffusion and migration based

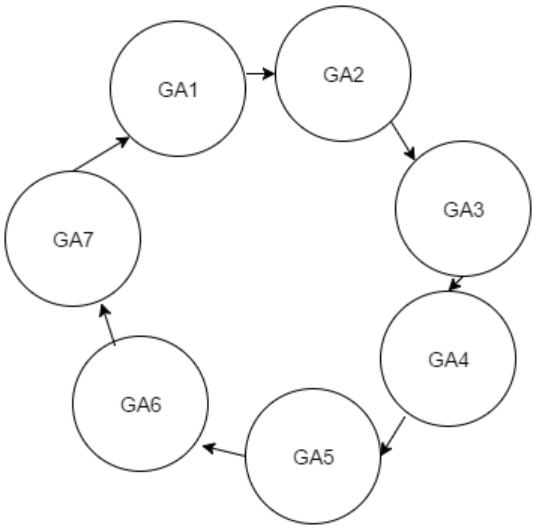

Figure 2.9: Ring Migration Parallelization as described in [TMKH96].

parallelization (Figure 2.9) is, that the different sub-breedings need not be kept synchronized at all points of time, and can therefore be operated independently of each other until genetic material is exchanged [TMKH96].

Chapter 3

Prior Research

3.1 Audio Coding using a Genetic Algorithm

In [Mar06], a group of researchers presented an approach where a psychoacoustic metric compared the output signal with the input signal to drive the bit allocation of the encoder, and uses a genetic algorithm in the feedback process. In their paper, they compared the audio quality of the presented genetic coder, with that of leading conventional audio coders. The aim of the work was, to assess how far Layer II coding can be improved, and whether any further progress can be made with conventional coding.

Their conclusion was that any improvements in Layer II coding will be small, yet their results did show that for some test material the genetic coder did show improvements. They postulated that if it is possible to find what caused the perceived improvements these might help in improving the design of a conventional encoder. They noted that these improvements are unlikely to increase the quality of all audio, but just some types of audio.

3.2 Efficient Perceptual Tuning of Hearing Aids with Genetic Algorithms

[DWVTR04] describes a system for integrating a GA with perceptual feedback to perform an efficient search in a perceptual space.The main system components are an efficient method for estimating perceptual rank order and genetic operators that take advantage of the types of parameters found in certain classes of audio processing systems.

In their paper, preference judgments of test subjects are used. The application to subjectively fitting a portable hearing aid based solely on binary feedback is discussed. An experiment was conducted using eight normal and eight hearing impaired subjects. Three

parameters were varied to control cancellation of acoustic feedback. The GA worked well for fitting this system, as indicated by both objective and subjective measures. In addition, users had greatly differing preferences for feedback cancellation parameters and these preferences did not change much when subjects were retested.

3.3 Automatic Parameter Optimization for a Perceptual Audio Codec

[HZ09a] is the first to describe the automatic parameter optimization of a perceptual codec. They report, that for a perceptual audio codec it is necessary to choose various parameters optimally to achieve good perceptual quality. They state that this optimum cannot be derived analytically and an educated guess of the designer may be far from optimal. Their paper claims that automatic parameter optimization could be fruitfully applied in the development of audio codecs in general. Their group proposes an approach in which an global parameter optimization method is combined with a method for local search. In their paper they examine the effectiveness of a global parameter search method called simulated annealing. While their initial optimization algorithm is the global search method , they explore the application of a local search method to further improve perceptual quality. In their paper they avoid the estimation of the gradient for local search and apply Rosenbrock's method to descend into a local optimum after the execution of the simulated annealing.

3.4 Adaptive Pre- and Post-Filtering for a Subband ADPCM-based Low Delay Audio Codec

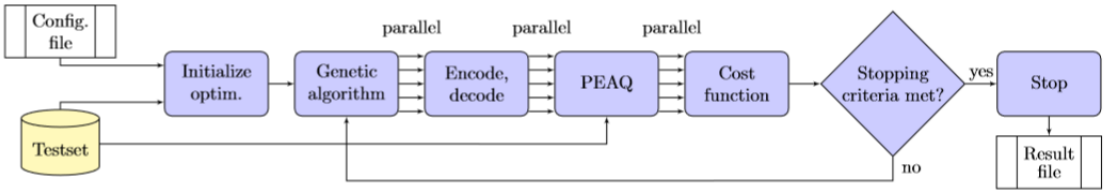

Figure 3.1: Description of the system in [PWO15].

Unlike [HZ09a], [PWO15] employ a GA as global parameter optimization method. The work is based on [HZ09a] and uses the same cost function to estimate the fitness of different solutions. They describe a system, in which a low delay subband ADPCM-based audio codec is subjected to global parameter optimization. They claim that the subband

coding of their work as well as the adaptive pre- and post-filtering include several partially interacting parameters that are hard to adjust manually. They introduce a framework for their perceptually controlled global optimization. They also describe a practical implementation of a system to carry out the optimization, depicted in Figure 3.1. Their experiments and their results show that the global optimization enables finding a meaningful operating point.

The work done by [HZ09a] and [PWO15] is the most relevant for the work of this thesis.

Chapter 4

Implementation

4.1 Optimization Process

The core process used to tune the encoder can be divided into two phases. In the first phase, a generation of new individuals with unknown fitness values is generated by the GA. This part of the algorithm is referred to as the "forward path". In the second phase, the resulting individuals are published to a cluster of machines, which subject a test set of excerpts to perceptual coding, with the parameters contained in the individuals and return a fitness value back to the GA. This path is referred to as the "backward path". The core system is depicted in Figure 4.1.

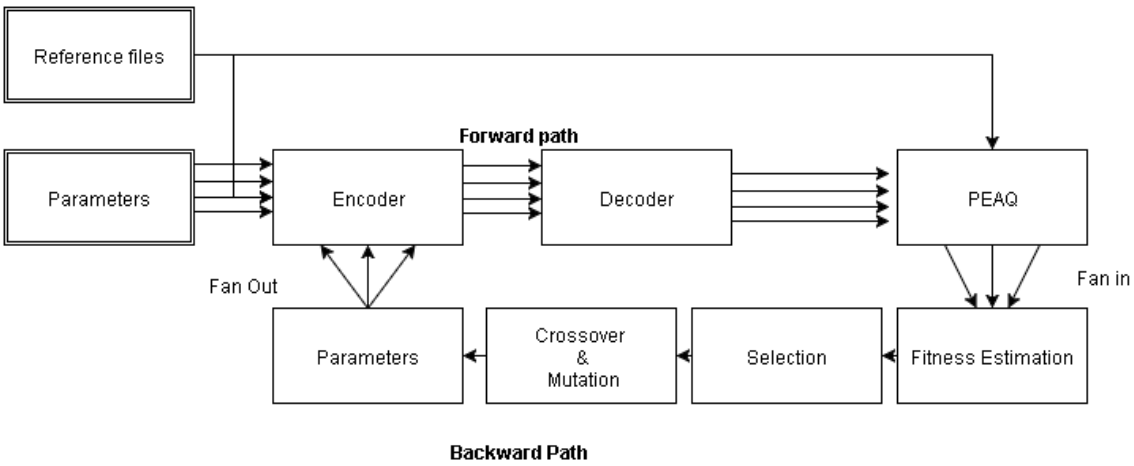

Figure 4.1: The core optimization process without local search.

The aim of the algorithm applied in the work of this thesis is to optimize the perceptual quality of audio, after it has been subjected to perceptual coding. In the process of encoding and decoding excerpts with the Dolby AC-4 algorithm, different parameters in the

algorithm are changed and the effects of parameter changes on audio quality are examined. Traditionally in audio encoder tuning the degradation of perceptual audio quality through the process of perceptual coding is estimated from formal listening tests [TTB+00]. The application of GAs to the problem of audio encoder tuning requires the evaluation of thousands of parameter combinations and conducting listening tests for all the trials is impractical.

4.1.1 Objective Measurement Tool

Instead of relying on listening tests, the PEAQ objective measurement tool is used by the algorithm to assess the perceptual audio quality. As a result of the PEAQ comparison, a real number between 0 and -5 is returned, which reflects the perceptual degradation of the file. This value is called Objective Difference Grade (ODG). To assess the degradation of perceptual quality in the general case, a set of audio excerpts, each containing different types of audio signals, is subjected to perceptual coding and evaluated by PEAQ. The optimization process is the act of minimizing the aggregate ODG of an evaluation set of audio excerpts. Let in the following, $CODEC(x_1, x_2, ..., x_i)$ denote the act of subjecting an audio excerpt to perceptual encoding using the parameters $x_1, x_2, ..., x_i$. Furthermore, let $PEAQ(File)$ denote the act of determining the ODG value of an excerpt that has been subjected to perceptual coding.

4.1.2 Objective Function

The GA produces individuals with different parameters (genes) $x_1, x_2, ..., x_i$ as trial solutions to the optimization problem. To proceed with the evolutionary process, each individuals must have an associated fitness value. This value must be obtained by aggregating the ODG values over the entire set of audio files, because one ODG value is obtained per excerpt.

$$ODG_{excerpt}(x_1, x_2, ..., x_i) = PEAQ_{excerpt}(CODEC_{excerpt}(x_1, x_2, ..., x_i)) \qquad (4.1)$$

Equation 4.1 describes how an ODG value is obtained, in case all parameters $x_1, x_2, ..., x_i$ represent have valid values for a particular excerpt. Values are considered valued if they do not crash the encoder. To achieve an effective improvement in perceptual audio quality, the set of ODG values for a particular parameter configuration must compare favorably to the ODG values achieved with the default settings of the encoder (i.e. values obtained after years of research). Comparing two sets of ODG values in order to determine, whether one is better than the other, can be formalized by calculating the distance between two points in the space of ODG values. The dimensionality of this space is equal to the number of excerpts used to evaluate the perceptual audio quality with PEAQ.

The distance between two points in this space is the key criterion to define improvement or degradation of perceptual audio quality. Different distance measures or norms can be used to influence the decision making of the algorithm. The choice of distance measure affects the algorithms ability to generalize or specialize. Different norms have been investigated and will be presented in the following. A visualization of different norms is given in Figure 4.2.

Manhattan Distance

If the distance is calculated using the l_1 norm, also called Manhattan Distance or Taxicab Geometry [Kra12], then for N different excerpts the aggregate score is calculated in Equation 4.2. Effectively it is the sum of degradation values of all excerpts.

$$ODG_{aggregate} = \sum_{excerpt \in TestSet} ODG_{excerpt}(x_1, x_2, ..., x_i) \tag{4.2}$$

Calculating the aggregate objective function value with (4.2) implies that trading an improvement in one excerpt $excerpt_1$, for an equal degradation of perceptual quality in another excerpt $excerpt_2$ does not change $ODG_{aggregate}$. If the optimization is run with this norm, then often the algorithm would find a configuration of parameters, that would match a particular type of excerpt (e.g.: many strongly transient signals, harmonic excerpts, speech items). The GA will then optimize for this specific kind of audio signal, and accept degradation in other types. As a result, a perceptual quality improvement would be reported for the set of reference files, when in reality some the quality of some excerpts had deteriorated.

In audio encoder tuning, sometimes the experimenter is interested in achieving exactly this. By analyzing the signal for particular characteristics, sometimes a perceptual audio codec is able to determine, that the content being processed is from a particular family of signals, e.g.: speech, and it will choose certain internal settings that are more advantageous for processing speech, than other kind of audio.

In the scope of the thesis however, the focus was put on achieving an improvement in perceptual audio quality for the general case. This implies that degradation of even one excerpt from the test set is sufficient to question the quality of a potential solution. Consequently, there should ideally be no or only insignificant degradation in any excerpt, and some items should improve audibly. The l_1 norm is poorly suited to achieve this desired effect [HZ09a].

Maximum Norm

A distance measure better suited to judge general improvement is given by the l_max, that is the the maximum value of all ODG scores of a set, indicated by PEAQ:

$$ODG_{aggregate} = max_{excerpt \in TestSet}(ODG_{excerpt}(x_1, x_2, ..., x_i)) \tag{4.3}$$

Utilizing Equation 4.3 to aggregate ODG scores, is equivalent to optimizing the perceptual quality of the excerpt with the lowest ODG only. This is often unsuitable, because minor degradation may not cause audible artifacts, and can therefore safely be ignored by the algorithm in favor of perceptual audio quality improvement of other items [HZ09a].

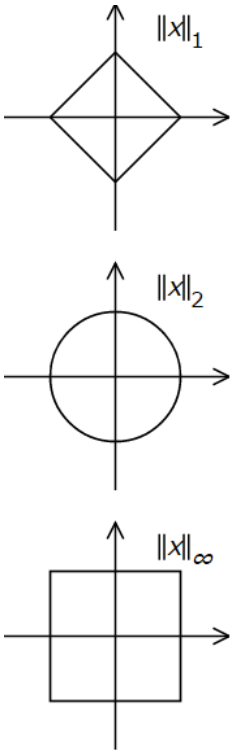

Figure 4.2: Comparison of the unit circle in l_1, l_2, and maximum norm.

Euclidian Norm

A useful distance measure to balance improvements between different items in a general optimization algorithm is the l_2 norm, also called euclidian norm. An aggregate objective function value over a set of excerpts using the l_2 norm can be calculated using Equation 4.4:

$$ODG_{aggregate} = \sqrt{\sum_{excerpt \in TestSet} ODG_{excerpt}(x_1, x_2, ..., x_i)^2} \tag{4.4}$$

In comparison to the Manhattan distance [Kra12], the algorithm is much less likely to trade an improvement in some items, for an equal degradation in other items. At the same time, minor degradation of items are tolerated in favor of large improvement of other excerpts. The reason behind is, that the individual scores $ODG_{excerpt}$ are squared before aggregation. Degradation in some items means that their PEAQ score will improve, and because of the squaring, the effect of degraded items on the aggregate score will be superproportional. The improvement of an item on the other hand, will lower its PEAQ score, the effect of which on $ODG_{aggregate}$ will be attenuated by squaring.

Other norms

The effect of introducing a non-linearity to increase the algorithms ability to generalize, can be amplified by using higher order polynomials, in order to calculate $ODG_{aggregate}$.

[HZ09a] and [PWO15] explored the choice of norm to be used for aggregating individual ODG values [HZ09a]. They compared taxicab geometry [Kra12] and the maximum norm to a norm defined by Equation 4.5.

$$ODG_{aggregate} = \sum_{excerpt \in TestSet} ODG_{excerpt}(x_1, x_2, ..., x_i)^4 \tag{4.5}$$

They concluded that the l_1 norm often produces unusable results through specialization, while the maximum norm fails to recognize good solutions because of insignificant degradation in some items. The cost function found to be most suitable and used in both [HZ09a] and [PWO15] is given by Equation 4.5. In the work of this paper both Equation 4.5 and Equation 4.4 have been tested as cost functions and it was found that Equation 4.5 produces better general results.

There are two special cases in which the value of the objective function deviates from ODG values.

Invalid parameter values

The genes of an individual represent particular properties that modify the runtime behavior of the encoder. Certain values, or ranges of values affect this process adversely, to the point of the encoder malfunctioning due to excessive values, buffer overflows or floating point exceptions.

When the encoder fails to encode excerpts with the genes of an individual, this individual has lost all validity to be reconsidered in the subsequent evolutionary process. To avoid its further survival and procreation, in Equation 4.6 a very high objective function value is assigned to individuals which caused the encoder to malfunction, usually a fixed value between 25 and 100 is assigned.

$$ODG_{aggregate}(x_1, x_2, ..., x_i) = 25 | (x_1, x_2, ..., x_i) \in X_{invalid} \tag{4.6}$$

The algorithm is expected to produce a large number of individuals with genes set to invalid values, while it traverses the available parameter space quickly. In the remainder of its execution, the occurrence of invalid individuals is expected to decline with time. The exploration of invalid individuals during the genetic optimization process, is an indicator that the evolution is considering areas of the parameter space that have not been mapped out yet, which is a good sign for the quality of the evolution.

Default Parameter Values

The second special area of the optimization process, are parameters with negative values. The genes of individuals generally represent positive floating point values in the source code of the encoder, so that negative values have no intrinsic meaning to the optimization process.

When negative parameter values are encountered by the encoder tool, it will discard the configuration and instead use the default for that particular value, i.e. the value that it was set to before the optimization through the algorithm of the thesis.

$$ODG_{excerpt}(x_1, x_2, ..., x_i) = const | (x_{1_{default}}, x_{2_{default}}, ..., x_{i_{default}}) \in X_{negative} \tag{4.7}$$

As Equation 4.7 shows, this creates a flat error surface in the negative half-space regions of the parameter space. During the creation phase of the first generation of individuals, gene values are initialized randomly with values ranging from 0 to 20 with equal probability for each floating point value in between. Through the genetic operation of mutation in the GA, it is expected that some genes will be negative in some individuals at some point of the GA. Through crossover operations the negative values can proliferate throughout the population of the GA, if the default values of the encoder offer an improvement over other values tried by the GA.

This property of the algorithm offers advantages for the success of the optimization, and also ensures that the default parameter values are included in the optimization process. The default values were manually adjusted by audio tuning experts and may be valuable for the optimization. In case there is some parameter, whose value is already a global optimum, determined by researchers, there is no use in using a GA to try to descend into that global optimum. The algorithm will descend into the negative subspace for that particular gene, and the superior performance of the individual containing the gene will ensure its survival and proliferation.

The amount of genes having negative values in the resulting individuals of an optimization process is also open for interpretation. If there are only negative values, it means that the algorithm has not been able to find any constellation of gene values that were able to outperform the default values and the encoder settings are sufficiently tuned in that regard. If on the other hand, if all values are positive and defined, then it indicates that

the perceptual quality of audio encoding offers a lot of room for improvement, because the default values have been outperformed and disregarded.

4.2 Description of the system

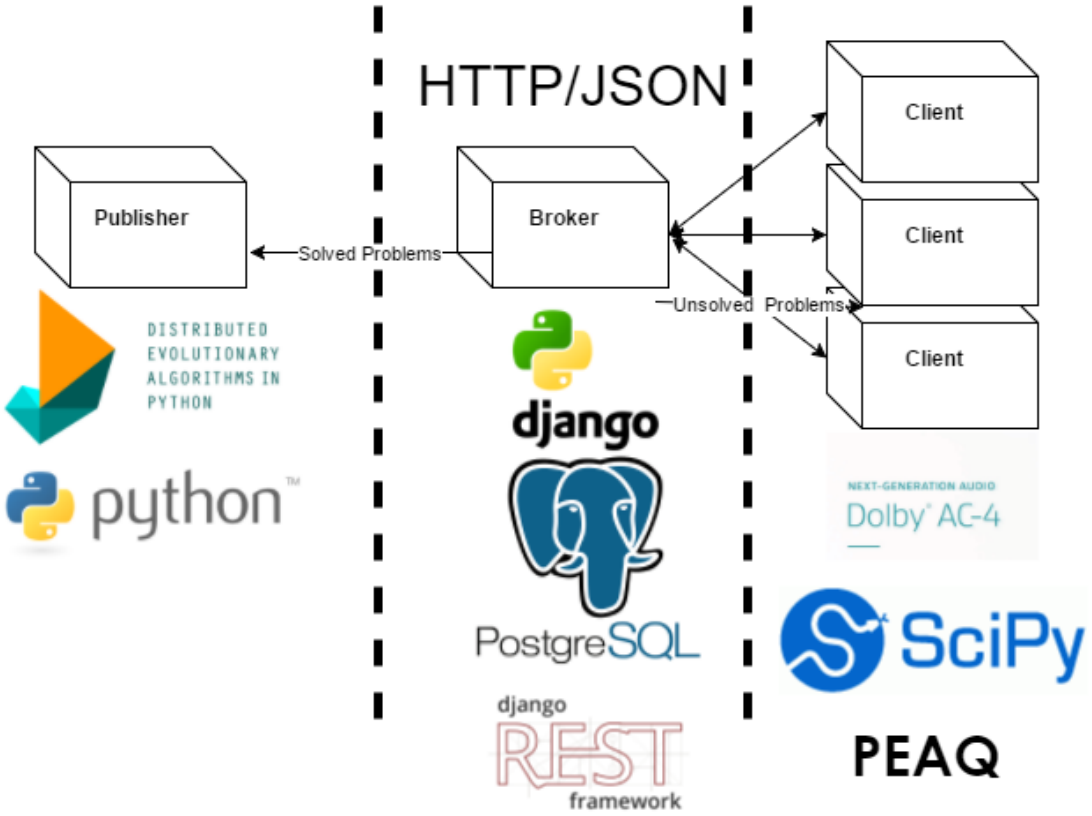

Figure 4.3: Overview of the system and technologies used in different parts.

The system used to run the optimization is depicted in Figure 4.3. In the following section an overview over the individual parts is given.

4.2.1 Publisher

The procedures that determine which individuals should be tested next, are encapsulated in the conceptual entity of the Publisher. The purpose of the Publisher is to find prospective new values for the parameters currently being optimized. On the one hand, the algorithm must use its accumulated results to generalize the underlying error surface and make predictions about new parameter constellations, that are likely to produce high fitness solutions.

Simultaneously, it must accept a certain amount of bad or random constellations, in order to break out of local optima and explore inaccessible areas of the parameter space.

GA

As mentioned above, the main mechanism driving the optimization forward is a GA. The probability for mutation and crossover were varied between 0.1 and 0.9. The individual parts of the GA such as crossover, mutation and selection strategies were implemented using the Python DEAP library [FRG+12] [RFG+14].

Local Search

GAs are able to traverse large portions of a parameter space in an efficient manner that escapes local optima well. However, they have relatively poor convergence behavior towards a final local optimum [TMKH96] [HZ09b]. This is the reason why the optimization process executed by the GA is often followed by a local search method like in [HZ09b]. To descend into local optima at the end of the execution of the GA, a gradient descent method is used on the best performing individuals after the evolution has finished. A common number of individuals to select is between one and ten.

The error surface defined by the objective function is non-continuous [HZ09a] and no obvious definition for the gradient at a point X in parameter space exists. In order to calculate the gradient, the partial differentials for the parameters $x_1, x_2, ..., x_i$ are approximated as in [BKS08]. Individuals for which one of the parameters has been incremented or decremented by a small value ϵ are evaluated in order to gain insight about the impact of small variations of a particular parameter on the ODG value.

Taking the difference between the ODG value of two points that only vary in one dimension x_1 by a total of 2ϵ, results in an approximation for the partial in dimension x_1, described in Equation 4.8. The value of ϵ should be big enough to cause a change in perceptual audio quality of the resulting individual, but as small as possible to approximate the value of the partial differential correctly.

$$\frac{\partial ODG}{\partial x_1} = PEAQ(CODEC(x_1 + \epsilon, x_2, ..., x_i)) - PEAQ(CODEC(x_1 - \epsilon, x_2, ..., x_i)) \quad (4.8)$$

For each chosen individuals 10 steps of iterative gradient descent steps are executed. Calculating the gradient in i dimensions requires the evaluation of $2 * i$ individuals. For a local search in $i = 10 dimensions$, $n = 10$ individuals, $j = 10$ steps the total number of evaluations is given by Equation 4.9.

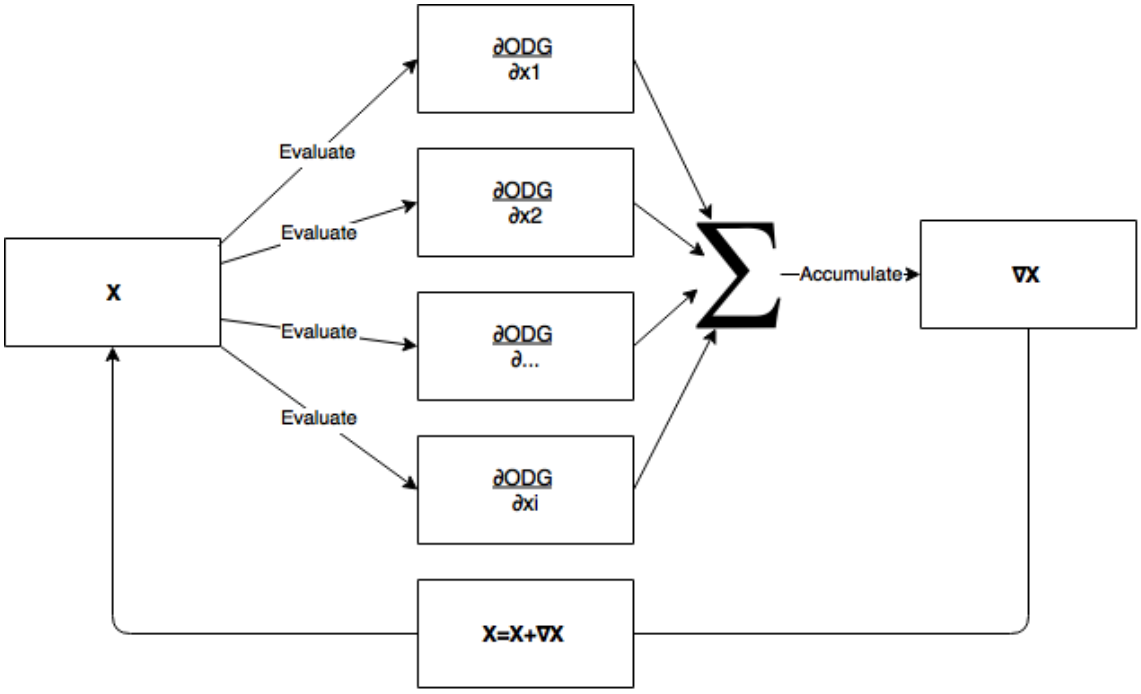

Figure 4.4: The gradient descent method used in the local search algorithm.

$$|individuals| = (1 + 20 * 10) * 10 = 2010 \tag{4.9}$$

While the amount of evaluations necessary to carry out the gradient descent method for local search is big. The advantage of the local search by gradient descent is, that it can improve the performance further for local optima. Furthermore by examining the proximity of some optimum found by the GA, the locally best parameter configuration can be found. The individuals created by the GA with the greatest fitness are few in number and only span a small portion of the entire parameter space. This means that the region containing the best performing individuals is only sparsely discovered. Many points in the high performing regions of parameter space are investigated, because the calculation of the gradient requires many evaluations of individuals from that regions. The amount of discovered points in the upper fitness region is increased, which is highly beneficial for the following steps.

Parameter Optimization

When attempting to optimize large sets of parameters it is often not feasible to optimize all of them at the same time, owing to the curse of dimensionality. The curse of dimensionality states, that when the dimensionality of a problem increases, i.e. the number of parameters,

the parameter space grows so fast that the available data, i.e. the individuals that have been evaluated, become sparse and the ability to generalize from the data and make predictions about it is inhibited. There is another problem when choosing parameters which directly correspond to source code changes of the encoder. The GA works best when the parameters $x_1, x_2, ..., x_i$ form a basis of the parameter space, and no linear dependencies exist between the different parameters. In reality however, it is common that several parameters are linearly dependent. It is possible to discover the dependencies between parameters and take advantage of them, to reduce the dimensionality of the problem.

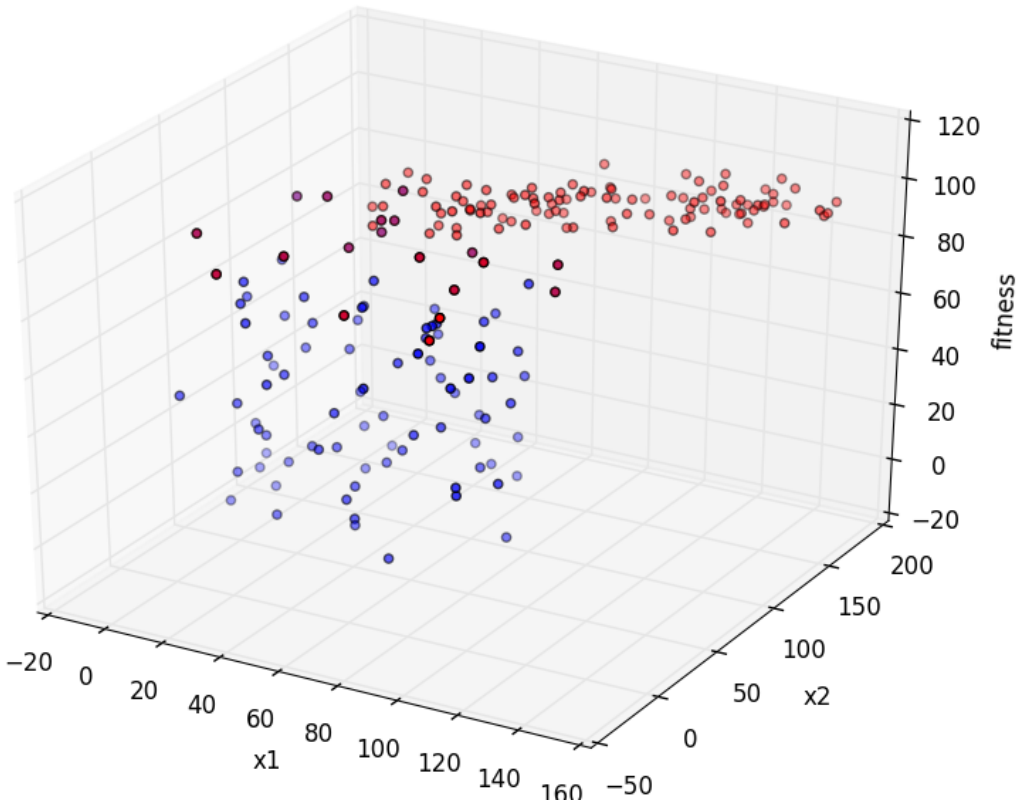

Figure 4.5: Fitness across individuals with a large region of fit individuals (marked red).

The work of this thesis has found that it is therefore efficient not to optimize many parameters at the same time, but rather group them in small groups, optimize them individually and then set the best found values for some parameters before proceeding to optimize other groups. This process was automated and embedded within the optimization algorithm

Applied to the problem of audio encoder tuning it means to classify the different parameters according to their impact on the perceptual audio quality. Some parameters may have a more pronounced effect on the resulting audio quality than others, and offer much room for improvement. Other parameters might have to be fixed to a certain value, but do not improve thereafter. To optimize the choice of parameters two different approaches have

been examined, one relying on simple statistical properties, and one more capable relying on principal component analysis.

The process starts out by choosing a set of parameters M, that should be taken into consideration by the algorithm. The cardinality of M may be large, and M forms a basis of the parameter space. In a second step a random set of parameters $\overline{M_1}$ is chosen from M. The cardinality of $\overline{M_1}$ may be set by the experimenter.

Subsequently, the optimization by the GA and local search methods are carried out. After the process finishes, statistical analysis is applied to the best performing of individuals $\overline{B}$. The set $\overline{B}$is marked in red in Figure 4.5. The process rates the importance of dimensions for the optimization. Subsequently, a choice about a new coordinate system for the following iterations of the GA and local search can be made.

Variance Optimization

In the simplest case, the chosen parameters are linearly independent, and only need to be classified by their variability in the set of the best performing indivduals $\overline{B}$. A sample set with two varying parameters x_1 and x_2 is depicted in Figure 4.6. Parameters which have little variability correspond to genes whose value must be set to some explicit value to achieve good perceptual audio quality in the resulting audio signals.

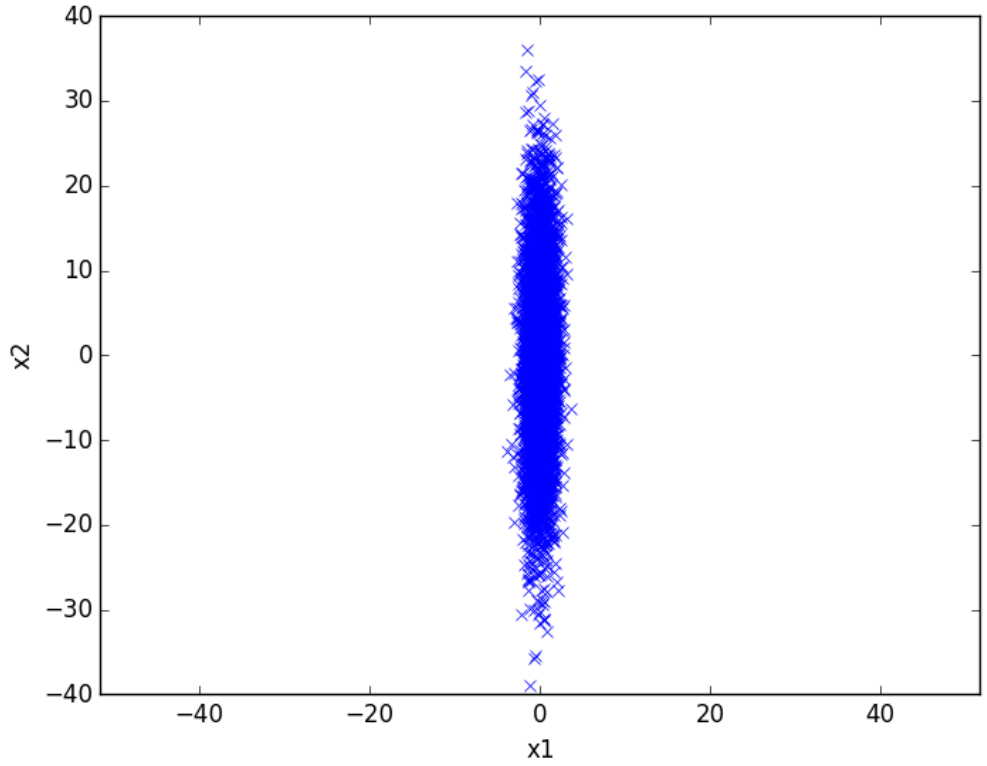

Figure 4.6: The set $\overline{B}$for a simple case.

$$cov(X) = (\begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}) \tag{4.10}$$

The distribution of point in Figure 4.6 was generated using the covariane matrix in Equation 4.10. It is apparent that the variance of x_2 is much greater than x_1. It is reasonable to set x_1 to its mean value 0, and keep it fixed for the remainder of the optimization. In a next step a new dimension x_3 may be added into the pool of parameters to optimize, because it is likely that there is more room for improvement by varying x_3.

Principal Component Analysis Optimization

In many cases, the parameters chosen to be optimized are not linearly independent and therefore the co-variance matrix of the cluster of $\overline{B}$is not diagonal, i.e. it is skewed or rotated as depicted in Figure 4.7. Analysis of the variance alone is then not sufficient to discover the redundancy in the data, and finding a new optimal coordinate system for the optimization.

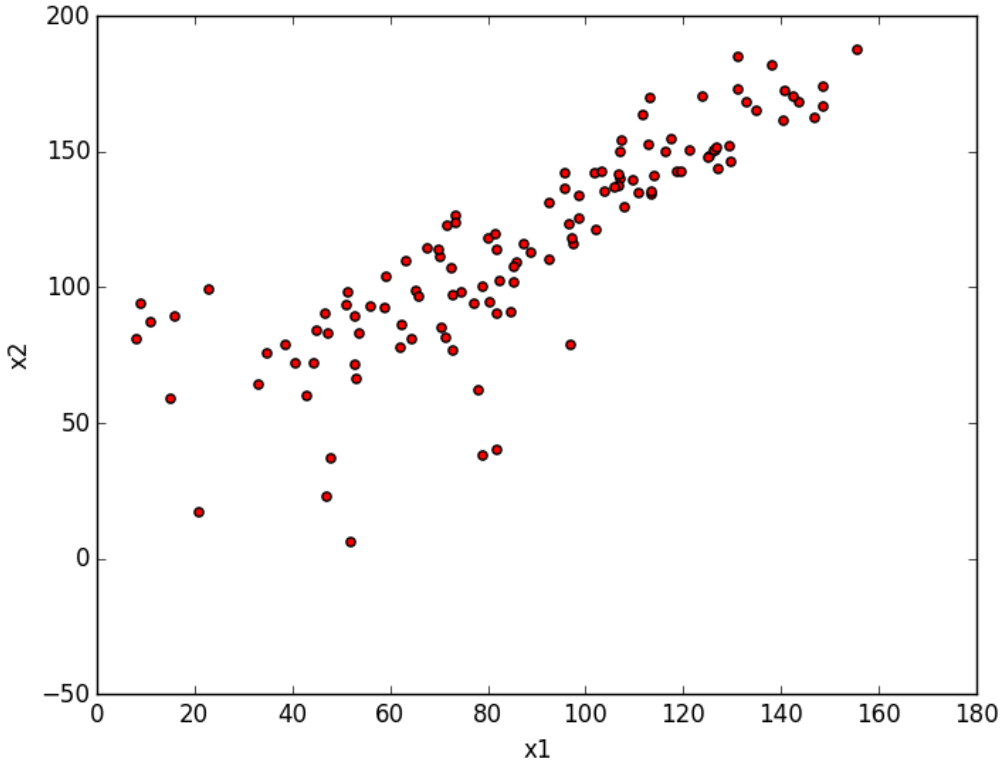

Figure 4.7: The set $\overline{B}$.

The underlying assumption is, that when modifying thresholds together, they are not entirely linearly independent of each other. It is more efficient to respect the linear dependency in the parameters, than to vary each parameter absolutely independently. Principal component analysis is suitable to solve this problem and choose a subset of parameters to be optimized [WEG87]. The Principal Component Analysis transforms the original coordinate system $\overline{M_1}$ into a new coordinate system $\overline{T_1}$. The purpose of this is to find a new basis for the original parameter space $\overline{T_1}$ that is normalized and finds linear relationships between parameter values in order to represent most of the variability in the underlying data with as few basis vectors as possible. The result of a PCA carried out on the data from Figure 4.7, is shown in Figure 4.8.

The principal component analysis also provides information about the variability of the data represented by each of the new base vectors in $\overline{T_1}$. In Figure 4.8 this is indicated by the length of the vectors, and is apparent that one of the vectors represents much more variability than the other. The parameter choice algorithm in this thesis then attempts to choose vectors from $\overline{T_1}$ so that a large portion of the variability, between 60% and 90% is retained. The values 60-90% were heuristically derived. The base vectors which only represent very little of the variability in the data indicate linear relationships between

parameter values that are similar across individuals.

Since the Principle Component Analysisis carried out only on the set of the top 10% performing individuals $\overline{B}$, linear relationships represented by the vectors with the least underlying variability point to parameter values that are advantageous for the individuals and may be set to a fixed value for the remainder of the optimization process.

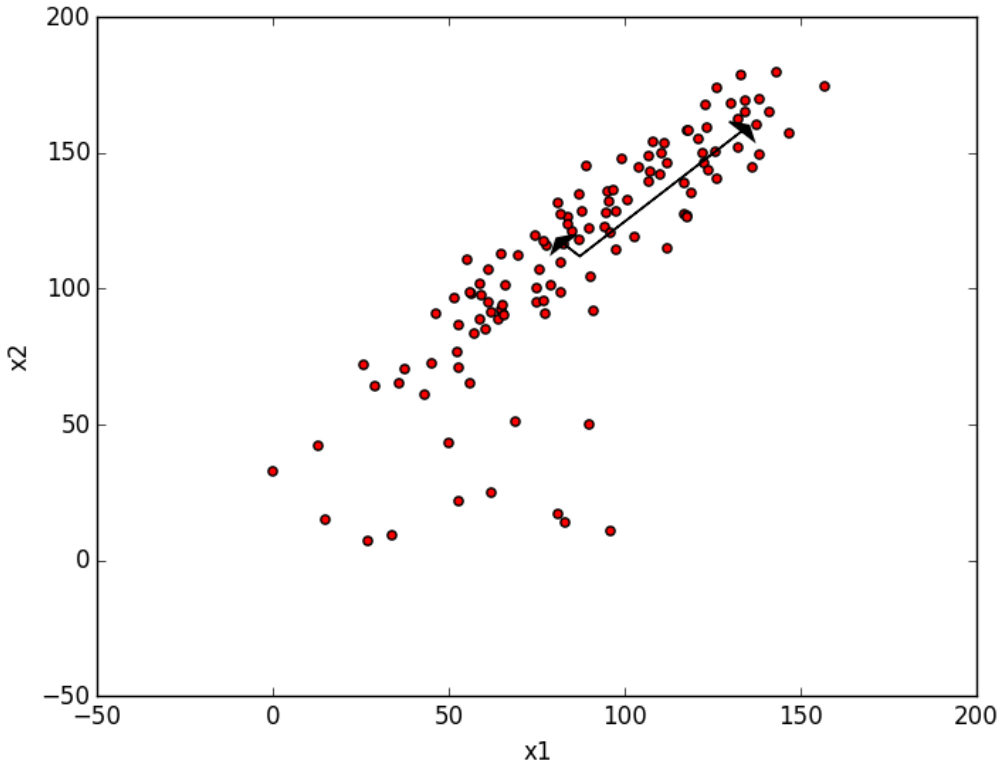

Figure 4.8: The principal components weighted by the variability in the data they represent.

In implementation that means averaging the values for the base vectors of $\overline{T_1}$ that cumulatively represent less than 40-10% of the variability across $\overline{B}$, and setting their values to fixed numbers accordingly. Additional optimization over these parameter values by successive iterations of the GA and local search represent relative changes.

This approach has two advantages over the simpler statistical variance optimization method. Firstly, the algorithm is not restricted to linearly independent dimensions and can operate in rotated or skewed parameter space. Secondly, because the newly found basis consists of unit vectors which are composed of the original parameters, by removing a vector with little variability from the new basis does not affect the ability of the algorithm to change the value of the underlying parameters in the code entirely. Most of the ability to modify the source code is preserved, for possible future interaction with parameters that

are added to the pool of parameters to be optimized. The result of the optimization process is determined by the sum over all the parameter values that have been set to specific values during the optimization process.

4.2.2 Client

The conceptual entity of the Client is concerned with processing the information contained in an individual with a particular set of genes. It is implemented as a Python script, that communicates with the Broker to retrieve new individuals, as they are produced by the Publisher. The Client coordinates the work of several tools that must be applied to a set of excerpts consecutively.

The first step in the processing chain is to retrieve the encoder configuration settings from an individual and apply these to the encoder shipped with the Client. At an early stage of the work presented in this thesis, this involved modifying the source code with the GNU/Linux command line tool "sed", before compiling the encoder executable from source.

This procedure had several drawbacks. The source code of the encoder utility had to be distributed to every machine the Client was supposed to be run on, which added significantly to the file size of the Client. Furthermore, the invocation of a C compiler like gcc for linux or MSVC for Windows increased the time needed for the entire process drastically.

In one of the major improvements during the implementation phase of the system, the encoder source code written in C was modified to enable the experimenter to set configuration settings from the command line. Thus, the necessity for recompilation of the encoder for changing parameters could be eliminated. This removed both time and space overhead on the Client, and reduced the complexity of the entire process.

The result of the encoding process is a .ac4 file, which contains a proprietary bit stream. The bit stream encodes the audio information of the excerpt. It must be decoded with the Dolby AC-4 decoder utility, which remained the same on all systems during the work of the thesis, and was only built once at the beginning.

Decoding the AC-4 bit stream extracts the encoded information, and the audio signal is converted back to a standard wave file format. During the encoding and decoding process a fixed buffer silence is added at the beginning of the audio signal. Objective measure tools like PEAQ, compare two audio excerpts for similarity and degradation on a frame-by-frame basis. The two files must therefore be time-aligned and the leading silence must be removed. This is achieved with a Python script using the standard library.

The Dolby AC-4 codec is a parametric codec, which does not preserve the original waveform. Instead, a technique called spetral band replication (SBR) is applied to the signal. SBR attempts to code relationships between the lower frequency bands of a signal and higher frequency bands efficiently. The information is coded into the signal in addition to

Excerpt Bitrate	Cutoff Frequency
48000 kbit/s	7500 Hz
64000 kbit/s	10500 Hz

Table 4.1: Cutoff frequencies for the Butterworth filter with respect to bitrate of the excerpt.

the waveform-preserving lower frequencies. When the AC-4 bit stream is decoded, noise and harmonics are inserted in the higher frequency bands according to the parametric information in the bit stream. This process produces artifacts that PEAQ was not designed to measure, and therefore PEAQ cannot be used to compare the full-bandwidth signals.

Before comparison of the original excerpt and the processed excerpt, both must be low-pass filtered to remove the frequency bands in which A-SPX is active. A Python script with the SciPy library applies a Butterworth low-pass filter of order 20 to both excerpts with a cutoff frequency which depends on the bit-rate used to encode the excerpt. The relevant bitrates are given in Table 4.1.

PEAQ

The low-pass filtered excerpts are then analyzed by PEAQ. PEAQ as recommended in [Rec98]is encumbered in patents and available under license according to ITU fair, reasonable and non-discriminatory terms. For the distribution in a cluster, many licenses would have been required, which was unfeasible. Instead a free implementation of PEAQ Basic, PEAQb was used. PEAQb achieves the same functions as PEAQ Basic, but was not validated with ITU data. PEAQb is available for free for educational use and was compiled from source for distribution. It was acquired from McGill university as part of the Audio File Programs and Routines package [EU10].The ODG for the entire file is parsed from the output of PEAQb on the client after the comparison is finished.

4.2.3 Broker

Distributed and massively parallel systems pose additional difficulties in comparison to systems that can be run on one machine. Communication over the network is slow and unreliable, which is why standardized communication interfaces are beneficial. Additionally, there is always the risk of partial failure of the systems. Causes for failure of the systems may be loss of power, network failure, deadlock or human intervention. Any kind of failure in the system will lead to an ambiguous state in which it is unclear to the rest of the system, whether the evaluation started by a machine will terminate, or whether the evaluation should be reassigned to another machine in the cluster. Finally, probing the state of the system or process requires a single point at which information is gathered,

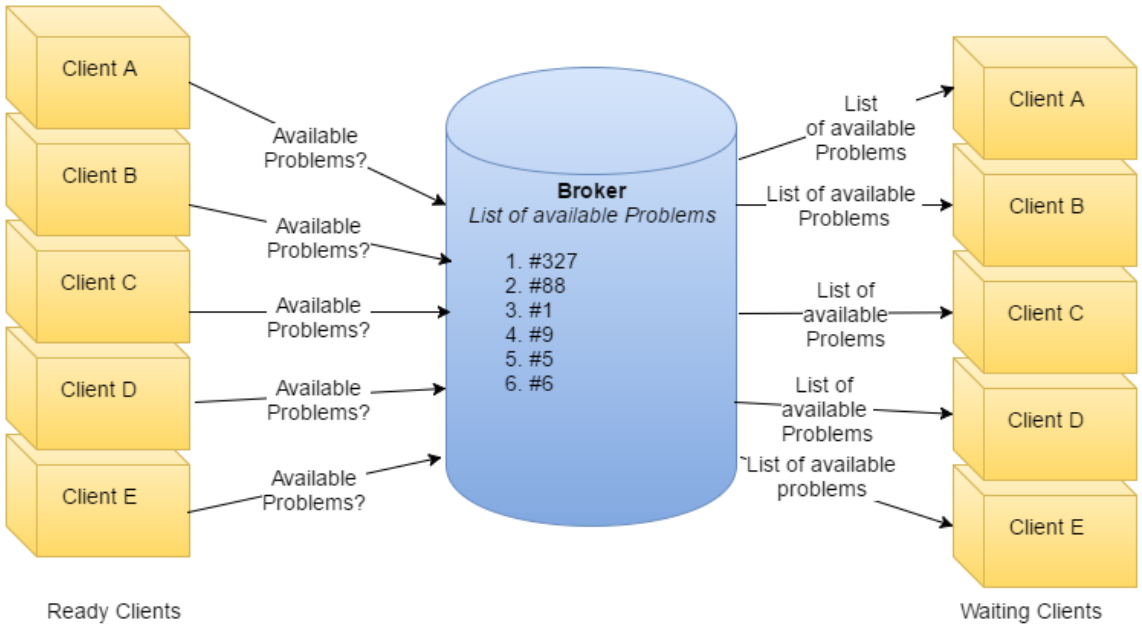

Figure 4.9: Waiting Clients requesting the list of available problems.

stored and refined for evaluation. This relates both to the actual results (individuals with parameter configurations and PEAQ score) and the performance Figures of the system (e.g.: runtime of evaluation and processing, overall number of evaluations per second).

The piece in the system connecting the cluster of Clients with the Publisher and the GAs is referred to as "Broker". It is run separately from the rest of the system, ideally on a separate machine to provide additional safety from machine specific failure. It is implemented as a web service, using the Python Django framework [dja15] to provide backend functionality. Django also implements a Object Relational Model, that allows storage and retrieval from a database system. The database used for the Broker in this thesis was the free and open source PostgreSQL [Pos15] database.

Data Model

Within the database system, information is stored about the individuals that have been evaluated so far, their parameter configurations and associated PEAQ scores. It also stores information concerning the status of the evaluation process of individuals. The exact information stored in the database is shown in Figure 4.10. The little n on the side of the parameter in the relation signals, that one Problem can have many Parameters.

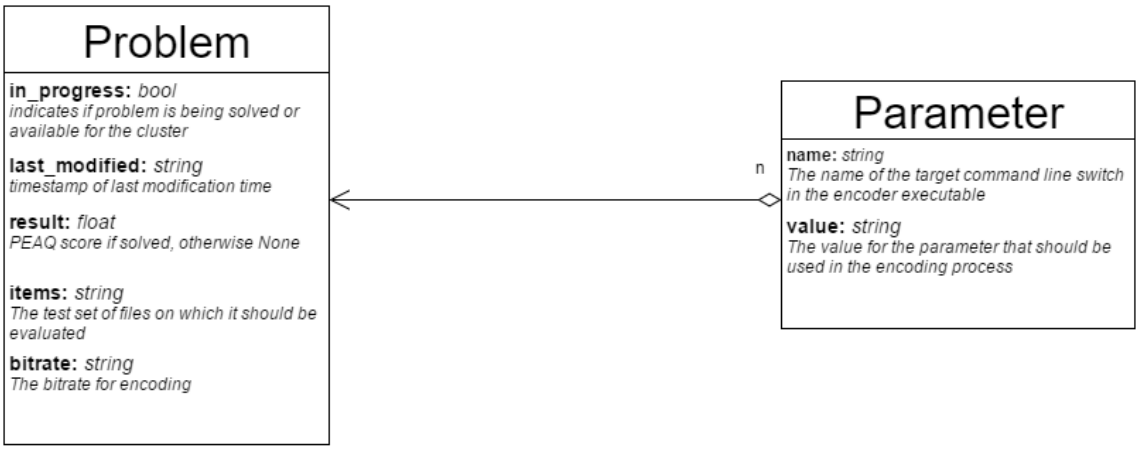

Figure 4.10: The data model in the PostgreSQL database.

Synchronization

The process begins with Clients who are currently not in the process of evaluating an individual requesting a list of available problems from the Broker. The Broker replies with a list of available individuals, as depicted in Figure 4.9. The Clients then choose a problem from the list randomly and request the Broker to mark the problem "in progress".

Occasionally, as depicted in Figure 4.11, two Clients request the same item simultaneously. Multiple evaluation of the same individual is a waste of computing power and should therefore be avoided. PostgreSQL implements atomic transactions, which are used to guarantee that one individual will always at most be distributed to one machine in the cluster, even in highly parallel environments. The Broker then accepts the first incoming request, and denies all subsequent ones. This way, there is no loss of performance of the cluster due to multiple evaluation of the same individual.

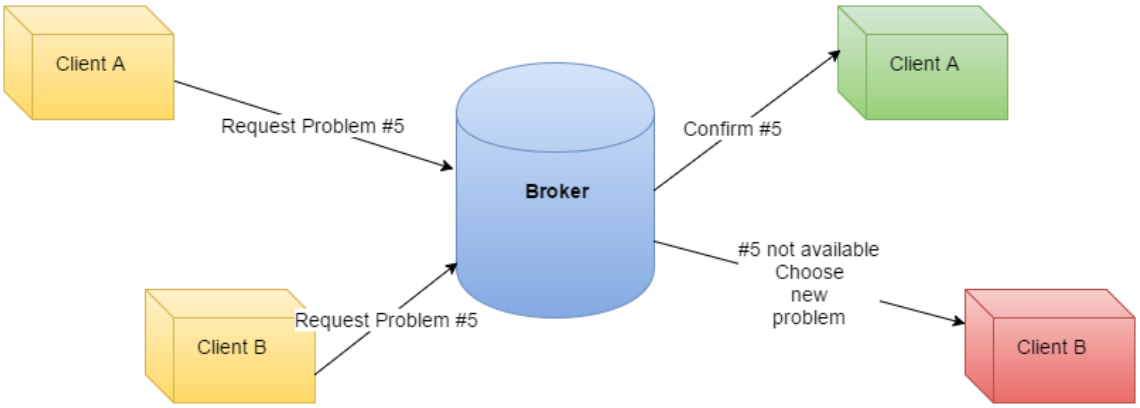

Figure 4.11: The Broker resolves an attempt of two Clients to solve the same problem.

Individuals which have been marked “in progress” for more than sixty seconds without having been assigned a PEAQ score are considered stuck, and are made available as unsolved problems for the cluster. This is an important feature of the system, in order to detect Clients who have failed. Because problems are redistributed by the Broker after a while, the adverse effects of Clients which become unresponsive is attenuated.

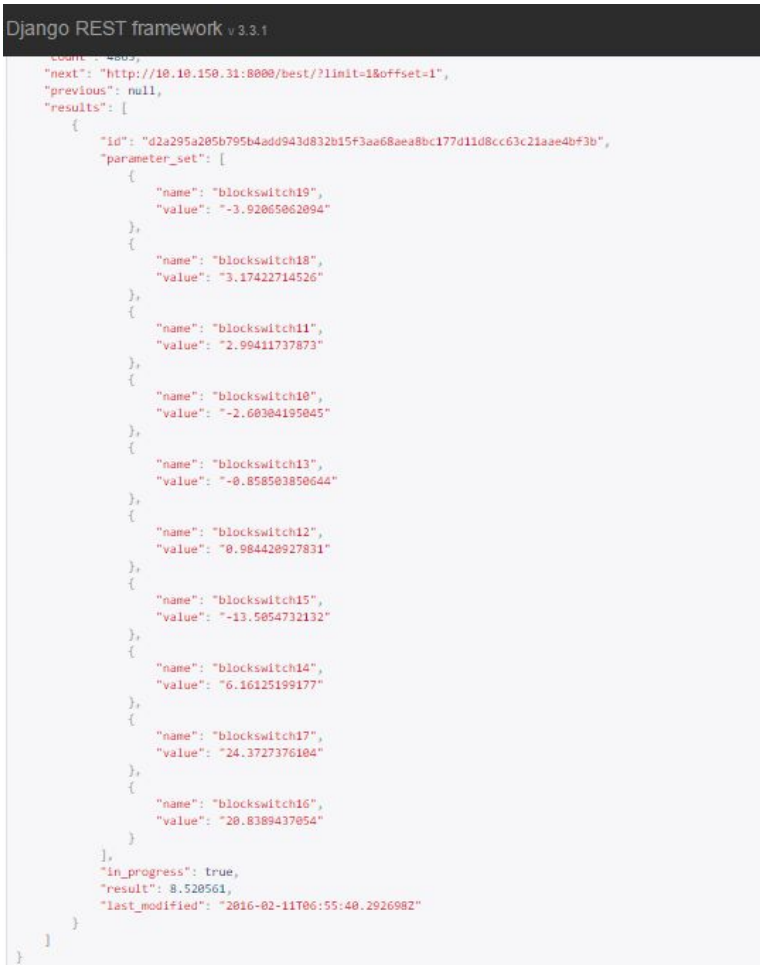

Figure 4.12: The Djangorestframework interface used for results retrieval.

Data probing and result extraction can be carried out via a Web interface provided by the popular djangorestframework library depicted in Figure 4.12.

4.3 Description of Sets of Excerpts

Different sets of excerpts have been used to either train the GA or validate resulting parameter constellations.

4.3.1 MPEG Set

The MPEG set, enumerated in Table 4.2, is one of the two major sets of audio excerpts with which the automatic audio encoder tuning in this thesis was carried out. It contains a variety of signals including pure speech items (es01, es02, es03), pure tonal signals (sc01), transient rich signals (si02) and mixed signals with speech, transients and tonal parts (sc03).

Item name	Description	Length
es01	Suzanne Vega, solo	00:10
es02	male speech, german	00:08
es03	female speech, english	00:07
sc01	trumpet	00:10
sc02	orchestra	00:12
sc03	pop music	00:11
si01	harpsichord	00:07
si02	castanets	00:07
si03	pitch pipe	00:27
sm01	bagpipe	00:11
sm02	glockenspiel	00:09
sm03	plucked strings	00:13

Table 4.2: The MPEG test set.

4.3.2 ATSC3.0 Set

The ATSC3.0 set enumerated in Table 4.3, is the set of audio files on which the current contenders for the audio standard of the new ATSC3.0 standard are evaluated. It contains a variety of signals, including pure speech items (09-Vega), pure tonal signals (03-Classic), transient rich signals (02-Castanets) and mixed signals with speech, transients and tonal parts (08-Rea). Typically evaluation sets for standards like ATSC3.0 are representative for general classes of audio.

4.3.3 Applause

In order to examine the possibility of optimizing the configuration of the encoder for the particular class of applause audio, a single item set “applause” was also used (Table 4.4). It is the Left-surround and Right-surround of the applause that was used in [Cod07].

Item name	Description	Length
01-Accordion	Accordion music and jingles	00:17
02-Castanets	Many sharp attacks and transients	00:14
03-Classic	Trumpets and wind instruments	00:25
05-Harpsichord	Harpsichord sounds rising in pitch	00:17
06-Hockey	Female speech with crowd cheering	00:19
07-Orchestra	Mixed signal with transient and tonal parts	00:19
08-Rea	Excerpt from rock song “On the beach”	00:15
09-Vega	Female singing	00:20
10-Golf-20-NBCU	Male Speech with crowd cheering	00:12
11-Hockey-20-NBCU	Male Speech with stadium sounds	00:12
12-News-20-NBCU	Male Speech	00:12

Table 4.3: The ATSC3.0 set.

Item name	Description	Length
09-Applaus	Clapping and standing ovations	00:19

Table 4.4: The applause excerpt used in [Cod07].

Chapter 5

Results

5.1 Window Switching

5.1.1 ATSC3.0 Set

The first experiments conducted, were concerned with optimizing the thresholds determining, how the length of the transformation window should be chosen for given frames in an audio excerpt. To influence the behavior, nine different parameters were adjusted with a GA. The GA was run with a mutation probability (MUTPB) of 0., and a random real number between -1 and 1 was added to a gene in order to perform mutation. Crossover was performed as multipoint crossover, with a probability of CXPB of 0.5, a population size of 100, and a total of 150 generations. The optimization was carried out over the ATSC3.0 test set, and PEAQ scores were acquired using PEAQb.

The experiment was run twice with two different cost functions, in order to compare the

Parameter	Default value	cost2 value	cost4 value
parameter0	0.3	0.35	0.64
parameter1	20.0	0.42	17.53
parameter2	10.0	10.75	6.46
parameter3	5.0	5.84	2.41
parameter4	1.4	7.29	8.9
parameter5	0.001	4.74	12.09
parameter6	0.001	0.72	-0.11
parameter7	0.001	-0.29	20.32
parameter8	0.001	13.84	15.72

Table 5.1: Parameter values for *cost*2 and *cost*4 optimizations.

influence of the cost function on the improvement in perceptual audio quality. The two cost functions used were Equation 4.4, referred to as *cost*2 in the results and Equation 4.5 referred to as *cost*4. The results of the optimization were nine new values for the parameters given in Table 5.1.

*cost*2 and *cost*4 differ in the choice of the thresholds from the default parameter settings, in some cases by orders of magnitude. Both *cost*2 and *cost*4 favor longer and fewer transforms in comparison with the default values, with *cost*4 preferring longer transforms more aggressively than *cost*2.

For the *cost*4 parameter set, a visualization of the analysis window switching behavior was done. Figure 5.1, Figure 5.2 and Figure 5.3 show a frame by frame analysis of impact of the parameters on the chosen transform lengths of the 01-Accordion, 02-Castanets and 03T-Classic items respectively. Figures 5.1, 5.2 and 5.3, show three charts for the respective

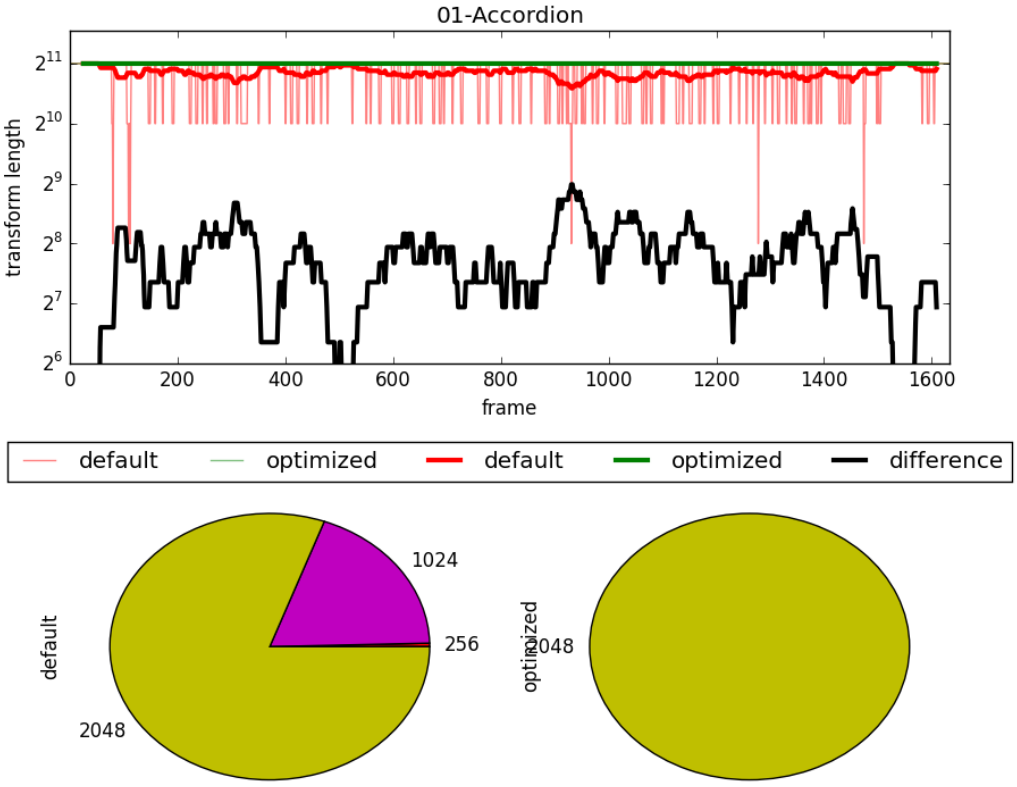

Figure 5.1: Effect of the optimization on the chosen transform lengths for the 01-Accordion item from the ATSC3.0 test set.

items. The pie charts display the total amount of individual transform lengths, that were chosen to code the excerpt. The left pie chart shows the distribution of transform lengths for the default settings of the encoder, and it is seen that three different lengths are chosen: 2048, 1024 and 256. Presumably the encoder chooses the short transform lengths for transient rich passages to gain better time resolution. The right pie chart shows the distribution for the *cost*4 parameter values from Table 5.1. For the optimized values, the encoder only chooses the longest transform length to code the entire excerpt, and does not use shorter transforms.

The top chart shows a frame-by-frame comparison of the chosen transform lengths on a logarithmic scale. The possible choices for the encoder are 128, 256, 512, 1024 and 2048 frames. The transform lengths chosen by the encoder with default values are depicted in red, while the transform lengths chosen for each frame by the encoder with *cost*4 values are drawn in green. For the convenience of the experimenter, the thick green and red lines represent rolling average values over thirty frames. The rolling average makes it easier to draw conclusions from the graph. The black line shows the difference between the two rolling averages and it can be clearly seen in which passages the biggest differences occur.

The clearest difference is seen in Figure 5.2, where the very short 128 transforms are replaced by slightly longer 256 transforms. The castanets item is

PEAQ Scores

The ODG scores obtained with PEAQb can be seen in Table 5.2

Filename	Default	cost2	Delta	cost4	Delta
01-Accordion	-3.209	-3.052	0.157	-3.037	0.172
02-Castanets	-3.25	-3.182	0.068	-3.184	0.066
03T-Classic	-2.885	-2.792	0.093	-2.776	0.109
05-Harpsichord	-3.027	-3.002	0.025	-3.033	-0.006
06T-Hockey	-3.323	-3.283	0.04	-3.287	0.036
07-Orchestra	-3.349	-3.294	0.055	-3.308	0.041
08-Rea	-3.434	-3.443	-0.009	-3.458	-0.024
09-Vega	-3.158	-3.152	0.006	-3.148	0.01
10-Golf-20-NBCU	-2.842	-2.842	0	-2.842	0
11-Hockey-20-NBCU	-3.576	-3.563	0.013	-3.567	0.009
12-News-20-NBCU	-2.055	-1.957	0.098	-2.013	0.042

Table 5.2: The ODG values measured with PEAQb over the ATSC3.0 set.

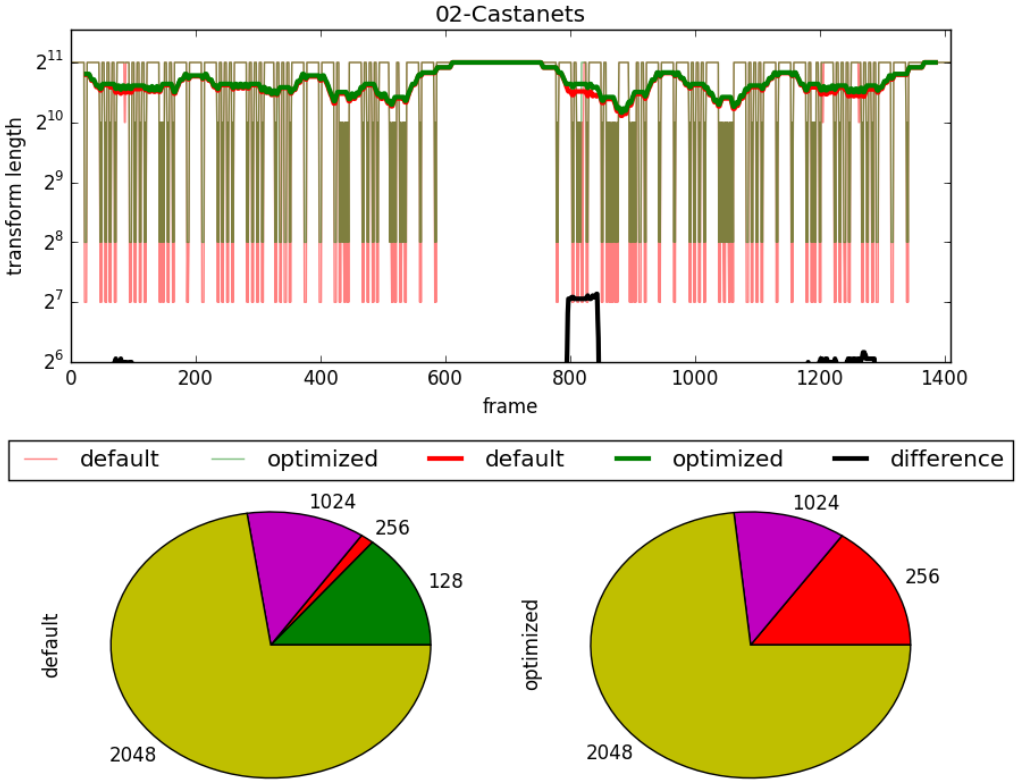

Figure 5.2: Effect of the optimization on the chosen transform lengths for the 02T-Castanets item from the ATSC3.0 test set using the *cost*4 parameters from Table 5.1.

Validation

The *cost*4 parameters were also evaluated with PEAQb over the MPEG set of excerpts. The results can be seen in Table 5.3.

The results in Table 5.3 show five items with slight degradation, five items with slight improvement and two items with improvement of over 0.1. Basing on the listening test in Section 5.1.1 items *es*03 and *sc*01 are likely to show audible improvement in a subjective listening test. The results confirm, that the values found by optimizing the transform-switching behavior with the ATSC3.0 set lead to an improvement of perceptual audio quality on the MPEG set.

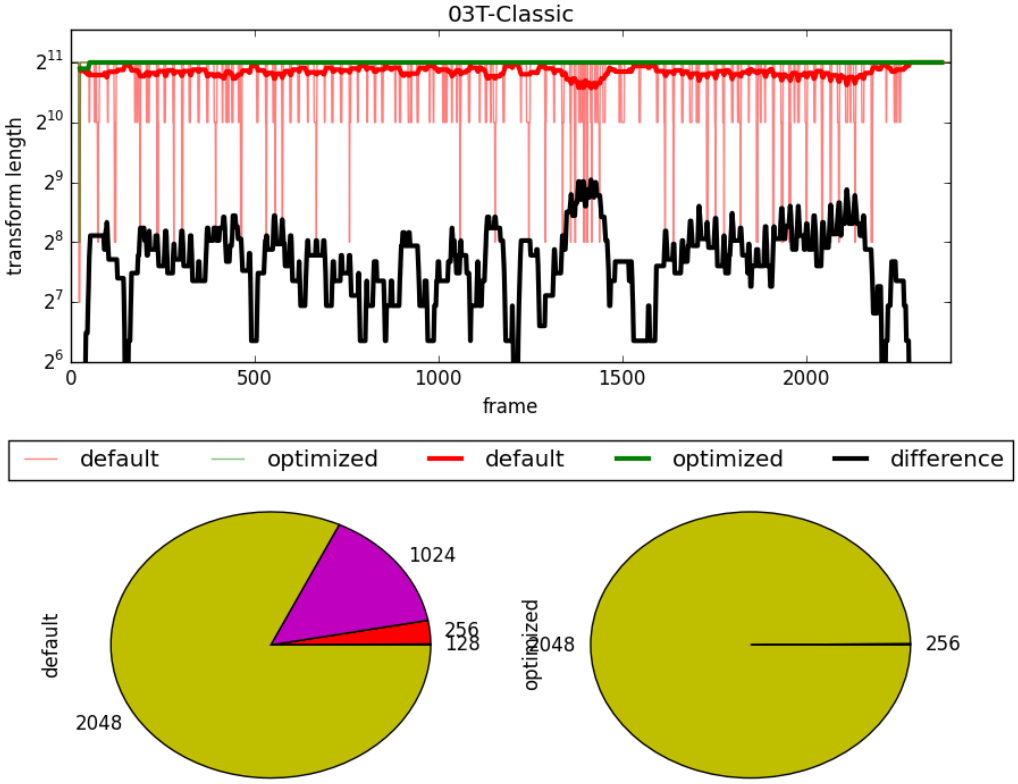

Figure 5.3: Effect of the optimization on the chosen transform lengths for the 03T-Classic item from the ATSC3.0 test set.

Listening Test Results

A MUSHRA subjective listening test was set up, designed to evaluate the performance of the Dolby AC-4 encoder configured with the found optimized parameter values in Table 5.1.

In the following Figures the absolute MUSHRA scores (Figure 5.4) as well as a differential analysis of the MUSHRA scores (Figure 5.5) are given.

The purpose of this test is to investigate if the improvements shown by PEAQ correlate with subjective perceptual audio quality improvement.

In this test there were two codecs at 64 kb/s, ten audio excerpts, and six listening subjects. The test included a hidden reference and a 3.5 kHz anchor; however, the 7 kHz anchor was omitted to minimize test time. Prior to the listening test, the codecs were low-pass filtered

Item	Default	cost4	Delta
es01	-2.568	-2.547	0.021
es02	-1.726	-1.675	0.051
es03	-2.326	-2.193	0.133
sc01	-2.749	-2.554	0.195
sc02	-3.212	-3.255	-0.043
sc03	-3.414	-3.432	-0.018
si01	-2.93	-2.929	0.001
si02	-3.211	-3.265	-0.054
si03	-2.022	-2.025	-0.003
sm01	-2.442	-2.458	-0.016
sm02	-3.12	-3.046	0.074
sm03	-3.175	-3.155	0.02

Table 5.3: The ODG values measured with PEAQb over the MPEG set.

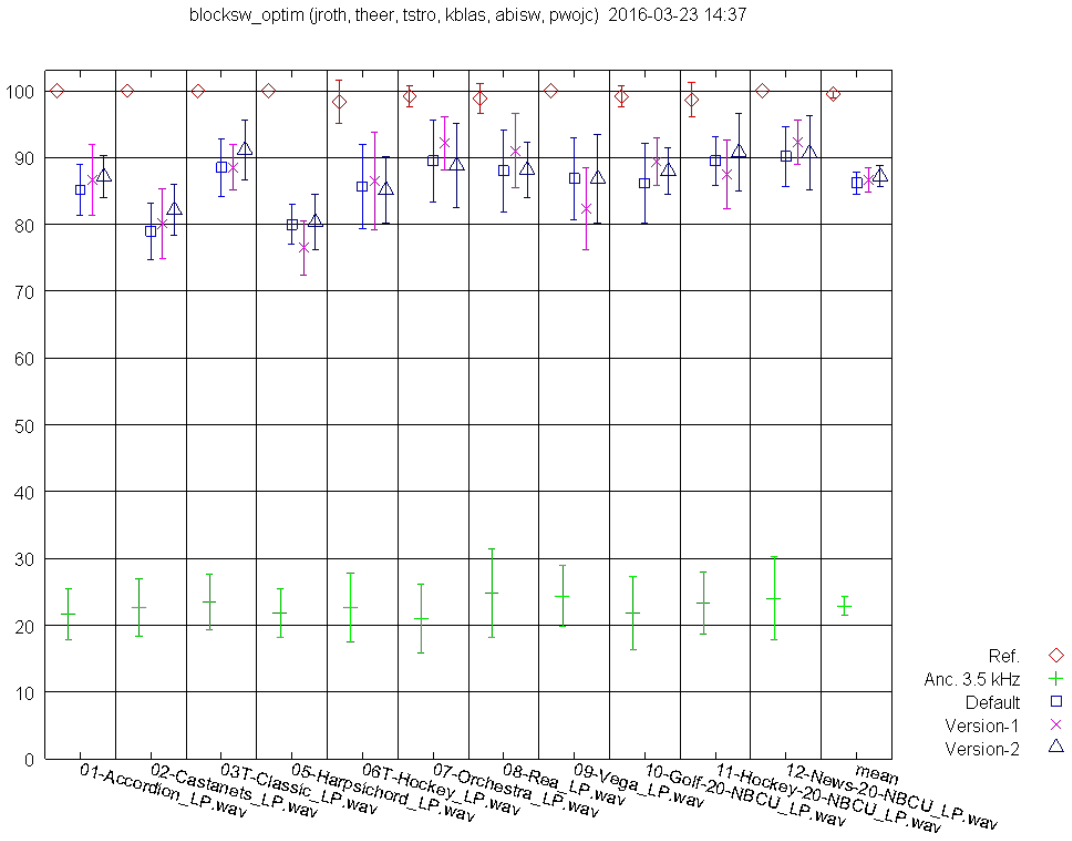

Figure 5.4: Listening test taken over the ATSC3.0 test set, taken by 7 expert listeners.

at 10.5kHz.The three codecs under test are:

1. Dolby AC-4 encoder (Baseline)

2. Dolby AC-4 encoder with parameters modified to the values in the *cost*4 column of Table 5.1

The scores are presented with 95% confidence intervals. The differential analysis is done on the difference scores between selected systems for each listener. In the differential analysis plots, the differential scores with respect to the Baseline encoder are displayed. A positive score with non-overlapping confidence intervals with the zero-line indicates that the systems under test (*cost*2, and *cost*4) are better than the Baseline encoder, and vice-versa.

Observing the differential scores as given in Figure 5.5, it is seen that three excerpts show clear improvement over the Baseline and no excerpts show audible degradation for the *cost*4 codec. For the *cost*2 codec, two items (05-Harpsichord and 09-Vega), show clear degradation. The *cost*4 set of parameters can therefore be considered strictly better than the *cost*2 set of parameters.

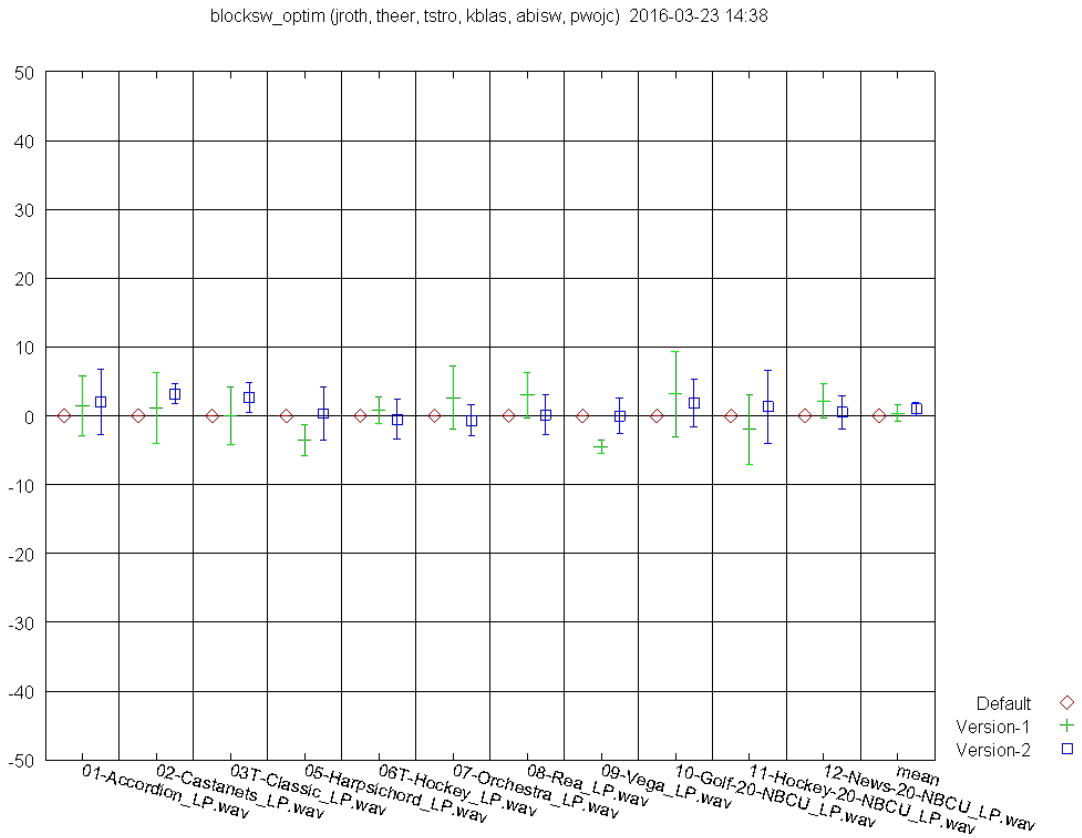

Figure 5.5: Differential listening test taken over the ATSC3.0 test set, taken by 7 expert listeners.

Item	Default	Cost4MPEG	Delta
01-Accordion	-3.24574	-3.085695	0.160045
02-Castanets	-3.274862	-3.213431	0.061431
03T-Classic	-2.949142	-2.872662	0.07648
05-Harpsichord	-3.1267	-3.107544	0.019156
06T-Hockey	-3.358255	-3.333591	0.024664
07-Orchestra	-3.376516	-3.325925	0.050591
08-Rea	-3.467907	-3.526051	-0.058144
09-Vega	-3.201488	-3.227189	-0.025701
10-Golf-20-NBCU	-2.845727	-2.844656	0.001071
11-Hockey-20-NBCU	-3.578988	-3.578023	0.000965
12-News-20-NBCU	-2.129096	-2.054068	0.075028

Table 5.4: PEAQ Basic ODG values for the validation experiment.

5.1.2 MPEG Set

In addition to the experiments yielding the values in Table 5.1, another experiment was set up to evaluate, if as a result of the optimization, an improvement of perceptual audio quality could be achieved. In this experiment the optimization was run on the MPEG set and the subjective listening experience was evaluated with a listening test over the ATSC3.0 set.

The same nine parameters as in Section 5.1.1 were adjusted with a GA. The GA was run with a mutation probability (MUTPB) of 0.3 and a random real number between -1 and 1 was added to a gene in order to perform mutation. Crossover was performed as multipoint crossover with a probability of CXPB of 0.5 a population size of 100 and a total of 150 generations.

PEAQ Results

The ODG scores obtained with PEAQ Basic can be seen in Table 5.4. The results show improved perceptual audio quality for 9 items, while only two items (08-Rea, 09-Vega) show degradation. Similar to the results obtained in Section 5.1.1, the items 01-Accordion, 02-Castanets, 03T-Classic and 12-News-20-NBCU show the biggest improvement.

Even though the results in Table 5.4 show less improvement than in Section 5.1.1, it is noteworthy that the same signals show the most improvement according to PEAQ. This indicates that by tuning on the MPEG set, improvement in perceptual audio quality has been achieved on the ATSC3.0 set.

Listening Test Results

Similar to the MUSHRA test set up in Section 5.1.1, an experiment with expert listeners was set up to confirm the perceptual audio quality improvement with a subjective listening test. The results for the absolute MUSHRA scores are given in Figure 5.6, as well as a differential analysis of the MUSHRA scores depicted in Figure 5.7.

In this test there were two codecs at 64 kb/s, ten audio excerpts, and five listening subjects. The test included a hidden reference and a 3.5 kHz anchor; however, the 7kHz achor was omitted to minimize test time. Prior to the listening test, the codecs were low-pass filtered at 10.5kHz. The two codecs under test are:

1. Dolby AC-4 encoder (Baseline)

2. Dolby AC-4 encoder with optimized parameters tuned on the MPEG set with a cost4 objective function

The scores are presented with 95% confidence intervals. The differential analysis is done on the difference scores between selected systems for each listener. In the differential analysis plot Figure 5.7, the differential scores with respect to the Baseline encoder are displayed. A positive score with non-overlapping confidence intervals with the zero-line indicates that the systems under test are better than the Baseline encoder, and vice-versa.

Observing the differential scores as given in Figure 5.5, it is seen that 01-Accordion and 12-News show a trend for improvement, while the other items either show a trend toward degradation, or are inconclusive due to high confidence intervals.

5.1.3 Applause

Applause is a very challenging class of signal to code with perceptual audio encoders [LKDP11]. In this experiment, the optimization algorithm was run with the same configuration as in Section 5.1.1. The GA was run with a mutation probability (MUTPB) of 0.3 and a random real number between -1 and 1 was added to a gene in order to perform mutation. Crossover was performed as multipoint crossover with a probability of CXPB of 0.5, a population size of 100 and a total of 150 generations. . The same nine parameters as in Section 5.1.1 were optimized using only the applause excerpt. The purpose of this experiment was to find out whether the window switching behavior for applause should be different than for general classes of audio. The obtained parameters are given in Table 5.5.

From Figure 5.8 it is apparent that the algorithm chooses different transforms with the optimized values from Table 5.5, than with the default values. The default algorithm chooses many short transforms (128 and 256). The optimized values make the encoder only choose long transforms for the entire excerpt.

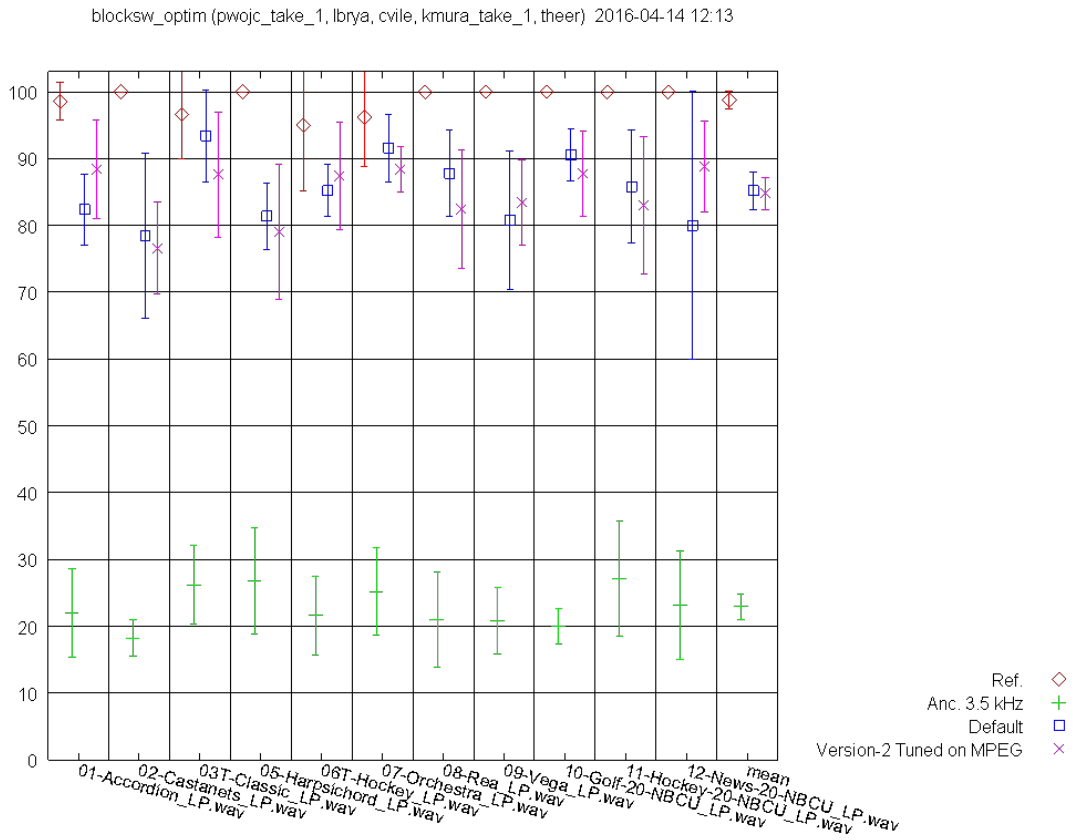

Figure 5.6: Listening test results for the validation experiment taken by six expert listeners.

PEAQ Scores

The ODG values for the applause item with optimized window switching behavior are given in Table 5.6. PEAQb indicates that there has been an improvement in the applause item after the optimization.

Listening Test Results

Similar to the MUSHRA test set up in Section 5.6, an experiment with expert listeners was set up to confirm the perceptual audio quality improvement with a subjective listening test. The results for the absolute MUSHRA scores are given in Figure 5.9, as well as a differential analysis of the MUSHRA scores depicted in Figure 5.10. The purpose of this test is to verify that the subjective listening experience is improved for the applause item.

In this test there were two codecs at 64 kb/s, one audio excerpt, and six listening subjects. The test included a hidden reference; however, the 7 and 3.5 kHz anchors were omitted to

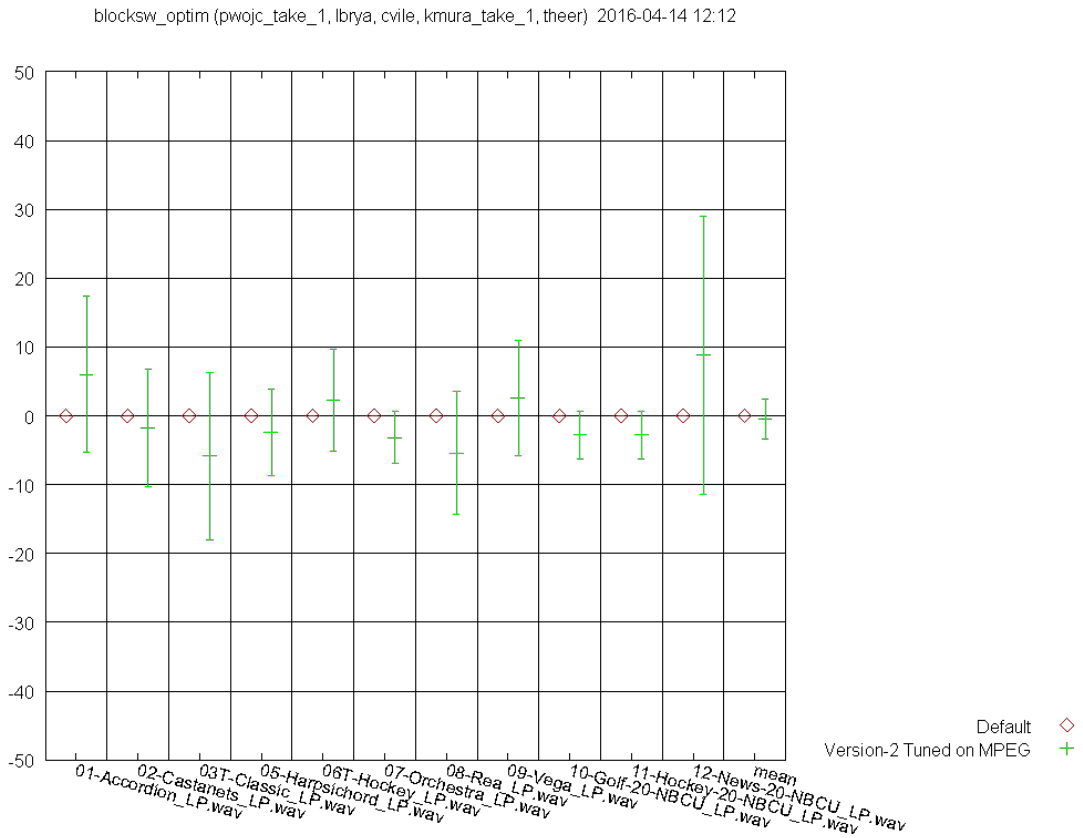

Figure 5.7: Differential listening test results for the validation experiment, taken by five expert listeners.

minimize test time. Prior to the listening test, the codecs were low-pass filtered at 10.5kHz. The three codecs under test are:

1. Dolby AC-4 encoder (Baseline)

2. Dolby AC-4 encoder with parameters modified to the values in the *cost*4 column of Table 5.5.

From the overall plot in Figure 5.9 and the differential analysis plot in Figure 5.10, it is clearly seen that the listening experiments show improvement of subjective listening experience for the applause item.

5.2 Bitreservoir

The bitreservoir technique is a means of controlling pre-echo in difficult to code audio passages. The purpose of this experiment was to investigate whether the default values

Parameter	Default value	cost4 value
parameter0	0.3	0.88
parameter1	20.0	9.88
parameter2	10.0	15.38
parameter3	5.0	6.90
parameter4	1.4	8.28
parameter5	0.001	19.89
parameter6	0.001	16.87
parameter7	0.001	15.63
parameter8	0.001	5.51

Table 5.5: Parameter values for the optimization carried out on only applause.

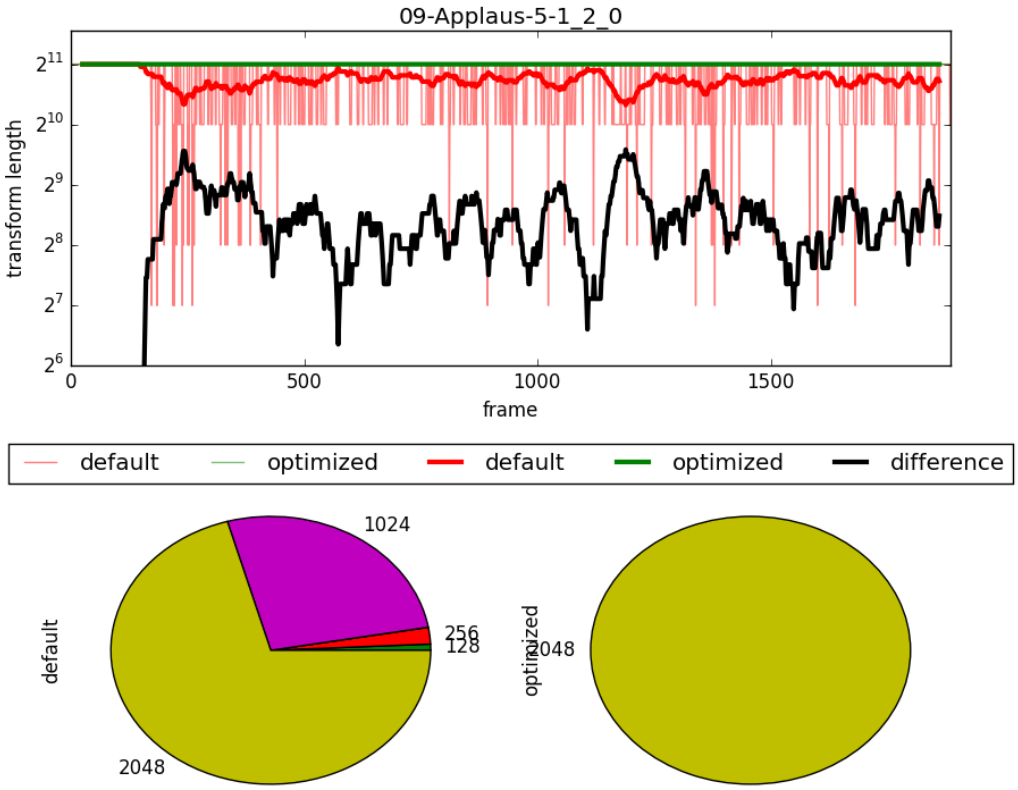

Figure 5.8: Frame-by-frame window switching behavior for applause and aggregate numbers.

Objective Measure	Default	cost4	Delta
PEAQb	-1.5091	-1.394488	0.114612

Table 5.6: The ODG values measured with different versions of PEAQ for the applause item with optimized transform-switching.

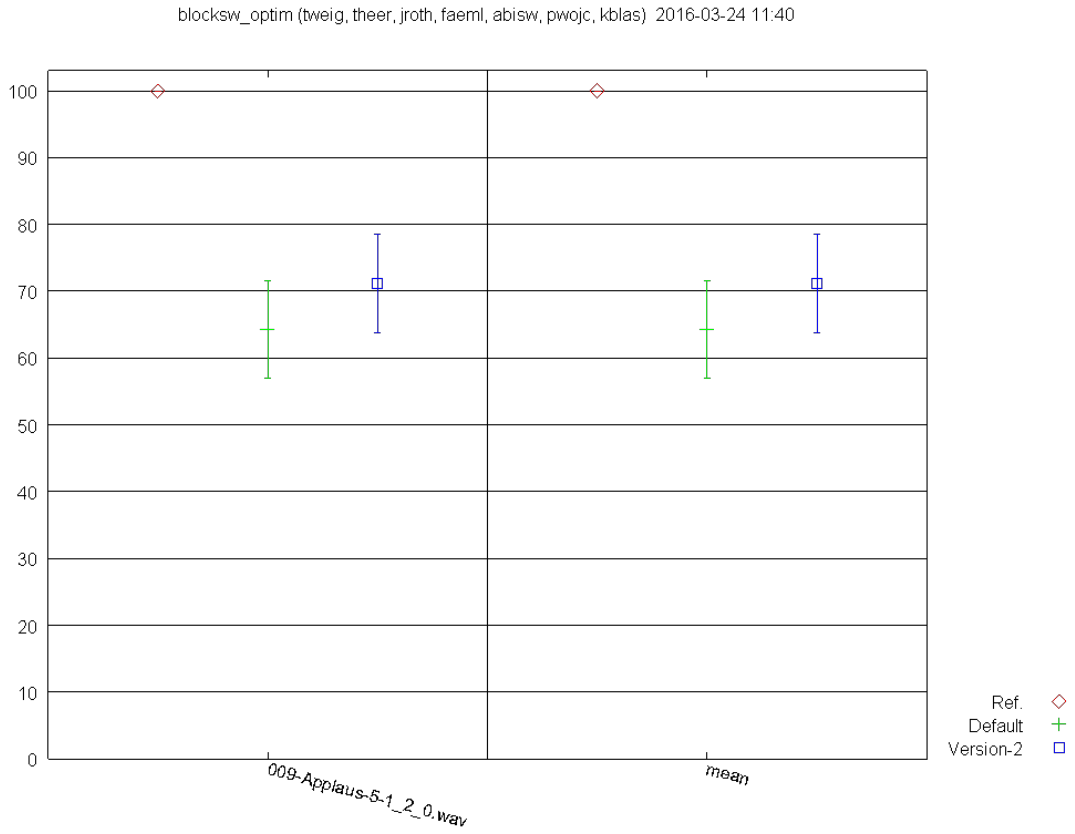

Figure 5.9: Listening test results for applause, taken by six expert listeners.

of the encoder for bitreservoir behaviour could be improved upon. As the bitreservoir has the biggest impact on the perceptual audio quality of transient rich signals, the castanets item was chosen as a single item to run the optimization with.

To influence the behavior, 10 different parameters were adjusted with a GA. The GA was run with a mutation probability (MUTPB) of 0.3, a crossover probability CXPB of 0.5, a population size of 150 and a total of 200 generations. It was possible it increase the amount of generations to 200, because only a single item was evaluated by PEAQ. A bitrate of 48 kb/s was used. The optimization was carried out with the castanets item and ODG values were acquired using PEAQb.

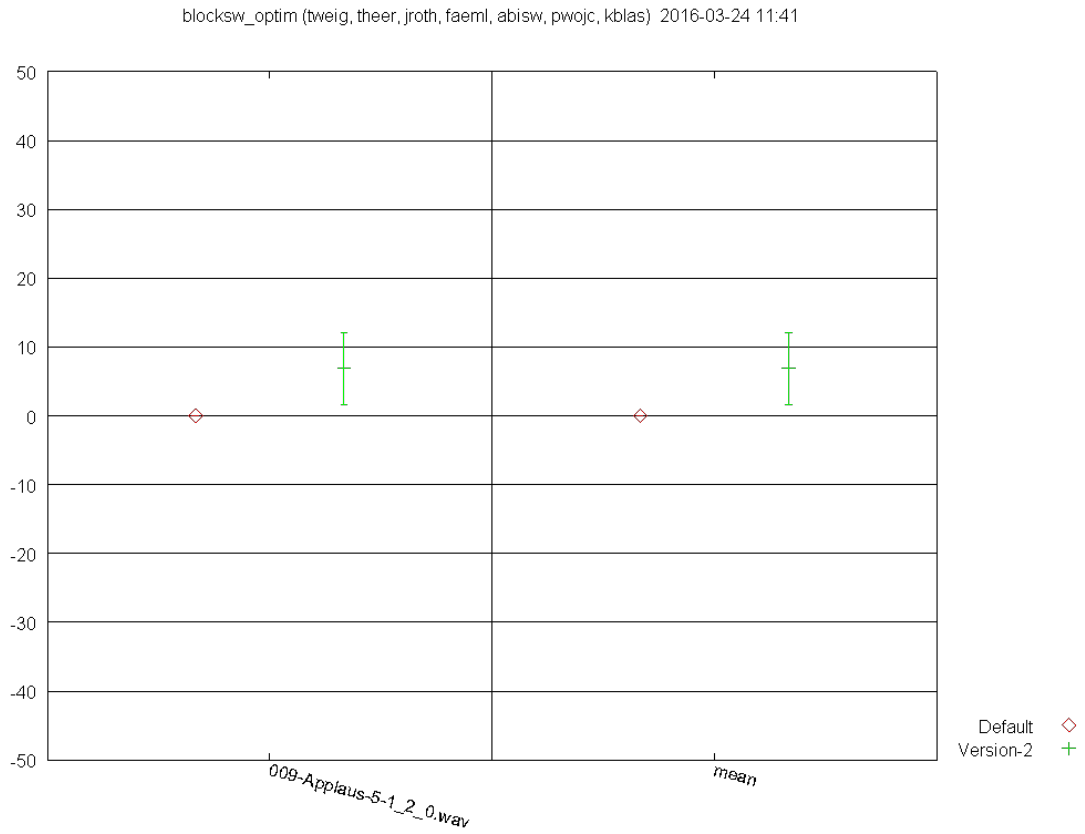

Figure 5.10: Differential listening test results for applause, taken by six expert listeners.

5.2.1 PEAQ Scores

Objective Measure	Default	Optimized	Delta
PEAQ BASIC	-2.5441	-2.5303	0.0138
PEAQ Advanced	-2.268587	-2.1673	0.101287
PEAQb	-2.814000	-2.840000	-0.026

Table 5.7: The ODG values measured with different versions of PEAQ for the castanets item with optimized bitreservoir control.

In addition to evaluation with PEAQb, the perceptual audio quality was assessed using the PEAQ Advanced and PEAQ Basic tools. As is seen in Table 5.7, no significant improvement was shown by PEAQ Basic, while the free PEAQb indicated slight degradation compared to the baseline default values. Even though PEAQ Advanced showed slight improvement over the default, informal listening probes showed no audible improvement and a formal listening test was omitted.

Chapter 6

Conclusion

6.1 Summary

The work of this thesis has shown that GAs can be used to tune complex perceptual audio coding systems successfully. Especially the results from optimizing window switching behavior show, that even with no guidance from audio encoder tuning experts GAs can find a meaningful operating point for a perceptual codec. Noteworthy in this context is the fact that all the parameters were initially chosen randomly by the GA and that no additional domain-specific knowledge was necessary for improving the performance of the Dolby AC-4 encoder.

The three part system that was build in order to carry out the optimization has proved flexible, efficient and maintainable. The different parts can be modified, updated and replaced independently of each other as long as the common communication interface is respected. Another beneficial property is that the system is robust to failure of individual parts. Failure of random elements of a distributed computer cluster is very common due to various reasons like maintenance of the machines, network communication errors or power failures. With the persistence and recovery capabilities of the system, such failures do not have a large impact on the overall progress of optimization; rather the effects are limited in time and gravity.

The results of the experiments which optimized the window-switching behavior, indicate that the current behavior of the AC-4 encoder is sub-optimal. The values found for the parameters that govern the window-switching behavior of the encoder, indicate that for the test bitrate, it is often beneficial for the audio quality to choose fewer and longer transform lengths.

In the end, audible improvement was achieved using the principles of GAs, which was shown in particular in the listening test experiments carried out over the ATSC3.0 and MPEG test data sets. The results obtained with PEAQ correlated sufficiently well with

the listening experiments. The parameter values, that were determined by the optimization algorithm were obtained by starting from random values, differ significantly from parameters determined by research. Still they produce equal or better audio quality. Clear audible improvement was also achieved in a single-item listening test for applause, which shows that audio encoder optimization using GAs can be also be used to optimize parameter settings for particular classes of audio.

6.2 Outlook

The concept of using GAs to optimize perceptual audio codecs still leaves a lot of room for potential improvements. In the context of the thesis, only the objective measurement tool PEAQ was used. In particular, only the aggregate ODG value for an excerpt was used, while the utility outputs the ODG values on a frame-to-frame basis. The additional model variables on a frame-by-frame basis, might be utilized in further work on the subject. Recently different algorithms that assess the perceptual degradation of audio have been developed, and could be integrated into the system, using more than one objective measurement tool. The evaluation of perceptual audio quality with more than one objective listener tool, could prevent errors that occur because an objective listener tool is calibrated badly. Moreover the system could also be directly fed with listening test data provided by expert listeners, in addition to the ODG scores of PEAQ. In the final stages of the evolution, this could be useful to facilitate the progress of the evolutionary optimization process, when the degradation differences fall below the tolerance level of what an objective measurement tool is able to distinguish.

The system also offers room for improvement. The fact that EAs process data in large batches, facilitates the application of massively parallel computing to the system. A parallel system can run at maximum performance only when there are enough trial solutions left to be evaluated. At the end of a batch, the amount of available computing resources is greater than the amount of data to be processed. Turning the batch-wise processing of data into a continuous process of evolution that can harness the information from each evaluated individual could boost the performance of the system.

The concept of applying the principles of GAs to the field of perceptual audio coding is promising. At the same time, the concrete choice of a machine learning algorithm for the Publisher is not predetermined, and can be substituted at any time with ease. This makes it easy to implement different machine learning algorithms on the same system. In the future, applications deep artificial neural networks could be used to search for optimal parameter constellations of encoder settings even more efficiently.

List of Figures

List of Tables

Bibliography

[ACD+04] Robert L Andersen, Brett G Crockett, Grant A Davidson, Mark F Davis, Louis D Fielder, Stephen C Turner, Mark S Vinton, and Phillip A Williams. Introduction to dolby digital plus, an enhancement to the dolby digital coding system. In *Audio Engineering Society Convention 117*. Audio Engineering Society, 2004.

[Bis06] Christopher M Bishop. Pattern recognition. *Machine Learning*, 2006.

[BKS08] Adil M Bagirov, Bülent Karasözen, and Meral Sezer. Discrete gradient method: derivative-free method for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 137(2):317–334, 2008.

[Bra99] Karlheinz Brandenburg. Mp3 and aac explained. In *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*. Audio Engineering Society, 1999.

[BS14] Rasmus Bro and Age K Smilde. Principal component analysis. *Analytical Methods*, 6(9):2812–2831, 2014.

[Cod07] Multichannel Audio Codecs. Ebu evaluations of multichannel audio codecs. 2007.

[dja15] django software foundation. Django. `http://www.djangoproject.org`, 2015.

[DLKK02] Martin Dietz, Lars Liljeryd, Kristofer Kjorling, and Oliver Kunz. Spectral band replication, a novel approach in audio coding. In *Audio Engineering Society Convention 112*. Audio Engineering Society, 2002.

[DWVTR04] Eric A Durant, Gregory H Wakefield, Dianne J Van Tasell, and Martin E Rickert. Efficient perceptual tuning of hearing aids with genetic algorithms. *Speech and Audio Processing, IEEE Transactions on*, 12(2):144–155, 2004.

[EHG05] Emad Elbeltagi, Tarek Hegazy, and Donald Grierson. Comparison among five evolutionary-based optimization algorithms. *Advanced engineering informatics*, 19(1):43–53, 2005.

[EU10] Peter Kabal Electrical and Computer Engineering McGill University. Audio File Programs and Routines package. `http://www-mmsp.ece.mcgill.ca/documents/downloads/afsp/`, 2010.

[Fle40] Harvey Fletcher. Auditory patterns. *Reviews of modern physics*, 12(1):47, 1940.

[FOW66] LJ Fogel, AJ Owens, and MJ Walsh. Artificial intelligence through simulated evolution john wiley. *New York*, 1966.

[FRG+12] Félix-Antoine Fortin, De Rainville, Marc-André Gardner Gardner, Marc Parizeau, Christian Gagné, et al. Deap: Evolutionary algorithms made easy. *The Journal of Machine Learning Research*, 13(1):2171–2175, 2012.

[GD91] David E Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*, 1:69–93, 1991.

[Gre60] David M Green. Psychoacoustics and detection theory. *The Journal of the Acoustical Society of America*, 32(10):1189–1203, 1960.

[Gre61] Donald D Greenwood. Critical bandwidth and the frequency coordinates of the basilar membrane. *The Journal of the Acoustical Society of America*, 33(10):1344–1356, 1961.

[Hel72] Rhona P Hellman. Asymmetry of masking between noise and tone. *Perception & Psychophysics*, 11(3):241–246, 1972.

[HNG94] Jeffrey Horn, Nicholas Nafpliotis, and David E Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 82–87. Ieee, 1994.

[Hol75] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[HZ09a] Martin Holters and Udo Zölzer. Automatic parameter optimization for a perceptual audio codec. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 13–16. IEEE, 2009.

[HZ09b] Martin Holters and Udo Zölzer. Automatic parameter optimization for a perceptual audio codec. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 13–16. IEEE, 2009.

[JM91] Cezary Z Janikow and Zbigniew Michalewicz. An experimental comparison of binary and floating point representations in genetic algorithms. In *ICGA*, pages 31–36, 1991.

[Joh88] James D Johnston. Estimation of perceptual entropy using noise masking criteria. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 2524–2527. IEEE, 1988.

[Koz91] John R Koza. Evolution and co-evolution of computer programs to control independently-acting agents. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats. MIT Press, Cambridge, MA*, pages 366–375, 1991.

[KP98] Ron Kohavi and Foster Provost. Glossary of terms. *Machine Learning*, 30(2-3):271–274, 1998.

[Kra12] Eugene F Krause. *Taxicab geometry: An adventure in non-Euclidean geometry*. Courier Corporation, 2012.

[KRW+16] K Kjörling, J Rödn, M Wolters, J Riedmiller, A Biswas, P Ekstrand, A Gröschel, P Hedelin, T Hirvonen, H Hrich, J Klejsa, J Koppens, K Krauss, H-M Lehtonen, K Linzmeier, H Muesch, H Mundt, S Norcross, J Popp, H Purnhagen, J Samuelsson, M Schug, L Sehlstrm, R Thesing, L Villemoes, and M Vinton. To be published at the aes convention paper ac-4 the next generation audio codec. In *AC-4 The Next Generation Audio Codec*, 2016.

[KV+83] Scott Kirkpatrick, Mario P Vecchi, et al. Optimization by simmulated annealing. *science*, 220(4598):671–680, 1983.

[LKDP11] Mikko-Ville Laitinen, Fabrian Kuech, Sascha Disch, and Ville Pulkki. Reproducing applause-type signals with directional audio coding. *Journal of the Audio Engineering Society*, 59(1/2):29–43, 2011.

[LMH+05] Tilman Liebchen, Takehiro Moriya, Noboru Harada, Yutaka Kamamoto, and Yuriy A Reznik. The mpeg-4 audio lossless coding (als) standard-technology and applications. In *AES 119th Convention paper*, 2005.

[Mar06] David Marston. Audio coding using a genetic algorithm. In *Audio Engineering Society Convention 120*. Audio Engineering Society, 2006.

[PJ95] John Princen and James D Johnston. Audio coding with signal adaptive filterbanks. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 5, pages 3071–3074. IEEE, 1995.

[Pos15] PostgreSQL. PostgreSQL. `http://www.postgresql.org`, 2015.

[PWO15] Stephan Preihs, Christoph Wacker, and Jorn Ostermann. Adaptive pre- and post-filtering for a subband adpcm-based low delay audio codec. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*, pages 1–5. IEEE, 2015.

[Rec98] ITUR Rec. Bs. 1387, method for objective measurements of perceived audio quality. *International Telecommunications Union, Geneva, Switzerland*, 1998.

[Rec03] ITURBS Recommendation. 1534-1: Method for the subjective assessment of intermediate quality level of coding systems. *International Telecommunication Union*, 2003.

[RFG+14] De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau, Christian Gagné, et al. Deap: enabling nimbler evolutions. *ACM SIGEVOlution*, 6(2):17–26, 2014.

[RHW85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[Sch70] Bertram Scharf. Critical bands. *Foundations of modern auditory theory*, 1:157–202, 1970.

[Sch81] Hans-Paul Schwefel. *Numerical optimization of computer models*. John Wiley & Sons, Inc., 1981.

[SPA06] Andreas Spanias, Ted Painter, and Venkatraman Atti. *Audio signal processing and coding*. John Wiley & Sons, 2006.

[TMKH96] Kit-Sang Tang, Kim-Fung Man, Sam Kwong, and Qun He. Genetic algorithms and their applications. *Signal Processing Magazine, IEEE*, 13(6):22–37, 1996.

[TTB+00] Thilo Thiede, William C Treurniet, Roland Bitto, Christian Schmidmer, Thomas Sporer, John G Beerends, and Catherine Colomes. Peaq-the itu standard for objective measurement of perceived audio quality. *Journal of the Audio Engineering Society*, 48(1/2):3–29, 2000.

[WD16] Vasant Pandian Gerhard-Wilhelm Weber and Vo Ngoc Dieu. *Handbook of Research on Modern Optimization Algorithms and Applications in Engineering and Economics*. IGI Global, 2016.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[WF05] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[WKHP03] Martin Wolters, Kristofer Kjorling, Daniel Homm, and Heiko Purnhagen. A closer look into mpeg-4 high efficiency aac. In *Audio Engineering Society Convention 115*. Audio Engineering Society, 2003.

[ZRRE65] J Zwislocki, Luce RD, Bush RR, and Galanter E. Analysis of some auditory characteristics. handbook of mathematical psychology, 1965.