Master thesis

Extending Evolutionary Approximation of Polyphonic Music via Multi-Objective Optimisation

Malte Schörken (460790)

Supervisors:
1st: Prof. Dr. Holger H. Hoos
2nd: Prof. Dr. Günter Rudolph
Daily supervisors: Dr. Igor Vatolkin and Justin Dettmer

RWTH Aachen University
Department of Computer Science
Chair of AI Methodology (Informatik 14)
https://www.aim.rwth-aachen.de/
Communicated by Prof. Dr. Holger H. Hoos

Abstract

Instrument and pitch detection in polyphonic audio recordings is a fundamental yet challenging problem in the field of musical information retrieval. State-of-the-art deep learning approaches require large annotated datasets and extensive training, whereas evolutionary algorithms (EAs) offer a data-independent alternative through direct signal approximation. Previous studies demonstrated that EAs are capable of approximating target recordings to detect pitches and instruments simultaneously. This work aims to improve the detection accuracy by introducing a multi-objective evolutionary algorithm (MOEA) that jointly optimises complementary fitness functions. A fitness function evaluates the similarity between the target signal and its synthesised approximation by combining an audio feature with a corresponding distance metric. Candidate fitness measures were first evaluated in the single-objective setting and groups of complementary objectives were identified using Shapley value analysis. The MOEA was evaluated on single chords and synthetic polyphonic pieces. It was compared to single-objective baselines using Jaccard distance, success rate and perceptual metrics. The resulting populations can further support applications ranging from music transcription and rearrangement to content-based music search and genre classification.

Table of Contents

List of Figures iii

List of Tables v

List of Algorithms vi

List of Abbreviations vii

1 Introduction 1

1.1 Goals . . . 2

1.2 Methodology . . . 2

2 Background 4

2.1 Music Theory . . . 4

2.1.1 Pitches . . . 4

2.1.2 Intervals . . . 5

2.1.3 Chords . . . 6

2.1.4 Onsets . . . 6

2.1.5 Timbre . . . 7

2.2 Evolutionary Algorithm . . . 7

2.2.1 Individuals . . . 9

2.2.2 Mutations . . . 10

2.2.3 Fitness Functions . . . 12

2.2.4 Single-Objective Algorithm . . . 15

2.2.5 Multi-Objective Algorithm . . . 16

2.3 Evaluation Metrics . . . 19

2.3.1 Standard Evaluation Metrics . . . 20

2.3.2 Chord Distances . . . 23

2.3.3 Instrument Distances . . . 27

2.3.4 MOEA Metrics 29
2.3.5 Shapley Value 31

3 Single-Objective Experiment **35**
3.1 Fitness Function Selection 37

4 MOEA Experiments **41**
4.1 Single Onset Approximation 41
4.2 Full Piece Approximation 45

5 Discussion **49**

6 Conclusions and Future Work **51**

References **53**

A Statistics about the Sample Library **57**

B Instrument-Style Distances **59**

C Single-Objective Experiment Full Results **61**

D Audio Feature Correlation **62**

E Statistics about the Tiny AAM Dataset **63**

F Approximation Results by Piece **64**

List of Figures

2.1 Section of piano keyboard with keys ranging from C3 to C5. (Source: Müller, 2015, p. 4) . . . 5

2.2 Illustration of attack, transient, onset and decay of a single note (a) Note played on a piano. (b) Idealized amplitude envelope. (Source: Müller, 2015, p. 311) . . . 7

2.3 Overview of the evolutionary algorithm. . . . 9

2.4 Calculation of the fitness value. . . . 12

2.5 Evolutionary loop with 3 individuals. STFT + COSH fitness value in green. . . . 17

2.6 Crowding-distance calculation. Points marked in filled circles are solutions of the same non-dominated front. (Source: Deb et al. (2002)) . . . 18

2.7 Hypervolume contributions from the third front of a population in a two-dimensional target space (left) and an example of a three-dimensional target space (right). (Source: Beume et al. (2007)) . . . 19

2.8 MDS projection of instrument distances into two-dimensional space. . . . 28

3.1 Fitness function success rate. . . . 36

3.2 STFT + KL-divergence: success rate versus sample number. . . . 36

3.3 Scatter matrix of fitness functions using instrument distances (Jaccard and McAdams) on the y-axis and chord distances (Jaccard, average Hausdorff Ginsel, Average Hausdorff and L_2^{88}) on the x-axis. . . . 37

3.4 Instrument-level versus pitch-level Jaccard distance for fitness functions with $J_i \leq 0.5$ and $J_p \leq 0.5$ . . . 38

3.5 Fitness function success rate in comparison to STFT + KL-divergence. . . . 39

3.6 Shapley values of the fitness function. . . . 39

4.1 SMS-EMOA fitness convergence for 5 objectives on a logarithmic scale. . . . 42

4.2 Average success rate across all 100 onsets for NSGA-II and SMS-EMOA . . . 43

4.3 Average GD across all pieces for the single-objective evolutionary algorithm (SOEA), NSGA-II and SMS-EMOA . . . 46

4.4 Average IGD across all pieces for the single-objective evolutionary algorithm (SOEA), NSGA-II and SMS-EMOA. . . . 47

4.5 Diversity by number of objectives. . . . 48

B.1 MDS projection of instrument distances for drums, flute and violin with all styles. . 59
B.2 MDS projection of instrument distances for violin, viola and cello with all styles. . . 60

C.1 Success matrix for all 100 fitness functions and all 100 chords. 61

D.1 Correlation matrix of audio features. 62

F.1 Average GD by piece and algorithm. 64
F.2 Average IGD by piece and algorithm. 64

List of Tables

2.1 Intervals within one octave in the equal temperament system. 5
2.2 Overview of the evolutionary algorithm parameters and their default values. 16
2.3 Instrument styles for exemplary fitness calculations. 16
2.4 Interval-vector representation of a C-Major chord. 23
2.5 Vector representation of a C-Major chord. 23
2.6 Chord distances to C-Major using various metrics 26
2.7 Chord distances to C-Major including Ginsel distance 27
2.8 Successes of the three fitness functions. 33

3.1 Distribution of the number of samples in the single onset dataset (mean = 2.71). 35
3.2 Top 10 fitness functions by success rate. 40
3.3 Top 10 fitness functions by Shapley value. 40

4.1 Hypothesis test results comparing success rate against single-objective baseline (0.7). 44
4.2 GD: Hypothesis test results comparing MOEA against single-objective baseline . . . 46
4.3 IGD: Hypothesis test results comparing MOEA against single-objective baseline . . 47

A.1 Instrument ranges and styles 58

E.1 Overview of dataset statistics. 63

List of Algorithms

1 Pseudocode of an evolution strategy, as presented by Bartz-Beielstein et al. (2014) . . 9
2 NSGA-II environmental selection, based on Deb et al. (2002) 18
3 SMS-EMOA environmental selection, based on Beume et al. (2007) 20

List of Abbreviations

AAM Artificial Audio Multitracks. 3, 41, 45, 51

AI Artificial Intelligence. 1

CNN Convolutional Neural Network. 1

CQT Constant-Q Transform. 13, 38–40, 45, 47

EA Evolutionary Algorithm. 1, 2, 4, 9, 10, 15, 17, 19, 20, 27, 29, 32, 35, 42, 43, 45, 49–51

EP Evolutionary Programming. 8

ES Evolution Strategy. 8, 15

FN False Negatives. 21

FP False Positives. 21

GA Genetic Algorithm. 8

GD Generational Distance. 29–31, 45–48, 50, 51

GP Genetic Programming. 8

IGD Inverted Generational Distance. 29–31, 45, 47, 48, 50, 51

IIRT Infinite Impulse Response Transform. 13, 38–40, 42, 45

LSTM Long Short-Term Memory. 1

MFCC Mel Frequency Cepstral Coefficients. 12, 35, 38, 49

MIR Musical Information Retrieval. 1, 2

MOEA Multi-Objective Evolutionary Algorithm. 2, 3, 16, 19, 20, 29, 31, 37–39, 41, 45, 46, 48–52

RNN Recurrent Neural Network. 1

STFT Short-Time Fourier Transform. 12, 16, 26, 27, 35, 37–42, 45, 49, 51

TP True Positives. 21

VQT Variable-Q Transform. 13, 38

Chapter 1

Introduction

In recent years, the music industry has undergone rapid transformation. In 2014 music streaming accounted for only 14 % of global recorded music revenues, which had declined to a historic low of US$12.9 billion. However, over the past decade global recorded music revenues have steadily increased, reaching US$29.6 billion in 2024. This growth has been largely driven by the rapid expansion of streaming, which now accounts for 69 % of total recorded music revenue (International Federation of the Phonographic Industry, 2025).

Major streaming platforms such as Spotify, Apple Music and Amazon Music offer a variety of features, including real-time recommendations, automatic playlist generation, advanced music discovery and user-driven melody search. A key capability underlying many of these services is the application of Artificial Intelligence (AI) to extract meaningful musical information from raw audio (Mokoena & Obagbuwa, 2025). This task is studied in the field of Musical Information Retrieval (MIR), an interdisciplinary research field that focuses on extracting and analysing information from music data in order to make large collections of music searchable and accessible (Burgoyne et al., 2015; Downie, 2004).

Instrument and pitch detection in polyphonic audio recordings is a fundamental yet challenging problem in the field of MIR. Accurate identification of simultaneous pitches and instruments enables not only music transcription and rearrangement but also applications such as content-based music search and genre classification (Dettmer et al., 2024; Han et al., 2017; Vatolkin, 2020). Factors that make this task particularly challenging include varying overtone distributions, non-harmonic frequencies, temporal dynamics (attack, decay, sustain, release phases), differences in playing styles, loudness variations and effects such as reverb and distortion (Benetos et al., 2019; Vatolkin, 2020).

Deep neural networks are widely used models for instrument and pitch detection (Benetos et al., 2019; Bhattarai & Lee, 2023). Architectures that achieve state-of-the-art performances include Convolutional Neural Networks (CNNs) (Dash et al., 2024) and Recurrent Neural Networks (RNNs) using Long Short-Term Memory (LSTM) units (Elghamrawy & Ibrahim, 2021). Recent work has shown that transformer-based architectures (Koutini et al., 2022), including encoder–decoder models (Chang et al., 2024) and CNN–Transformer hybrids (Cheuk et al., 2022), demonstrate competitive performance for polyphonic instrument detection. A key limitation of supervised approaches is their reliance on extensive labelled datasets (Benetos et al., 2019). In MIR, however, such annotated data is limited.

As an alternative, Evolutionary Algorithms (EAs) have shown promising results in tasks with extremely large, multimodal or poorly understood search spaces (Das et al., 2011; Tian et al., 2021). EAs are a class of stochastic optimisation techniques inspired by Darwinian evolution. A population of candidate solutions is evolved over successive generations using operators such as selection, mutation and sometimes recombination. Each solution is evaluated by a fitness function,

which measures its quality with respect to the task at hand. Solutions with better fitness are more likely to contribute to the next generation, allowing the population to gradually converge towards optima. EAs are particularly well suited to problems where gradient-based methods are ineffective or labelled data is scarce (Sloss & Gustafson, 2020).

In the context of MIR, Vatolkin (2020) introduced an evolutionary synthesis approach that jointly detects pitches and instruments by approximating a target audio signal. This approach has been extended in subsequent work, including two master's theses: Ginsel (2021) compared several audio features and distance measures to optimise the fitness function, while Dettmer (2023) introduced a weighted initialisation strategy to seed the population with likely pitches. However, these implementations optimise only a single-objective. Despite effective optimisation of the fitness function, detection accuracy remains limited. This motivates the transition to a Multi-Objective Evolutionary Algorithm (MOEA). MOEAs have shown success in other MIR tasks such as melody harmonisation (Freitas & Guimaraes, 2011; Majidi & Toroghi, 2023), melody generation (de León et al., 2016; Jeong & Ahn, 2015) and feature selection for instrument recognition tasks (Vatolkin et al., 2012).

To select suitable objectives for the MOEA, we use Shapley values. Shapley values were first introduced by Shapley (1951) in the context of game theory, but are now a fundamental concept in data science (Qin et al., 2025). In the context of MOEAs, Wu (2021) used Shapley values to balance the weight of the objectives by their respective contribution.

1.1 Goals

The overarching goal of this thesis is to improve the accuracy of instrument and pitch detection in polyphonic audio through the use of multi-objective optimisation. In particular, this work investigates whether MOEAs can outperform existing single-objective evolutionary approaches in approximating polyphonic chords and artificially generated musical pieces.

A first objective is to define evaluation metrics that better reflect human auditory perception. Traditional measures such as the Jaccard distance treat all errors equally, regardless of their perceptual severity. In contrast, the proposed metrics aim to account for similarity relationships between instruments and pitches. For example, confusing a violin with a viola should be penalised less than confusing a violin with drums and predicting a neighbouring pitch should be considered less severe than predicting a distant one. These perceptually informed metrics will serve as the basis for evaluating all subsequent experiments.

A second objective is to systematically analyse and compare different fitness functions for evolutionary audio approximation. Building on previous work, combinations of audio features and distance measures are evaluated to identify functions that are particularly effective for pitch and instrument detection. These functions are then used as candidate objectives in the multi-objective setting.

The final objective is to extend the existing evolutionary framework to a multi-objective formulation. Different strategies for selecting objective functions, including success-rate-based, Shapley-value-based and random selection, are investigated. The performance of established MOEAs, specifically SMS-EMOA and NSGA-II, is evaluated on single chords and artificially generated musical pieces and compared against a single-objective baseline.

1.2 Methodology

This thesis follows an experimental approach to evaluate the effectiveness of multi-objective evolutionary algorithms for instrument and pitch approximation. The experiments are structured into two main stages:

1. Analysis and selection of candidate fitness functions (Chapter 3).
2. Evaluating the performance of the MOEA against the single-objective baseline (Chapter 4).

In the first stage, a large set of candidate fitness functions is systematically evaluated in a single-objective setting. These fitness functions are combinations of audio features and distance measures. The evaluation is conducted on a single-onset dataset consisting of short polyphonic audio signals, allowing controlled analysis of approximation quality.

Based on the results of this analysis, subsets of fitness functions are selected to form multi-objective optimisation problems. Different selection strategies are considered, including recall-based selection, Shapley value-based selection and random selection.

In the second stage, established MOEAs, namely SMS-EMOA (Beume et al., 2007) and NSGA-II (Deb et al., 2002), are applied to the task of audio approximation. Their performance is evaluated on a single-onset dataset and the Artificial Audio Multitracks (AAM) dataset (Ostermann et al., 2023). The AAM dataset contains synthetically generated polyphonic audio pieces with precise annotations for onset times, instruments and pitches.

To assess the quality of the results, both traditional metrics and newly proposed perceptually motivated metrics are used. This enables a comprehensive evaluation of whether multi-objective optimisation leads to improvements that are not only numerically significant but also perceptually meaningful.

Chapter 2

Background

This chapter introduces the fundamental concepts required for this work. As the proposed approach operates on musical data using EAs, it combines elements from both music theory and optimization.

We begin with a brief overview of basic music theory, including pitches, intervals and chords. Subsequently, we introduce evolutionary algorithms as the central methodological framework of this work. We describe their core components, including individuals, mutation, selection and fitness evaluation. Finally, we present the evaluation metrics used to assess the quality of generated solutions.

2.1 Music Theory

To work with music computationally, it is useful to formalise the fundamental concepts of music theory. This section introduces the core building blocks of Western music: pitches, intervals and chords. Afterwards, the concepts of onsets and timbre are briefly introduced capturing temporal and perceptual dimensions of music beyond harmony.

2.1.1 Pitches

Pitch is defined as the perceptual attribute of a sound that allows it to be ordered on a scale from low to high. It is closely related to the frequency content of the acoustic signal, although it is ultimately a perceptual quantity rather than a purely physical one (Weihs et al., 2016, p. 16).

This perceptual property enables a linear ordering of tones, forming the basis for tonal systems in Western music. Pitches are commonly organized using the 12-tone equal temperament system. In this system, the interval between two tones where one has twice the frequency of the other is called an octave, which is divided into twelve equal steps called semitones. Each semitone corresponds to a constant multiplicative change in frequency. Specifically, moving up by one semitone multiplies the frequency by $2^{1/12}$. After twelve such steps, the frequency doubles, completing the octave.

A standard reference pitch is A4, which is defined as 440 Hz. Based on this reference, the frequency of any pitch can be calculated depending on its distance in semitones. For a pitch that is k semitones higher than A4, the frequency is given by

$$f = 440 \cdot 2^{k/12}. \tag{2.1}$$

This formulation allows pitches to be represented numerically, which is particularly useful in computational applications. Although pitch is physically determined by frequency, it is often treated as a discrete symbolic label in algorithmic contexts.

In Western music notation, pitch classes are represented using letter names combined with accidentals. Seven of them correspond to the natural notes C, D, E, F, G, A and B, which are typically associated with the white keys of a piano keyboard (Figure 2.1). The remaining five pitch classes correspond to the black keys and are represented using sharps or flats. A sharp raises a pitch by one semitone, while a flat lowers it by one semitone. For example, C# and D♭ denote the same pitch class in the equal-tempered system, even though they are distinguished in musical notation depending on harmonic context (Müller, 2015, p. 4).

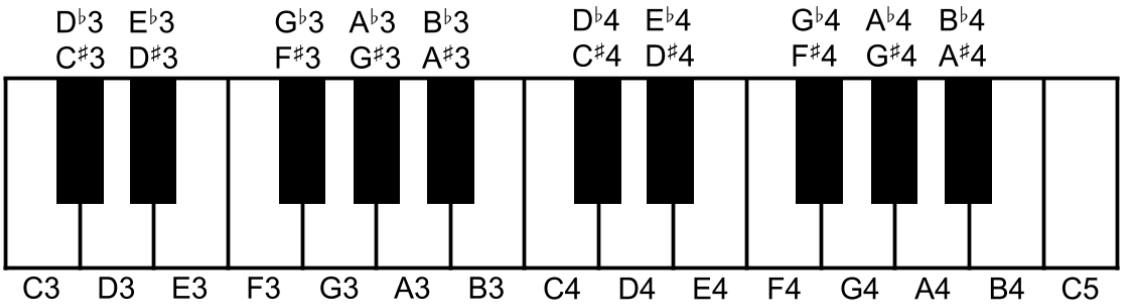


Figure 2.1: Section of piano keyboard with keys ranging from C3 to C5. (Source: Müller, 2015, p. 4)

2.1.2 Intervals

An interval describes the distance between two pitches. Perceptually, intervals correspond to differences in pitch height, while acoustically they can be described by ratios between the frequencies of the corresponding tones (Weihs et al., 2016, p. 22). In the equal temperament system, intervals are measured in semitones, i.e., the number of discrete steps between two pitches.

Table 2.1 lists the intervals within one octave, measured relative to a root pitch. The naming of the intervals follows Weihs et al. (2016), with additional labels for intervals that are commonly referred to by multiple names in the literature (e.g., Kostka et al. 2018).

Table 2.1: Intervals within one octave in the equal temperament system.

Semitones	Interval Name	Frequency Ratio
0	Pure prime / Perfect unison	1
1	Minor second	$2^{1/12}$
2	Major second	$2^{2/12}$
3	Minor third	$2^{3/12}$
4	Major third	$2^{4/12}$
5	Pure fourth / Perfect fourth	$2^{5/12}$
6	Augmented fourth / Tritone / Diminished fifth	$2^{6/12}$
7	Pure fifth / Perfect fifth	$2^{7/12}$
8	Minor sixth / Augmented fifth	$2^{8/12}$
9	Major sixth	$2^{9/12}$
10	Minor seventh	$2^{10/12}$
11	Major seventh	$2^{11/12}$
12	Pure octave / Perfect octave	2

Intervals play a central role in musical perception. Some intervals, such as the octave or perfect fifth, are generally perceived as consonant (stable), while others, such as the minor second or tritone, are considered dissonant (tense).

This perceptual distinction is related to the ratio between the frequencies of the two pitches. Intervals with simple integer ratios (e.g., $2:1$ for the octave or $3:2$ for the perfect fifth) tend to sound more consonant. In the equal temperament system, the octave retains its exact $2:1$ ratio, while other intervals are approximated by powers of $2^{1/12}$. For example, the perfect fifth having an ideal ratio of $\frac{3}{2} = 1.5$ is approximated by

$$2^{7/12} \approx 1.4983, \tag{2.2}$$

This allows instruments to play in all keys with consistent tuning, at the cost of slight deviations from these ideal ratios.

2.1.3 Chords

In Western music, harmony refers to the simultaneous sounding of multiple notes forming a perceptually coherent entity. The primary building blocks of harmony are chords, which are typically defined as collections of three or more pitches (Müller, 2015, p. 241).

Chords are typically constructed by stacking intervals on top of a root note. The most common class of chords are triads, which consist of three pitches: the root, the third and the fifth. A major triad is formed by combining a root with a major third (4 semitones) and a perfect fifth (7 semitones). A minor triad consists of a root, a minor third (3 semitones) and a perfect fifth. Other important triads include the diminished triad, which contains a minor third and a diminished fifth (6 semitones) and the augmented triad, which contains a major third and an augmented fifth (8 semitones).

More complex chords can be constructed by extending triads with additional intervals. A common example is the seventh chord, which adds a seventh interval to the triad. For instance, a major seventh chord consists of a major triad plus a major seventh (11 semitones), while a dominant seventh chord combines a major triad with a minor seventh (10 semitones).

Chords are not restricted to a single ordering of pitches. If a note other than the root is the lowest pitch, the chord is said to be inverted. For example, in a triad, placing the third or the fifth in the bass yields the first or second inversion, respectively. Inversions are musically equivalent in terms of pitch content but differ in their harmonic function and sound.

In this work, chords are often represented as unordered sets or vectors of pitch classes, allowing for efficient comparison (see Section 2.3.2). This representation naturally accommodates simplified chord definitions, including dyads (two-note chords) and single-note events.

2.1.4 Onsets

Besides pitch-related information, the temporal structure of music is of central importance. A key concept in this context is the onset, which denotes the time point at which a musical event begins.

In acoustic signals, the onset is typically associated with the beginning of a transient, i.e., a short and often noise-like signal component characterized by a rapid increase in energy (Müller, 2015, p. 311). Following the onset, the sound enters the attack phase, during which the amplitude increases until it reaches its peak. This is followed by a decay phase, in which the amplitude decreases until a more stable level is reached.

It is important to distinguish these concepts: while the attack and decay describe temporal intervals of the sound's evolution, the onset refers to a single time point marking the beginning of the event. Figure 2.2 illustrates these components for a musical note. In symbolic music representations,

onsets correspond to note start times and, together with durations, define the temporal structure of a piece.

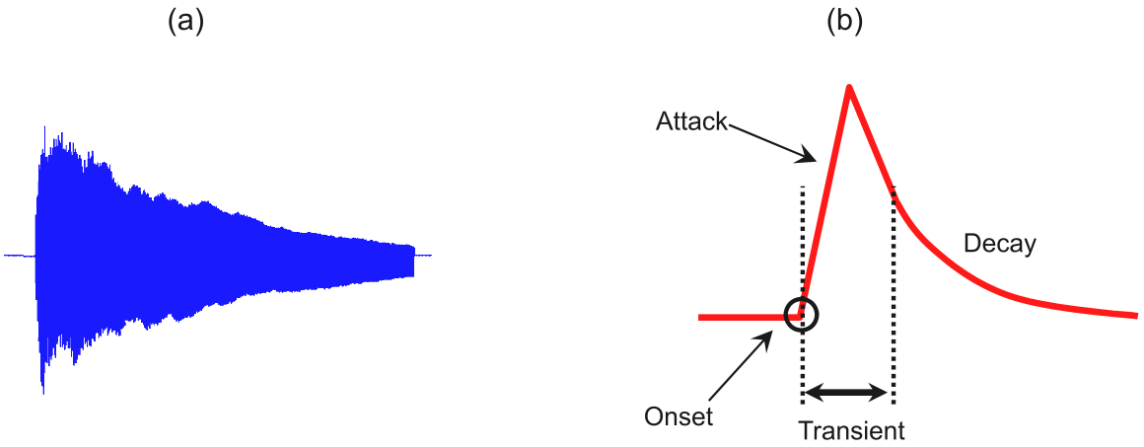


Figure 2.2: Illustration of attack, transient, onset and decay of a single note (a) Note played on a piano. (b) Idealized amplitude envelope. (Source: Müller, 2015, p. 311)

2.1.5 Timbre

In addition to pitch and loudness, timbre (or tonal colour) is a fundamental perceptual attribute of sound. It describes the quality that allows listeners to distinguish between sounds that share the same pitch, loudness and duration (Weihs et al., 2016, p. 50).

From a perceptual perspective, timbre is a multidimensional construct rather than a single measurable quantity. It has been shown to be strongly influenced by both spectral and temporal properties of sound, with no unique physical correlate that fully captures it (McAdams, 2013).

Spectrally, timbre depends on the distribution of energy across frequencies, including harmonic structure, spectral envelope and inharmonic components. Temporally, it is shaped by the evolution of sound over time, in particular the attack phase and transient behaviour. The onset region has been identified as especially important for timbre perception and instrument identification (Müller, 2015, pp. 26–30).

Importantly, timbre is not solely determined by the nominal instrument category (e.g. piano or violin). Instead, it arises from the underlying sound production mechanism. Different physical models of the same instrument, such as different piano models or synthesis methods, can produce perceptually distinct timbres even when playing the same note. Likewise, different playing techniques-such as bowing position on a string instrument or articulation style in wind instruments can significantly alter the resulting timbre.

2.2 Evolutionary Algorithm

The concept of evolution was fundamentally shaped by Charles Darwin's book "On the Origin of Species" (Darwin, 1859). Evolution describes the change in the genetic composition of a population over generations and is driven by variation, inheritance and selection. Variation arises through mutations and recombination. Mutation introduces random changes in genetic material and recombination reshuffles existing genetic material during reproduction to create new combinations of traits. The differences in genetic material are reflected in phenotypes, the observable characteristics of organisms.

Selective pressure describes environmental or biological factors that influence which phenotypes are advantageous in a given context, such as climate, food availability, predators, or mating preferences in sexual selection. These pressures affect the fitness of individuals, meaning their ability to survive and reproduce. Individuals whose phenotypes are better suited to the prevailing selective pressures tend to have higher fitness. As a result, they are more likely to survive and pass their traits on to the next generation. This process is known as natural selection. Over time, it leads to an increase in the frequency of advantageous traits and thus a gradual shift in the genetic composition of the population.

Different forms of selection describe how selective pressure acts on traits. In directional selection, one extreme of a trait is favoured. A well-known example is the peppered moth during the Industrial Revolution (Cook, 2003). Originally, most moths were light-coloured, which helped them blend in with trees and avoid predators. Due to industrial pollution, trees became darker. As a result, dark-coloured moths were better camouflaged, had a higher chance of survival and reproduced more frequently. This caused the population to shift towards the darker colour phenotype.

In stabilizing selection, intermediate traits are favoured, while extreme variations are selected against. A common example is the sexual selection of the male peacock tail (Petrie & Halliday, 1994). Males with larger and more colourful tails are more attractive to females which leads to increased reproduction. However, excessively large tails can reduce mobility and increase predation risk. This creates a balance where an intermediate tail size yields the highest overall fitness.

In diversifying selection, both extremes of a trait are favoured over intermediate forms. An example is the African finch Pyrenestes (Smith, 1993). These birds feed on either very hard or very soft seeds. Individuals with large beaks can crack hard seeds efficiently, while those with small beaks are better suited for soft seeds. Birds with intermediate beak sizes are less efficient at handling either type of food, resulting in lower fitness. Consequently, both extremes are favoured and the population splits into distinct groups.

We can apply the concepts of biological evolution by formulating an optimisation problem as

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{subject to} && x \in \mathcal{X}, \end{aligned} \tag{2.3}$$

where the objective function $f : \mathcal{X} \to \mathbb{R}$ is referred to as the fitness function. Further, x denotes an individual and $\mathcal{X}$ denotes the set of all feasible individuals. For our specific context of approximating music, the set $\mathcal{X}$ is defined in Equation 2.5. In an evolutionary algorithm, the biological evolution process is simulated with the goal of optimising the fitness function.

There are four main families of evolutionary algorithms: Genetic Algorithms (GAs), Evolutionary Strategies (ESs), Evolutionary Programming (EP) and Genetic Programming (GP). These approaches primarily differ in their representation of individuals and the variation operators applied during evolution. GAs typically operate on fixed-length encoded representations and emphasize recombination through crossover. These algorithms mainly focus on mating selection. Evolutionary Strategies (ESs) are designed for continuous optimization problems and primarily rely on mutation and environmental selection. EP originally focused on the evolution of behavioural models such as finite-state machines and is mainly mutation-driven. In contrast, GP evolves tree-structured representations, typically corresponding to executable programs or expressions (Bartz-Beielstein et al., 2014).

ESs were originally introduced by Rechenberg (1971) and further developed by Schwefel (1975), who formalized key components such as self-adaptive mutation and selection schemes. They form the basis for the evolutionary algorithm used for pitch and instrument approximation. An overview of this evolutionary process is illustrated in Figure 2.3.

Algorithm 1 consists of the key building blocks of evolution: mating selection, recombination, mutation and environmental selection. Together, these steps form the evolutionary loop, where

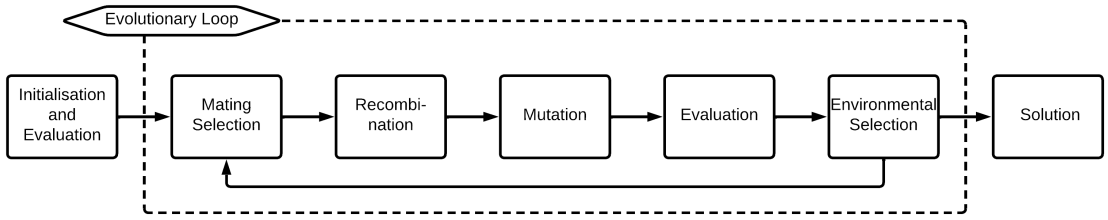


Figure 2.3: Overview of the evolutionary algorithm.

Algorithm 1: Pseudocode of an evolution strategy, as presented by Bartz-Beielstein et al. (2014)

Input: Number of offspring λ, population size μ, fitness function f
Output: Final population P

```
Initialise a population P of μ individuals;
Evaluate the fitness of all individuals in P;
while termination condition not met do
    Select parents from P ;                          // Mating selection
    Generate λ offspring from the selected parents ;    // Recombination
    Mutate the offspring ;                                // Mutation
    Evaluate the offspring ;                            // Evaluation
    Set P = P ∪ {offspring};
    Remove λ individuals from P ;              // Environmental selection
return P;
```

each iteration is considered a generation. A common termination condition is a fixed number of generations. Alternatively, a threshold for the fitness value of the best individual can be defined, such that the algorithm terminates once a sufficiently good solution is found.

2.2.1 Individuals

Individuals are a core building block of the EA. In our setting, the individuals are chords consisting of one or more samples. Each sample is drawn from a library of 6826 audio recordings each corresponding to a single instruments playing exactly one note. The library is compiled from three sources: the library used by Vatolkin et al. (2012), the Ethno World Complete 5 sample collection[1] and the Native Instruments Komplete Ultimate 11 collection[2]. The set of all 51 instruments Table A.1 in the library is denoted by $\mathcal{I}$. For some instruments, several recordings exist that differ with respect to the instrument model, playing styles, or recording source. For a given instrument $I \in \mathcal{I}$ the set of these different recordings is referred to as the set of styles and denoted by $\mathcal{S}_I$. For each combination of instruments $I \in \mathcal{I}$ and style $S \in \mathcal{S}_I$ there exist recordings of different pitches. The corresponding set of pitches is denoted by $\mathcal{P}_{I,S}$.

Using this notation a sample k is represented as a tuple of instrument, style and pitch. The set of all samples is defined as

$$\mathcal{K} = \{(I, S, P) \mid I \in \mathcal{I},\ S \in \mathcal{S}_I,\ P \in \mathcal{P}_{I,S}\} \tag{2.4}$$

[1] Version 5 is no longer available for purchase. The current version is available here: https://www.bestservice.com/en/ethno_world_7_instruments.html (Visited 02/02/2026)

[2] Version 11 is no longer available for purchase. The current version is available here: https://www.native-instruments.com/en/products/komplete/bundles/komplete-15-ultimate/ (Visited 02/02/2026)

with $|\mathcal{K}| = 6826$.

Finally, an individual x is represented by a multiset of samples, since a sample may occur multiple times and the order of elements does not affect the resulting sound. The set of all individuals is defined as

$$\mathcal{X} = \left\{ x : \mathcal{K} \to \mathbb{N}_0 \;\middle|\; 1 \leq \sum_{k \in \mathcal{K}} x(k) \leq N \right\} \tag{2.5}$$

where each individual x assigns a non-negative integer multiplicity to each sample $k \in \mathcal{K}$. In this notation N is the upper bound on the number of samples contained in one individual. The experiments in this thesis use $N = 5$, as in Dettmer (2023). In this case the number of all possible individuals is given by

$$|\mathcal{X}| = \sum_{n=1}^{N} \binom{|\mathcal{K}| + n - 1}{n} = \sum_{n=1}^{5} \binom{6826 + n - 1}{n} \approx 1.24 \times 10^{17} \tag{2.6}$$

which makes exhaustive search infeasible. This is further motivation for the use of EAs which can deal with extremely large search spaces.

For example, an individual with

- $k_1 = (\text{Violin}, \text{Violins1_Essential}, \text{C5})$, $x(k_1) = 1$
- $k_2 = (\text{Piano}, \text{The_Grandeur}, \text{C4})$, $x(k_2) = 2$

as well as $x(k) = 0$ for all other $k \in \mathcal{K}$ takes the following form:

$$x = \{(\text{Violin}, \text{Violins1_Essential}, \text{C5}), (\text{Piano}, \text{The_Grandeur}, \text{C4}), (\text{Piano}, \text{The_Grandeur}, \text{C4})\}$$

2.2.2 Mutations

Based on the individual representation introduced in Section 2.2.1, we consider three aspects that may be mutated:

1. The number of samples
2. The instrument-style tuple of a sample
3. The pitch of a sample

We treat instrument and style as a combined entity, since not all styles of an instrument necessarily cover the same pitch range. Furthermore, the number of available styles varies significantly between instruments. Mutating only the style could therefore lead to inconsistent behaviour, or have no effect in cases where only a single style exists. For this reason, both attributes are always modified jointly.

Mutation of the number of samples

To mutate the number of samples in an individual, we define a set of thresholds that depend on the current number of samples. A random number $r \in [0, 1)$ is drawn from a uniform distribution and compared to the corresponding threshold. If r is smaller than the threshold, a new sample is added; otherwise, an existing sample is removed uniformly at random.

For a maximum number of samples max_samples $= 5$, the thresholds are given by

$$[1,\ 0.8,\ 0.4,\ 0.1,\ 0],$$

following the work of Dettmer (2023). These values can be interpreted as the probability of increasing the number of samples, conditioned on the current size of the individual. In particular, the upper bound is enforced by assigning probability zero to further increases once the maximum is reached.

When adding a new sample, an instrument is drawn uniformly from the available sample library. Subsequently, a valid style for this instrument is selected, followed by a pitch that is compatible with the chosen instrument-style combination.

Mutation of the instrument-style tuple

For this mutation, a sample (I, S, P) is selected uniformly at random from the individual. Given the pitch P, we determine the set of instruments that provide at least one sample at this pitch. This restriction ensures that the resulting tuple remains valid, as different instruments may support different pitch ranges.

An instrument is then sampled uniformly from this set, after which a compatible style is selected uniformly from the available styles of that instrument. The pitch remains unchanged and the original tuple is replaced by the newly constructed one. Sampling instruments and styles separately ensures that all instruments are equally likely to be selected, independent of the number of styles they provide.

Pitch mutation

In contrast to the previous mutation, the pitch is not redrawn independently but modified relative to its current value. Let P_{old} denote the original pitch. The new pitch is computed as

$$P_{\text{new}} = \lfloor P_{\text{old}} + p_{\text{shift}} \rfloor , \tag{2.7}$$

where p_{shift} is sampled from a Gaussian distribution with mean μ_p and standard deviation σ_p. In our experiments, we use $\mu_p = 0$ and $\sigma_p = 15$, following Dettmer, 2023.

Number and selection of mutations

To determine how many mutations are applied to an individual, we draw a value G from a standard normal distribution $\mathcal{N}(0, 1)$ and compute

$$n_{\text{mut}} = \lfloor G \cdot \alpha + \beta \rfloor , \tag{2.8}$$

where α and β are hyperparameters. The resulting value is clipped to the range of $[l_{bound}, u_{bound}]$, where l_{bound} denotes the lower bound and u_{bound} denotes the upper bound on the number of applied mutations. The mutation strength is lowered over time by multiplying u_{bound} with a hyperparameter ζ in every iteration

For each mutation step, one of the three mutation types is selected probabilistically. We assign a probability of 20% to mutating the number of samples and 40% each to the instrument-style mutation and the pitch mutation, following Dettmer (2023).

2.2.3 Fitness Functions

The fitness is used to evaluate the similarity between a candidate solution and the target signal. An overview of the complete computation pipeline is shown in Figure 2.4.

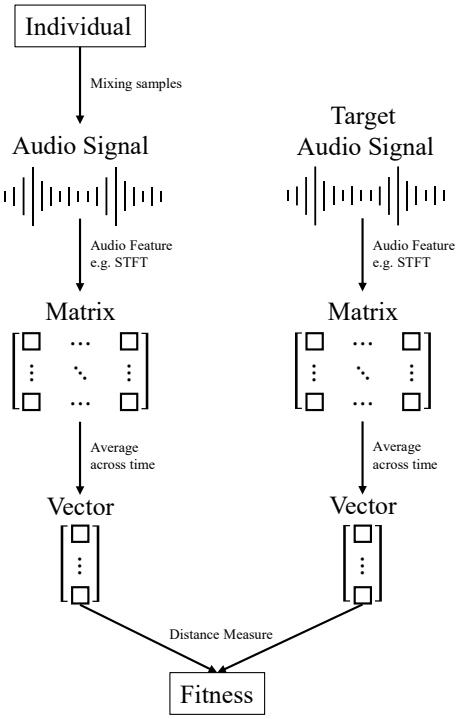


Figure 2.4: Calculation of the fitness value.

Each signal is transformed into a feature representation $\Phi(s) \in \mathbb{R}^{d \times T}$, where d is the dimension of the audio feature and T is the number of time steps. Temporal aggregation is performed by averaging across time:

$$\bar{\Phi}(s) = \frac{1}{T} \sum_{t=1}^{T} \Phi(s)_t. \tag{2.9}$$

This results in a d-dimensional feature vector $\bar{\Phi}(s) \in \mathbb{R}^d$. In this work 10 different audio features implemented in the librosa library (McFee et al., 2015) were considered:

1. **Short-Time Fourier Transform (STFT)**
   The STFT represents a signal by computing discrete Fourier transforms (DFT) over short overlapping windows.

2. **Mel-spectrogram**
   By aggregating the energy of uniformly spaced STFT frequency bins into mel-scaled bands, the representation reflects the non-uniform frequency resolution of the human auditory system (Zwicker & Fastl, 2013). As a result, frequency bands are narrow at low frequencies and become progressively wider as frequency increases.

3. **Mel Frequency Cepstral Coefficients (MFCC)**
   The mel spectrum can be represented more compactly using MFCC (Davis & Mermelstein, 1980). This is achieved by applying a discrete cosine transform to the logarithmic mel spectrum, which decorrelates the spectral components and enables a lower-dimensional representation with fewer features.

4. **Tonal Centroid Features (Tonnetz)**
   The Tonnetz is a well-known representation of pitch relationships, originally attributed to Euler and later widely adopted by 19th-century music theorists such as Riemann and Oettingen. Based on this concept, tonal centroid features are computed following the method of Harte et al. (2006), which projects chroma features onto a six-dimensional space, representing the perfect fifth, minor third and major third each as two-dimensional coordinates.

5. **Constant-Q Transform (CQT)**
   CQT is a time-frequency representation in which frequency bins are geometrically spaced, with all bins sharing a constant Q-factor, defined as the ratio of center frequency to bandwidth (Schörkhuber & Klapuri, 2010).

6. **Pseudo Constant-Q Transform (pseudo CQT)**
   The pseudo CQT employs a single fast Fourier transform (fft) size, chosen as the smallest power of two that is greater than or equal to the maximum of the longest CQT filter length and twice the number of samples between successive CQT columns.

7. **Hybrid Constant-Q Transform (hybrid CQT)** The hybrid CQT uses the pseudo CQT for higher frequencies where the number of samples between successive CQT columns is longer than half the filter length and the full CQT for lower frequencies.

8. **Variable-Q Transform (VQT)**
   VQT is similar to CQT with logarithmic frequency bin spacing and smoothly varying Q-factors (Schörkhuber et al., 2014).

9. **Infinite Impulse Response Transform (IIRT)**
   The input signal is first resampled according to the specified sample rates and then processed through band-pass infinite impulse response filters. The filter outputs are divided into frames and for each band, the short-time mean-square power (STMSP) is computed by summing a fixed number of consecutive filtered samples, producing a phase-linear, band-wise time-frequency representation.

10. **Combination of spectral features (Spectral)**
    This feature combines the spectral centroid, spectral bandwidth, spectral contrast, spectral flatness and roll-off frequency.

After applying an audio feature to the sound signal of the target and approximation, we are left with two d-dimensional feature vectors. To calculate the similarity of the feature vectors a distance $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ has to be applied. For each feature representation Φ and each distance measure D, a fitness function is defined as

$$f_{\Phi,D}(x) = D\big(\Phi(s_x), \Phi(s_{target})\big), \tag{2.10}$$

The following distance measures between two vectors $x, y \in \mathbb{R}^d_{>0}$ were considered.

1. L^p **Distance**
   The L^p distance (Minkowski distance) is defined as

$$D_p(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{\frac{1}{p}}, \qquad p \geq 1 \tag{2.11}$$

   In the limiting case of $p \to \infty$, we have

$$D_\infty(x,y) = \lim_{p\to\infty} \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{\frac{1}{p}} = \max_i |x_i - y_i|. \tag{2.12}$$

In this thesis we consider the three special cases:

$$p = 1: \quad \text{Manhattan distance} \tag{2.13}$$

$$p = 2: \quad \text{Euclidean distance} \tag{2.14}$$

$$p \to \infty: \quad \text{Chebyshev distance} \tag{2.15}$$

2. **COSH distance**
The COSH distance was first introduced by Gray and Markel (1976) as a distance measure for speech processing. As reformulated by Wei and Gibson (2001), the COSH distance between two vectors is given by

$$D_{\text{COSH}} = \frac{1}{2d} \sum_{i=1}^{d} \left(\frac{x_i}{y_i} - \log \frac{x_i}{y_i} + \frac{y_i}{x_i} - \log \frac{y_i}{x_i} - 2 \right) \tag{2.16}$$

It holds that $D_{\text{COSH}} \geq 0$. To proof this, we us the identity

$$\begin{aligned} &\log(a) \leq a - 1 \\ \Leftrightarrow &a - \log(a) \geq 1, \end{aligned} \tag{2.17}$$

which holds for all $a > 0$ (with equality iff $a = 1$) and can be derived from the Taylor series. First Equation 2.17 is used with $a = \frac{x_i}{y_i}$ and then with $a = \frac{y_i}{x_i}$:

$$\begin{aligned} D_{\text{COSH}} &= \frac{1}{2d} \sum_{i=1}^{d} \left(\frac{x_i}{y_i} - \log \frac{x_i}{y_i} + \frac{y_i}{x_i} - \log \frac{y_i}{x_i} - 2 \right) \\ &\geq \frac{1}{2d} \sum_{i=1}^{d} \left(1 + \frac{y_i}{x_i} - \log \frac{y_i}{x_i} - 2 \right) \\ &\geq \frac{1}{2d} \sum_{i=1}^{d} (1 + 1 - 2) = 0. \end{aligned} \tag{2.18}$$

3. **Cosine distance**
Cosine distance measures the dissimilarity between two vectors based on the cosine of the angle θ between them.

$$D_{\text{Cosine}}(x, y) = 1 - \cos(\theta) = 1 - \frac{x \cdot y}{||x|| \cdot ||y||} = 1 - \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \sqrt{\sum_{i=1}^{d} y_i^2}} \tag{2.19}$$

4. **Wasserstein distance**
The Wasserstein distance (Earth Mover's Distance) measures the minimal cost of transforming one probability distribution into another, considering the geometry of the underlying space. In the discrete case for 2 vectors $x, y \in \mathbb{R}^d$ the Wasserstein distance is defined as

$$W_p(x, y) = \inf_{\pi \in \Pi^d} \left(\frac{1}{d} \sum_{i=1}^{d} |x_i - y_{\pi(i)}|^p \right)^{1/p} \tag{2.20}$$

where Π^d denotes the set of all permutations of the set $\{1, \ldots, d\}$. In this thesis only the 1-Wasserstein distance will be used ($p = 1$).

5. **Itakura-Saito divergence**
   The Itakura–Saito divergence was first introduced by Itakura and Saito (1968) for speech processing. It is defined as

$$D_{\mathrm{IS}}(x,y) = \frac{1}{d}\sum_{i=1}^{d}\left(\frac{x_i}{y_i} - \log\frac{x_i}{y_i} - 1\right) \tag{2.21}$$

6. **Kullback–Leibler divergence**
   The Kullback–Leibler (KL) divergence measures how one probability distribution diverges from a reference distribution. In the discrete case for 2 vectors $x, y \in \mathbb{R}^d$ the KL divergence distance is defined as

$$D_{\mathrm{KL}}(x,y) = \sum_{i=1}^{d} x_i \log\frac{x_i}{y_i} \tag{2.22}$$

7. **Jensen-Shannon divergence**
   The Jensen–Shannon divergence is a measure of similarity between probability distributions that provides a symmetrised and smoothed version of the Kullback–Leibler divergence. It is defined as

$$D_{\mathrm{JS}}(x,y) = \frac{D_{\mathrm{KL}}(x,m) + D_{\mathrm{KL}}(y,m)}{2}, \qquad m = \frac{x+y}{2} \tag{2.23}$$

8. **β-divergence with $\beta = 1.5$**
   The β-divergence is a generalisation of the Itakura-Saito divergence (for $\beta = 0$), the Kullback-Leibler divergence (for $\beta = 1$) and the squared Euclidean distance (for $\beta = 2$). In this thesis we restrict ourselves to $\beta = 1.5$, then the β-divergence is given by:

$$D_{\beta}(x,y) = \sum_{i=1}^{d} \frac{1}{\beta(\beta-1)}\left(x_i^{\beta}(\beta-1)y_i^{\beta} - \beta x_i y_i^{\beta-1}\right), \quad \beta = 1.5 \tag{2.24}$$

Using the three special cases from the L^p distance we end up with 10 distance measures. Therefore, we can construct $10 \cdot 10 = 100$ fitness functions all of which will be evaluated in the single-objective EA.

2.2.4 Single-Objective Algorithm

The fitness function and individual representation described earlier are effective for approximating a single onset. However, musical pieces typically consist of many onsets, each potentially featuring different pitches and instruments. To account for this, we adapt the approach by assuming that multiple onsets within the same piece can be similar or even identical in terms of pitches and instrumentation. Instead of having an individual approximate only one onset, we allow it to represent a group of similar onsets distributed throughout the piece.

To achieve this, we introduce a parameter ϕ, which specifies the proportion of onsets used to evaluate an individual's fitness. The fitness is first computed for every onset. These values are then ranked and only the ϕ% with the best fitness scores are retained. The individual's overall fitness is defined as the mean of this selected subset. Following prior work (Dettmer, 2023; Vatolkin, 2020), $\phi = 5\%$ is used.

In our evolutionary algorithm, we generate only a single offspring per generation, a strategy commonly referred to as a steady-state ES (Bäck et al., 2000). To produce this offspring, a parent is selected uniformly at random and duplicated. Mutations, as described in Section 2.2.2, are then applied to the copy. Since parent selection is purely random, no selection pressure is exerted during reproduction. The only source of selection pressure occurs during the environmental selection phase: in each generation, the individual with the lowest fitness is eliminated.

The values of the parameters that are used in this thesis were established in previous work by Vatolkin (2020), Ginsel (2021) and Dettmer (2023) (Table 2.2). For the population size, the parameter value is specified in each experiment.

Table 2.2: Overview of the evolutionary algorithm parameters and their default values.

Parameter Name	Descriptor	Default Values
Population Size	μ	10, 300
Offspring per Generation	λ	1
Percentage of Approximated Onsets	ϕ	0.05
Maximum Number of Generations	n_{gen}	5000
Probabilities for n samples on Individual Creation	$p_{nsamples}$	[0.1, 0.3, 0.3, 0.2, 0.1]
Standard Deviation for Number of Mutations	α	5
Mean Number of Mutations	β	10
Lower Bound for Number of Mutations	l_{bound}	1
Upper Bound for Number of Mutations	u_{bound}	10
Pitch Shift Mutation Mean	μ_p	0
Pitch Shift Mutation Std. Deviation	σ_p	15
Step Size Adaption Multiplier	ζ	0.9954

Example

Figure 2.5 illustrates one generation of the evolutionary algorithm with $\mu = 3$ and $\lambda = 1$. The fitness value was calculated using STFT together with COSH distance and rounded to two decimal places. For each instrument, a fixed style was used so that this example can be easily replicated. The styles can be found in Table 2.3. The parent individual generated one offspring which underwent two mutations. The first was a pitch mutation, where the piano's pitch changed from A4 to A5. The second mutation involved the number of samples, resulting in the removal of the Flute playing E4. The fitness value of the offspring (shown in green) is 5.84, indicating that it is closer to the target than the parent. During environmental selection, the individual {(Flute, D6), (Trumpet, F5)}, which had the worst fitness of 10.23, was replaced by this new offspring.

Table 2.3: Instrument styles for exemplary fitness calculations.

Instrument	Style
Violin	Violins1_Essential
Piano	The_Grandeur
Trumpet	Trumpet_Essential
Flute	Flute_Essential

2.2.5 Multi-Objective Algorithm

The MOEA optimizes two or more objectives simultaneously. For m fitness functions $f_1, \ldots, f_m$, the problem can be formulated as

$$\text{Minimize} \quad F(x) = [f_1(x), f_2(x), \ldots, f_m(x)] \tag{2.25}$$

$$\text{subject to} \quad x \in \mathcal{X}, \tag{2.26}$$

where $\mathcal{X}$ denotes the set of all feasible individuals defined in 2.5 (Zhou et al., 2011).

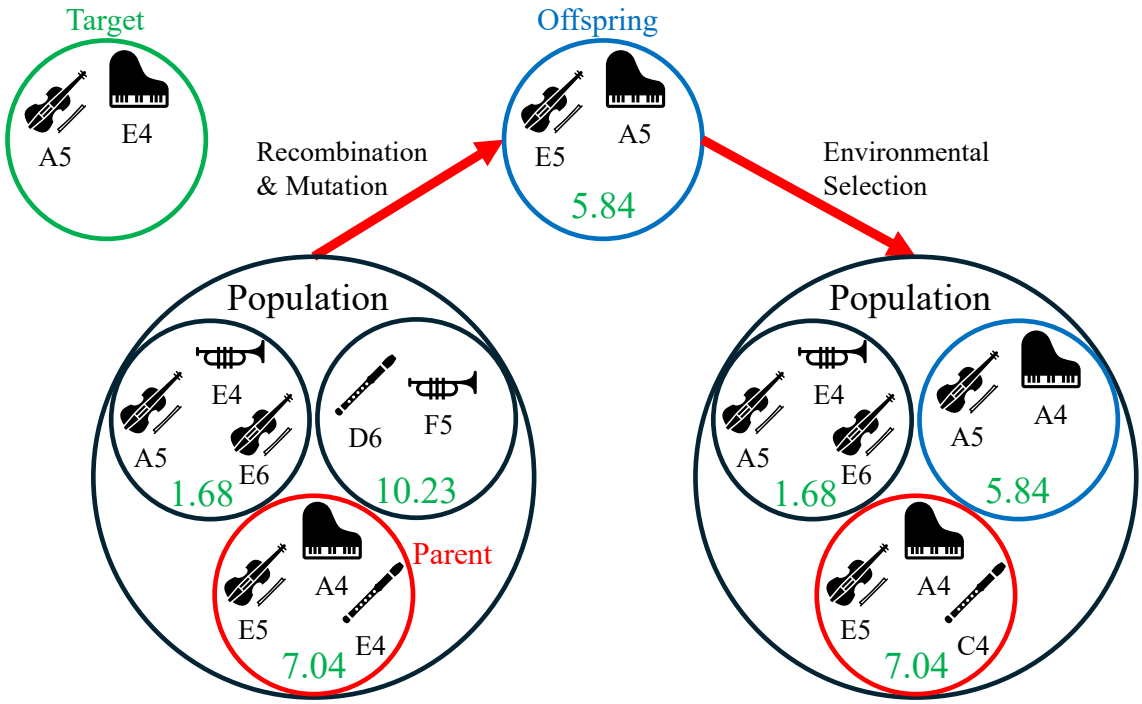


Figure 2.5: Evolutionary loop with 3 individuals. STFT + COSH fitness value in green.

In contrast to the single-objective EA, comparing individuals becomes non-trivial since each individual is evaluated with respect to multiple objective values. To define a comparison relation, we introduce the concept of Pareto dominance.

An individual $x^* \in \mathcal{X}$ dominates another individual $x \in \mathcal{X}$ (denoted by $x^* \prec x$) if

$$\forall i \in \{1, \ldots, m\} : f_i(x^*) \leq f_i(x) \tag{2.27}$$

$$\exists j \in \{1, \ldots, m\} : f_j(x^*) < f_j(x). \tag{2.28}$$

That is, x^* is not worse than x in all objectives and strictly better in at least one objective.

The Pareto set (or Pareto front) P consists of all non-dominated solutions and is defined as

$$P = \{x^* \in \mathcal{X} \mid \nexists x \in \mathcal{X} : x \prec x^*\}. \tag{2.29}$$

Since comparison of individuals is only required during the environmental selection phase, the rest of the algorithm can remain unchanged from the single-objective EA. The primary difference is that the fitness of each individual is now represented as a vector of multiple objective values. For the environmental selection, we implemented two different approaches.

NSGA-II

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002) uses non-dominated sorting to rank individuals based on Pareto dominance. Rank 1 contains all non-dominated solutions, rank 2 contains solutions dominated only by those in rank 1 and so on. During selection, individuals with lower ranks are preferred. Within each rank, a crowding distance metric is used

to maintain diversity by favouring individuals located in less crowded regions of the objective space.

The crowding distance provides an estimate of the local density of solutions surrounding a given individual. For a set of solutions I belonging to the same non-dominated front, the crowding distance of an individual is computed based on the distances to its nearest neighbours along each objective dimension. Formally, for each objective m, the solutions in I are sorted in ascending order according to their objective values. Boundary solutions are assigned an infinite crowding distance to ensure their preservation. For all other solutions, the crowding distance is calculated as

$$d_i = \sum_{m=1}^{M} \frac{f_m(i+1) - f_m(i-1)}{f_m^{\max} - f_m^{\min}}, \tag{2.30}$$

where $f_m(i)$ denotes the m-th objective value of the i-th solution in the sorted list and $f_m^{\max}$ and $f_m^{\min}$ are the maximum and minimum values of the m-th objective within the front. This normalization ensures that all objectives contribute equally to the distance measure.

AS shown in Figure 2.6, the resulting crowding distance corresponds to the perimeter of the hyper-rectangle formed by the nearest neighbours of a solution. It serves as a measure of how isolated the solution is in the objective space. During selection, individuals with larger crowding distance are preferred, thereby promoting diversity. The pseudo-code for the environmental selection of NSGA-II is presented in Algorithm 2.

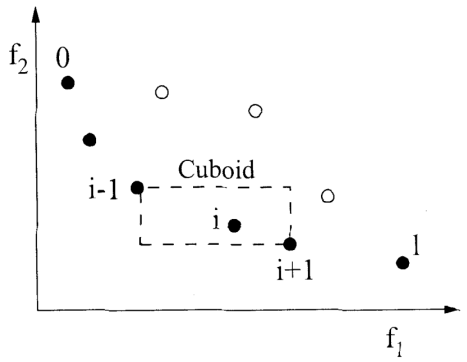


Figure 2.6: Crowding-distance calculation. Points marked in filled circles are solutions of the same non-dominated front. (Source: Deb et al. (2002))

Algorithm 2: NSGA-II environmental selection, based on Deb et al. (2002)

Input: Population P with $|P| = \mu + \lambda$
Output: Reduced population P with $|P| = \mu$

```
Perform non-dominated sorting to obtain fronts F_1, ..., F_v;
P' ← ∅;
i ← 1;
while |P'| + |F_i| ≤ μ do
    Compute crowding distances in F_i;
    P' ← P' ∪ F_i;
    i ← i + 1;
Sort F_i in descending order of crowding distance;
Add the first μ − |P'| individuals from F_i to P';
return P';
```

SMS-EMOA

The S-Metric Selection Evolutionary Multi-Objective Algorithm (SMS-EMOA) (Beume et al., 2007) is a MOEA that uses the hypervolume contribution as a selection criterion. SMS-EMOA directly optimizes a quality indicator that simultaneously captures convergence towards the Pareto front and diversity among solutions.

The underlying quality measure is the so-called S-metric (also referred to as hypervolume). Given a set of points $M \subset \mathbb{R}^d$ and a reference point $r \in \mathbb{R}^d$, the S-metric measures the volume of the portion of the objective space dominated by M and bounded by r. Formally, it is defined as

$$S(M, r) = \Lambda \left(\bigcup_{y \in M} h(y) \right), \tag{2.31}$$

where Λ denotes the Lebesgue measure and $h(y) = [y_1, r_1] \times \cdots \times [y_d, r_d]$ is the hyper-rectangle dominated by a point y.

Based on this measure, the hypervolume contribution of an individual y with respect to M is defined as the decrease in hypervolume when removing y from the set:

$$\Delta S(y, M) = S(M, r) - S(M \setminus \{y\}, r). \tag{2.32}$$

An example of the hypervolume contribution is illustrated in Figure 2.7.

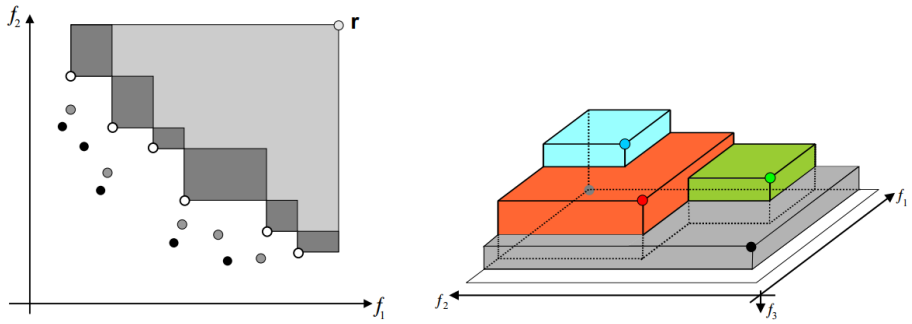


Figure 2.7: Hypervolume contributions from the third front of a population in a two-dimensional target space (left) and an example of a three-dimensional target space (right). (Source: Beume et al. (2007))

SMS-EMOA follows a steady-state evolutionary scheme, which is consistent with the EA used for instrument and pitch detection. In the environmental selection phase (Algorithm 3) a non-dominated sorting procedure is applied. The individual from the worst non-dominated front (similar to the approach used in NSGA-II) that has the smallest hypervolume contribution is removed. By removing the least contributing individual, the algorithm greedily maximizes the overall hypervolume.

2.3 Evaluation Metrics

Both evaluation metrics and fitness functions aim to measure the quality of an approximation. However, they differ fundamentally in the information they use. Fitness functions measure the distance between two audio signals without access to the target annotation, whereas evaluation metrics measure the distance between two annotations.

Algorithm 3: SMS-EMOA environmental selection, based on Beume et al. (2007)

Input: Population P with $|P| = \mu + 1$
Output: Reduced population P with $|P| = \mu$
Perform non-dominated sorting to obtain fronts $F_1, \ldots, F_v$;
while $|P| > \mu$ **do**
    Select worst front F_v;
    Compute $\Delta S(y, F_v)$ for all $y \in F_v$;
    Remove $a^* = \arg\min_{y \in F_v} \Delta S(y, F_v)$;
return P;

In the single-objective setting, the individual with the lowest fitness value for each onset of the target is stored in an archive. To evaluate the performance of the single-objective EA, each target onset is compared to its corresponding best individual and the results are averaged over all onsets. This requires defining a metric between a single individual and a target onset.

We begin with commonly used metrics that do not incorporate perceptual similarity between sounds and gradually introduce more advanced metrics that better capture the perceptual quality of the approximation. Subsequently, evaluation metrics for the MOEA are presented. Finally, the Shapley value is introduced as a method for selecting subsets of fitness functions.

2.3.1 Standard Evaluation Metrics

Recall that an individual $x : \mathcal{K} \to \mathbb{N}_0$ is a multiset of samples. We define the support of x as

$$\operatorname{supp}(x) = \{k \in \mathcal{K} \mid x(k) > 0\}. \tag{2.33}$$

Using this notation, we define the following sets associated with an individual x.

Set of instruments in x.

$$\mathcal{I}_x = \{I \mid \exists\, S \in \mathcal{S}_I,\ \exists\, P \in \mathcal{P}_{I,S} \text{ such that } (I, S, P) \in \operatorname{supp}(x)\}. \tag{2.34}$$

Set of pitches in x.

$$\mathcal{P}_x = \{P \mid \exists\, I \in \mathcal{I},\ \exists\, S \in \mathcal{S}_I \text{ such that } (I, S, P) \in \operatorname{supp}(x)\}. \tag{2.35}$$

Set of instrument-pitch combinations in x.

$$\mathcal{C}_x = \{(I, P) \mid \exists\, S \in \mathcal{S}_I \text{ such that } (I, S, P) \in \operatorname{supp}(x)\}. \tag{2.36}$$

Note that the style S is intentionally ignored. We consider an instrument as correctly identified if the correct instrument is present, regardless of the specific style. Identifying the exact style would only be possible if the target sound had been generated using the same sample library as the algorithm. However, for many target recordings, the precise styles are unknown.

Let $x \in \mathcal{X}$ denote an approximation produced by the evolutionary algorithm, and let y denote the corresponding target audio signal. Since the target is given as a sound signal rather than an individual in $\mathcal{X}$, it is represented through annotated labels describing the contained instruments and pitches.

For evaluation, we consider set-based representations derived from both the approximation and the target. Let $\mathcal{A}$ denote one of the sets derived from x (e.g., $\mathcal{I}_x$, $\mathcal{P}_x$, or $\mathcal{C}_x$), depending on whether the evaluation is performed on instrument level, pitch level, or instrument-pitch level. Using the annotations of y, we define a corresponding target set $\mathcal{B}$ in the same label space. In particular, $\mathcal{B}$ contains the annotated instruments, pitches, or instrument-pitch combinations present in the target signal.

Note that $\mathcal{B}$ is not required to be restricted to elements representable by $\mathcal{X}$, as the target may contain instruments or pitches that are not available in the sample library. Nevertheless, both $\mathcal{A}$ and $\mathcal{B}$ are defined in a common label space, enabling a direct comparison.

True Positives (TP), False Positives (FP), False Negatives (FN).

$$\mathrm{TP} = |\mathcal{A} \cap \mathcal{B}|, \tag{2.37}$$

$$\mathrm{FP} = |\mathcal{A} \setminus \mathcal{B}|, \tag{2.38}$$

$$\mathrm{FN} = |\mathcal{B} \setminus \mathcal{A}|. \tag{2.39}$$

True positives correspond to correctly identified elements. False positives represent elements predicted by the approximation that are not present in the target. False negatives denote elements that are present in the target but missing in the approximation. True negatives are not considered, since the universal set is large and not meaningful for evaluation in this context.

Precision

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}|}. \tag{2.40}$$

Precision measures the proportion of predicted elements that are actually correct. A high precision indicates that few incorrect elements are included in the approximation.

Recall

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{B}|}. \tag{2.41}$$

Recall measures the proportion of relevant target elements that are successfully recovered by the approximation. A high recall indicates that few relevant elements are missed.

F_1 Score

$$F_1 = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} = \frac{2|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}| + |\mathcal{B}|}. \tag{2.42}$$

The F_1 score is the harmonic mean of precision and recall. It provides a balanced measure that is high only if both precision and recall are high.

Jaccard Distance

$$J = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} = 1 - \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|} \tag{2.43}$$

The Jaccard distance is equal to zero if both sets are identical and approaches one as the sets become increasingly different.

Example

To illustrate the differences between the evaluation metrics, consider the following example on instrument-pitch level (styles are ignored).

Let the target set be

$$\mathcal{B}_{IP} = \{(\text{Violin}, C5), (\text{Piano}, C4), (\text{Piano}, E4)\}. \tag{2.44}$$

Assume the approximation produces

$$\mathcal{A}_{IP} = \{(\text{Viola}, C5), (\text{Piano}, E3)\}. \tag{2.45}$$

Using the equations from above we can calculate the metrics on instrument level, pitch level and jointly on the instrument-pitch level.

Instrument Level

$$\mathcal{B}_I = \{\text{Violin}, \text{Piano}\}, \quad \mathcal{A}_I = \{\text{Viola}, \text{Piano}\}.$$

$$\text{TP}_I = 1, \quad \text{FP}_I = 1, \quad \text{FN}_I = 1.$$

$$\text{Precision}_I = \frac{1}{2}, \quad \text{Recall}_I = \frac{1}{2}, \quad F_{1,I} = \frac{1}{2}, \quad J_I = \frac{2}{3}.$$

Pitch Level

$$\mathcal{B}_P = \{C5, C4, E4\}, \quad \mathcal{A}_P = \{C5, E3\}.$$

$$\text{TP}_P = 1, \quad \text{FP}_P = 1, \quad \text{FN}_P = 2.$$

$$\text{Precision}_P = \frac{1}{2}, \quad \text{Recall}_P = \frac{1}{3}, \quad F_{1,P} = \frac{2}{5}, \quad J_P = \frac{3}{4}.$$

Instrument–Pitch Level

$$\text{TP}_{IP} = 0, \quad \text{FP}_{IP} = 2, \quad \text{FN}_{IP} = 3.$$

$$\text{Precision}_{IP} = 0, \quad \text{Recall}_{IP} = 0, \quad F_{1,IP} = 0, \quad J_{IP} = 1.$$

Although the approximation of a Violin with Viola and the approximation of an E4 with an E3 is perceptually very close, the metrics suggest a very poor performance. On the instrument-pitch level we can even see the worst errors possible, however, an approximation using drums and trumpets would perceptually be further away than the approximation in the example. It becomes evident, that these metrics are only suited for comparison of different settings of the algorithm but give us very little information about the actual quality of an approximation. In the next chapters we will introduce metrics that include information about the perceptual distance of approximation and target.

2.3.2 Chord Distances

In this chapter we will tackle the pitch level metrics. One common definition of a chord is a group of two or more notes sounding together. Thus, we can interpret the set $\mathcal{P}_x$ of pitches in x as a chord allowing also for 1-note chords, which would technically not be considered a chord. Note, that we exclude the empty set since our algorithm is forced to include at least one sample and the target also shouldn't be empty since we are not interested in approximating silence.

There has been a lot of research on the topic of chord distances ranging from the music theory perspective (Rocher et al., 2010) to a very mathematical point of view (Hurajová & Hutník, 2024). To calculate distances between chords of different sizes there are several different approaches. There are metrics such as the Hausdorff distance and the Fujita distance that generalize from an element-wise distance to a distance on sets. In this case we just need to define a distance between single notes to get a distance between chords. For other distances more sophisticated approaches are necessary. One such method is to represent a chord as a set of intervals, as used for the intervalic distance (Rocher et al., 2010). This approach encodes a chord as a 6-dimensional binary vector, where each entry indicates the presence or absence of a specific interval (ranging from a minor second to a diminished fifth). Note that the intervals only range up to a diminished fifth which is a distance of 6 semitones. This is because for any pair of pitch classes, one can choose an ordering such that the interval between them is at most six semitones. The interval-vector representation of a C major chord can be seen in Table 2.4:

Table 2.4: Interval-vector representation of a C-Major chord.

1 (Minor 2^{nd})	2 (Major 2^{nd})	3 (Minor 3^{rd})	4 (Major 3^{rd})	5 (Perfect 4^{th})	6 (Dim 5^{th})
0	0	1 (E-G)	1 (C-E)	1 (G-C)	0

Another representation encodes a chord as a 12-dimensional binary vector, where each entry corresponds to a pitch class (e.g., see Table 2.5) (Hurajová & Hutník, 2024). This representation is invariant to shifting the pitches by octaves. Since for our application the range is limited to 88 different pitches we can also consider an 88-dimensional binary vector representation that preserves octave information.

Table 2.5: Vector representation of a C-Major chord.

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
1	0	0	0	1	0	0	1	0	0	0	0

In the following various distances between chords are introduced. A comparison between the metrics on selected chords can be found in Table 2.6 and Table 2.7

ℓ_p-norm on the 12-dim representation (ℓ_p^{12})

Let V_C^{12} the 12-dimensional binary representation of the a chord C. Then the ℓ_p-norm between two chords C_1 and C_2 is defined as

$$\ell_p^{12}(C_1, C_2) = \|V_{C_1}^{12} - V_{C_1}^{12}\|_p = \left(\sum_{i=1}^{12} |V_{C_1}^{12}(i) - V_{C_2}^{12}(i)|^p \right)^{\frac{1}{p}} \tag{2.46}$$

Similarly we can also define the ℓ_p-norm on the 88-dimensional binary representation V_C^{88} as

$$\ell_p^{88}(C_1, C_2) = \|V_{C_1}^{88} - V_{C_1}^{88}\|_p \tag{2.47}$$

Intervalic Distance (I)

Let V_C^6 be the interval-vector representation of a chord C. Then the intervalic distance (Rocher et al., 2010) between two chords C_1 and C_2 is defined as the ℓ_1-norm:

$$d_I(C_1, C_2) = \sum_{i=1}^{6} |V_{C_1}^6(i) - V_{C_2}^6(i)| \tag{2.48}$$

Geometric Distance (G)

This distance aims to mimic the human perception by building on Riemanns Tonnetz which is a network of harmonically related pitches. Chew (2000) introduced a spiral model to represent tonal relationships. For this a root pitch k is placed on a helix:

$$P(k) = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} r \cdot \sin\left(\frac{k\pi}{2}\right) \\ r \cdot \cos\left(\frac{k\pi}{2}\right) \\ k \cdot h \end{bmatrix} \tag{2.49}$$

Chords are then represented by converting each pitch in the chord to its index in the circle of fifth. For a C-major chord this means C is represented as $P(k)$, G is represented as $P(k+1)$ since it is 1 perfect fifth away from C and E is represented as $P(k+4)$ as E is 4 perfect fifth away from C.

Then the centroid of the chord U is computed as the average of its helix positions $p_1, ..., p_n$:

$$C(U) = \frac{1}{n} \sum_{i=1}^{n} p_i \tag{2.50}$$

The distance between two chords U and V can then be calculated as the euclidean distance between their centroids as proposed by Rocher et al. (2010):

$$d_G(U, V) = \|C(U) - C(V)\|_2 \tag{2.51}$$

In (Chew, 2000) it was shown that the ratio of the hyperparameters r and h should be in the range of:

$$\sqrt{\frac{2}{15}} < \frac{h}{r} < \sqrt{\frac{2}{7}} \tag{2.52}$$

so approximately $0.365 < h/r < 0.535$. For the experiments in this thesis the values $h = 0.5$ and $r = 1$ were chosen.

Distance on the n-Torus ($d_{\mathcal{A}_n}$)

In the 12-dimensional representation we basically think of chords with n notes as points in $\mathbb{R}^n$ under the equivalence modulo an octave and under the equivalence of permutations of the pitch classes (unordered chords). Mathematically speaking the equivalence modulo an octave is the equivalence modulo a group of translations by integers $\mathbb{Z}_{12}^n$. The space $\mathbb{R}^n/\mathbb{Z}_{12}^n$ is known as the

n-torus $\mathbb{T}^n$. Hence, a chord consisting of n notes can be seen as $\mathcal{A}_n = \mathbb{T}^n/\mathcal{S}_n$ where S_n is the group of permutations of n. In (Hurajová & Hutník, 2024) it was shown that $\mathcal{A}_n$ is a metric space with the distance:

$$d_{\mathcal{A}_n}([x],[y]) = \inf_{\sigma\in S_n} d_{\mathbb{T}^n}(\bar{x}, \overline{\sigma(y)}) \tag{2.53}$$

with

$$d_{\mathbb{T}^n}(\bar{x},\bar{y}) = \inf_{k\in\mathbb{Z}_{12}^n} \|x-(y+k)\|_{\mathbb{R}^n} \tag{2.54}$$

where $\|\cdot\|_{\mathbb{R}^n}$ is a norm on $\mathbb{R}^n$. From now on the ℓ_1-norm will be used in line with Hurajová and Hutník (2024). Note that this distance is only defined between chords of equal sizes and therefore can not be used for comparing chords of different sizes. Nevertheless it is a core building block for the following distances. In the 1-dimensional case where the chords are just single notes the distance simplifies to:

$$\begin{aligned} d_{\mathcal{A}_1}([x],[y]) &= \inf_{\sigma\in S_1}\inf_{k\in\mathbb{Z}_{12}^1} \|x-(\sigma(y)+k)\|_{\mathbb{R}} && (2.55)\\ &= \inf_{k\in\mathbb{Z}_{12}} |x-(y+k)| && (2.56)\\ &= \inf_{n\in\mathbb{Z}} |x-(y+12n)| && (2.57)\\ &= \min(|x-y| \mod 12, 12-(|x-y| \mod 12)) && (2.58) \end{aligned}$$

which is the minimum interval between the two pitch-classes in semitones. For example, for pitches C and G:

$$d_{\mathcal{A}_1}(C,G) = \min(7,5) = 5 \tag{2.59}$$

Hausdorff distance (H)

Given two chords U and V (as sets of pitches), the Hausdorff distance is defined as:

$$d_H(U,V) := \max\left\{\sup_{u\in U}\delta(u,V), \sup_{v\in V}\delta(v,U)\right\}, \tag{2.60}$$

where $\delta(u,V) = \inf_{v\in V} d_X(u,v)$ and $\delta(v,U) = \inf_{u\in U} d_X(u,v)$. In this thesis $d_X(u,v) = d_{\mathcal{A}_1}(u,v)$ was used as suggested in (Hurajová & Hutník, 2024).

Average Hausdorff distance (AH)

Given two chords U and V (as sets of pitches), the average Hausdorff distance is defined as:

$$d_{AH}(U,V) = \max(GD_p(U,V), GD_p(V,U)) \tag{2.61}$$

where

$$GD_p(U,V) := \left(\frac{1}{\#(U)}\sum_{i=1}^{\#(U)}\delta(u_i,V)^p\right)^{\frac{1}{p}} = \left(\frac{1}{\#(U)}\sum_{i=1}^{\#(U)}\min_{j=1,\dots,\#(V)}\left\{d_X(u_i,v_j)^p\right\}\right)^{\frac{1}{p}} \tag{2.62}$$

In the experiments in this thesis we set $p=2$ and $d_X(u,v) = d_{\mathcal{A}_1}(u,v)$ consistent with Hurajová and Hutník (2024).

Projection distance (P)

Let $U \in \mathcal{A}_m$ and $V \in \mathcal{A}_n$ be chords with $m \leq n$ (U is the chord that contains more notes) then the projection distance is given by

$$d_p(U,V) := \max_{S \in \mathrm{Proj}_m(V)} d_{\mathcal{A}_m}(U,S) \tag{2.63}$$

where $\mathrm{Proj}_k(V) := \{S \subset V : \#(S) = k\}$ (Hurajová & Hutník, 2024).

Fujita distance (F)

This distance was first introduced by Fujita (2013) as a generalization from a distance between elements to a distance between sets. As a distance between two chords U and V (as sets of notes) it is defined as

$$d_F(U,V) = \frac{1}{\#(U \cup V)\#(U)} \sum_{u \in U} \sum_{v \in V \setminus U} d_X(u,v) + \frac{1}{\#(U \cup V)\#(V)} \sum_{u \in U \setminus V} \sum_{v \in V} d_X(u,v) \tag{2.64}$$

using $d_X(u,v) = d_{\mathcal{A}_1}(u,v)$ (Hurajová & Hutník, 2024).

STFT+COSH fitness

As a comparison between the distances and a fitness function the chords in Table 2.7 were constructed using the piano in the style "The_Grandeur" from the sample library. Then the STFT was computed before applying the COSH-distance as introduced in 2.2.3.

Table 2.6: Chord distances to C-Major using various metrics

Chord	Notes	ℓ_2^{12}	ℓ_2^{88}	I	G	H	AH	P	F	STFT+COSH
C_maj	c4, e4, g4	0.00	0.00	0	0.00	0	0.00	0	0.00	0.00
C_maj_inv	e4, g4, c5	0.00	1.41	0	0.00	0	0.00	0	0.00	0.78
C_maj_high	c5, e5, g5	0.00	2.45	0	0.00	0	0.00	0	0.00	2.47
C_min	c4, dis4, g4	1.41	1.41	0	0.96	1	0.58	1	1.33	1.95
C7	c4, e4, g4, b4	1.00	1.00	3	0.48	1	0.50	5	0.83	0.16
D_maj	d4, fis4, a4	2.45	2.45	0	1.80	2	1.73	6	3.11	2.59
E_maj	e4, gis4, b4	2.00	2.00	0	2.00	1	0.82	2	2.40	1.29
F_maj	f4, a4, c5	2.00	2.45	0	1.83	2	1.29	3	2.47	3.02
G_maj	g4, b4, d5	2.00	2.00	0	1.17	2	1.29	3	2.47	1.50
G7	g4, b4, d5, f5	2.24	2.24	3	1.80	2	1.22	5	2.50	2.23
A_min	a4, c5, e5	1.41	2.45	0	0.75	2	1.15	2	1.67	5.16
B_dim	b4, d5, f5	2.45	2.45	4	2.41	2	1.41	5	3.00	3.00
C	c4	1.41	1.41	3	0.96	5	3.70	5	3.00	5.40
D	d4	2.00	2.00	3	1.71	5	3.32	5	3.00	3.77
C_all	c4–b4 (all)	3.00	3.00	63	2.06	2	1.22	9	2.33	3.27

Ginsel Distance

Ginsel (2021) introduced a pitch distance that is based on a model of the human hearing by Ebeling (2007). The distance between pitches p_1 and p_2 is defined as:

$$d_{\mathcal{G}}(p_1, p_2) = \begin{cases} 0 & |p_1 - p_2| = 0 \\ 1 & |p_1 - p_2| = 12 \\ 2 & |p_1 - p_2| = 7 \\ 3 & |p_1 - p_2| \in \{3; 4; 5; 9\} \\ 4 & |p_1 - p_2| \in \{1; 2; 6; 8; 10; 11\} \\ 5 & \text{otherwise} \end{cases} \tag{2.65}$$

Since this is a distance between single pitches it can be used in the Hausdorff distance (H_G), average Hausdorff distance (AH_G) and Fujita distance (F_G). The results in comparison with the corresponding distances using $d_{\mathcal{A}_1}$ and the fitness can be seen in Table 2.7.

Table 2.7: Chord distances to C-Major including Ginsel distance

Chord	H	H_G	AH	AH_G	F	F_G	STFT+COSH
C_maj	0	0	0.00	0.00	0.00	0.00	0.00
C_maj_inv	0	1	0.00	0.58	0.00	1.17	0.78
C_maj_high	0	1	0.00	1.00	0.00	3.11	2.47
C_min	1	3	0.58	1.73	1.33	1.67	1.95
C7	1	2	0.50	1.00	0.83	0.75	0.16
D_maj	2	4	1.73	3.37	3.11	3.67	2.59
E_maj	1	3	0.82	2.45	2.40	2.73	1.29
F_maj	2	3	1.29	2.52	2.47	3.22	3.02
G_maj	2	2	1.29	1.63	2.47	2.67	1.50
G7	2	4	1.22	2.45	2.50	3.14	2.23
A_min	2	3	1.15	1.91	1.67	3.00	5.16
B_dim	2	4	1.41	2.83	3.00	3.78	3.00
C	5	3	3.70	2.08	3.00	1.67	5.40
D	5	4	3.32	3.70	3.00	3.67	3.77
C_all	2	4	1.22	2.63	2.33	2.67	3.27

2.3.3 Instrument Distances

In order to evaluate the proposed EA, the definition of a perceptual distance measure between instruments is a core component. The sample library used in this work contains 51 instruments, including not only common Western instruments but also less typical ones such as Bouzouki Sakis, Oud, Erhu and Shakuhachi. Classical studies on instrument distances consider only a small subset of instruments, typically ranging between 10 and 20 common Western instruments (Grey, 1977). Such approaches are not directly applicable in our setting due to the significantly larger and more diverse instrument set.

To obtain perceptually meaningful distances, we rely on the timbre features proposed by McAdams et al. (1995). These features capture perceptually relevant properties of musical sounds and have been shown to reflect human similarity judgments of timbre. Precomputed McAdams features for each instrument-style-pitch combination were provided by Ginsel (2021). For each pitch, pairwise absolute distances between all instrument-style combinations were computed in this feature space. Our goal is to derive a single distance measure between instruments from these distances.

To this end, we aggregate the distances as follows. For each pair of instruments, we consider all combinations of their respective styles and all pitches that are available for both instruments. The instrument distance is then defined as the average of the corresponding McAdams distances across all overlapping pitches and styles. After aggregation, all distances are normalized using min-max scaling to the interval $[0, 1]$, where 0 corresponds to identical timbrel characteristics and 1 to maximal dissimilarity.

Averaging across styles is necessary in order to obtain a single distance value for each instrument pair that can subsequently be used in the evaluation. However, this aggregation is only meaningful if different styles of the same instrument are sufficiently close under McAdams distance. To assess this assumption, we performed an exploratory analysis using multidimensional scaling (MDS) on the style-wise distances. For instruments with clearly distinct timbres, such as drums, flute, and violin, the resulting MDS visualization shows well-separated clusters corresponding to the individual instruments, while different styles of the same instrument remain grouped together. In contrast, for more closely related instruments such as violin, viola, and cello, the styles do not form clearly separable clusters and instead overlap substantially. The corresponding MDS visualizations are provided in the appendix (Section B).

A challenge arises for instrument pairs that do not share any overlapping pitch range. For example, the Bass (range: C1 to C#4) and the South American flute Pinkillo (range: C5 to C7) have no pitches in common. In such cases, no McAdams-based distance can be computed. This affects 19 instrument pairs in the dataset, primarily combinations of bass and flute-like instruments. For these pairs, we assign the maximal distance of 1, reflecting maximal dissimilarity.

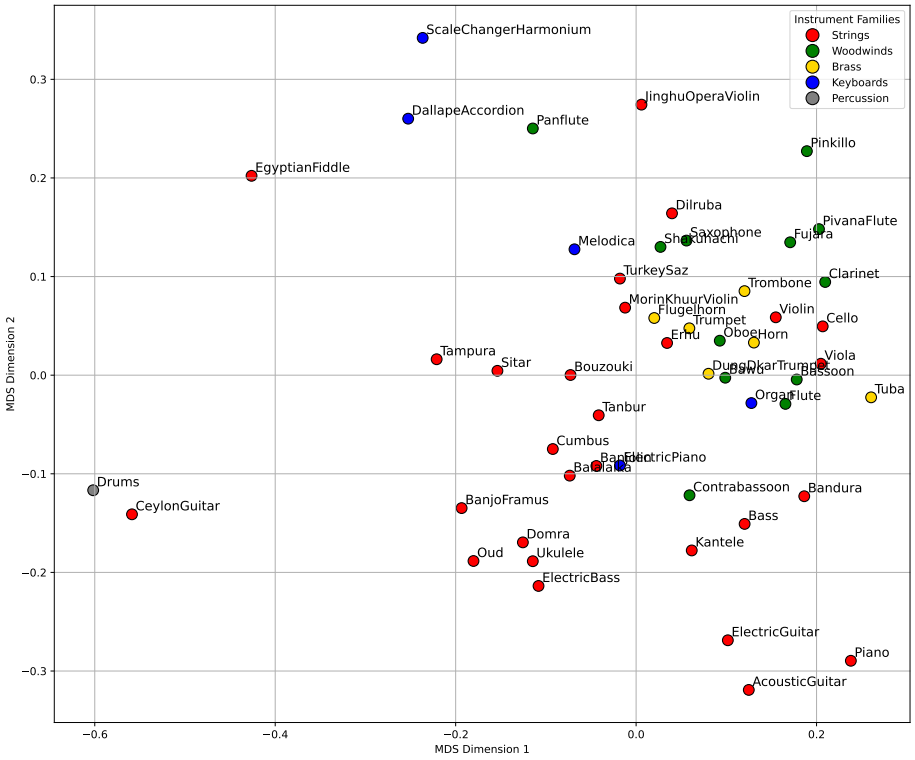


Figure 2.8: MDS projection of instrument distances into two-dimensional space.

To visualize the resulting distance structure, we apply MDS to project the high-dimensional distance matrix into a two-dimensional space (Figure 2.8). The resulting embedding reveals several expected patterns: closely related string instruments such as Violin, Viola and Cello cluster together, as do electric and acoustic guitars, as well as brass instruments including Trombone, Flügelhorn, Trumpet and Horn. Percussive instruments (e.g., drums) are positioned far from most tonal

instruments, reflecting their distinct timbral properties. Interestingly, drums appear relatively close to the CeylonGuitar, which is not immediately intuitive.

2.3.4 MOEA Metrics

The key difference between the single-objective and multi-objective EA lies in the notion of optimality. In the multi-objective setting, there is no single best individual. Instead, a set of non-dominated solutions forms the Pareto front, as defined in 2.29.

This makes a direct comparison between the single-objective EA and the MOEA challenging. To address this, we first introduce the success rate as an evaluation metric for the single-onset approximation task. For the approximation of multiple onsets simultaneously, we then introduce the Generational Distance (GD) and Inverted Generational Distance (IGD) as evaluation metrics.

Success Rate

In the single onset approximation, we can treat the task as a strict classification problem of a single polyphonic audio signal drawn from the sample library. A run is considered successful if there exists at least one individual x in the final population $P \subset \mathcal{X}$ that has the same instrument-pitch combinations as the target x^*. Formally,

$$\text{successful run} \Leftrightarrow \exists x \in P : J_{ip}(x, x^*) = 0 \tag{2.66}$$

where J_{ip} denotes the Jaccard-distance on the instrument-pitch level defined in Equation 2.43. Then, the success rate s is defined as

$$s = \frac{|\text{successful runs}|}{|\text{all runs}|}. \tag{2.67}$$

GD and IGD

GD and IGD are standard metrics in MOEAs used to evaluate how closely a population approximates the Pareto front (Menchaca-Mendez & Coello Coello, 2015; Sun et al., 2018). In the context of this work, we repurpose these metrics to assess the ability of an evolutionary algorithm to approximate a set of target musical onsets in a given piece.

Let P denote the final population produced by the algorithm and let P^* denote an *idealized population*. This idealized population contains one individual for each onset in the target musical piece, where each individual exactly represents the instruments and pitches present at that onset. Thus, P^* can be interpreted as the best possible reconstruction of the piece in the chosen representation.

GD measures the average distance from individuals in the generated population P to their closest counterparts in the idealized population P^*:

$$GD_p(P, P^*) := \left(\frac{1}{|P|} \sum_{x \in P} \min_{x^* \in P^*} \left\{ d\left(x, x^*\right)^p \right\} \right)^{\frac{1}{p}} \tag{2.68}$$

where $d(x, x^*)$ is a distance between the two individuals x and y. We can use any of the distances defined in Section 2.3.1, Section 2.3.2, or Section 2.3.3. A smaller GD indicates that the population is, on average, closer to the target onsets.

IGD is complementary to GD and measures the average distance from individuals in the idealized population P^* to the nearest individual in the generated population P:

$$IGD_p(P, P^*) := GD_p(P^*, P) = \left(\frac{1}{|P^*|} \sum_{x^* \in P^*} \min_{x \in P} \left\{ d\left(x^*, x\right)^p \right\} \right)^{\frac{1}{p}} \tag{2.69}$$

IGD captures how well the generated population covers all target onsets. A low IGD value indicates that every target onset is closely approximated by at least one individual in P. For both GD and IGD we use $p = 1$.

In this context, GD and IGD can be interpreted analogously to precision and recall. GD behaves similarly to a precision metric: it is small when every individual in the generated population closely matches some target onset. However, this does not guarantee that all target onsets are covered, as the population may focus on approximating only a subset of the piece.

In contrast, IGD behaves similarly to a recall metric: it is small when every target onset in P^* is well approximated by at least one individual in P. This ensures that the generated population captures the full structure of the musical piece. An important property of IGD in this setting is its dependence on the population size. As the number of individuals in P increases, the likelihood that each target onset in P^* is approximated by at least one individual also increases. In the limiting case of an exhaustive population containing all possible individuals, IGD would approach zero. Consequently, IGD is not directly comparable across populations of different sizes. To ensure a fair evaluation, it is therefore necessary to keep the population size constant when comparing different algorithmic configurations. This guarantees that differences in IGD reflect the quality of the approximation rather than variations in population size.

For the task considered in this work, IGD is the more informative metric, as the primary objective is to approximate the complete set of onsets rather than achieving very accurate reconstructions of only a subset. Nevertheless, GD provides a complementary perspective by penalizing individuals that do not correspond well to any target onset.

Example

In this example we omit the styles as they are not relevant for computing the distances. Consider a target piece with three onsets represented by the idealized population

$$P^* = \{y_1, y_2, y_3\},$$

where

$$y_1 = \{(\text{Violin}, D5), (\text{Flute}, F5)\}, \quad y_2 = \{(\text{Piano}, C4)\}, \quad y_3 = \{(\text{Piano}, G4)\}.$$

For the distance $d(x, x^*)$ we choose the Jaccard-distance on the instrument-pitch level defined in Section 2.3.1. Assume a population size of $|P| = 2$. We compare two different populations:

Population 1:

$$P_1 = \{x_1, x_2\}, \quad x_1 = \{(\text{Violin}, D5), (\text{Flute}, F5)\}, \quad x_2 = \{(\text{Violin}, D5), (\text{Flute}, F5)\}.$$

Both individuals perfectly match y_1 resulting in

$$\text{GD}(P_1, P^*) = \frac{1}{2}(\min(0, 1, 1) + \min(0, 1, 1)) = 0$$

and

$$\text{IGD}(P_1, P^*) = \frac{1}{3}(\min(0,0) + \min(1,1) + \min(1,1)) = \frac{2}{3}.$$

Population 2:

$$P_2 = \{x_1, x_2\}, \quad x_1 = \{(\text{Flute}, F5)\}, \quad x_2 = \{(\text{ Piano}, C4), (\text{ Piano}, G4)\}.$$

Here, x_1 provides an approximate version of y_1 while x_2 includes y_2 and y_3 and therefore serves as a reasonable approximation for both. This results in

$$\text{GD}(P_1, P^*) = \frac{1}{2}(\min(0.5,1,1) + \min(1,0.5,0.5)) = 0.5$$

and

$$\text{IGD}(P_1, P^*) = \frac{1}{3}(\min(0.5,1) + \min(1,0.5) + \min(1,0.5)) = 0.5.$$

This example demonstrates that GD favours precise approximations of individual onsets, while IGD rewards coverage of the full set of onsets. For the task considered in this work, where capturing the complete musical structure is essential, IGD provides the more informative evaluation.

2.3.5 Shapley Value

Let $F = \{f_1, \ldots, f_{100}\}$ denote the set of fitness functions. Selecting a subset of objectives to optimise simultaneously is infeasible by exhaustive search, since the number of possible combinations is $2^{100} - 1 \approx 1.2 \cdot 10^{30}$. One selection strategy ranks fitness functions by their standalone contribution, defined as

$$\text{contr}_{st}(f_i, F) = s(f_i), \tag{2.70}$$

where $s(f_i)$ denotes the success rate of a fitness function f_i and st stands for standalone. This approach does not account for the complementarity of fitness functions.

To address this, the marginal contribution of a fitness function measures how much each fitness function contributes to the over all performance. For a given target signal and a set of fitness functions $C \subseteq F$ we consider a run successful if least one fitness function in the set C was successful for the given target. Consequentially, the success rate $s(C)$ is the number of successful runs of C divided by the total number of runs. Using this notation, the marginal contribution of f_i w.r.t. F is defined as

$$\text{contr}_{ma}(f_i, F) = s(F) - s(F \setminus \{f_i\}), \tag{2.71}$$

where ma stands for marginal. This metric penalises highly correlated fitness functions (Fréchette et al., 2016). For instance, if 2 very strong fitness functions perform exactly the same on the given dataset then both fitness functions would have 0 marginal contribution and would be excluded for the MOEA. Furthermore, with a dataset of 100 signals and 100 fitness functions, the marginal contribution of a single function with respect to the full set is typically very small.

Shapley values which were originally defined for coalitional games provide a measure of contribution overcoming these limitations. Formally, a coalitional game is defined as a pair (N, v) where $N = \{1, ..., n\}$ is a set of n players and $v : 2^N \to \mathbb{R}$ is a characteristic function that maps each subset (or coalition) of players $C \subseteq N$ to a real number $v(C)$, known as the value of a coalition C. Further, let Π^N denote the set of all permutations of the set $\{1, \ldots, N\}$. For a given permutation $\pi \in \Pi^N$ let $C_i^\pi = \{j \in N : \pi(j) < \pi(i)\}$, where $\pi(i)$ denotes the position of i in π. Then, the Shapley value of player $i \in N$ is defined as:

$$\phi_i = \frac{1}{n!} \sum_{\pi \in \Pi^N} \left(v\left(C_i^\pi \cup \{i\}\right) - v\left(C_i^\pi\right) \right). \tag{2.72}$$

In our context, each fitness function corresponds to a player ($N = F$) and the coalition value is the success rate ($v(C) = s(C)$). Note that $v\left(C_i^\pi \cup \{f_i\}\right) - v\left(C_i^\pi\right)$ is the marginal contribution of f_i with respect to C_i^π. Therefore, the Shapley value can be understood as the average marginal contributions over the possible coalitions of each size. It incorporates both the standalone contributions $\text{contr}_{st}(f_i, F)$ and marginal contributions $\text{contr}_{ma}(f_i, F)$ (Fréchette et al., 2016).

The Shapley value is the only function satisfying the following desirable properties of coalitional games (Shapley, 1953):

1. **Efficiency**
   The total payoff is distributed among all players:

   $$\sum_{i \in N} \phi_i(v) = v(N). \tag{2.73}$$

   In our setting this means that the sum of the Shapley values is the success rate of all fitness functions ($s(F)$).

2. **Symmetry**
   If players i and j satisfy

   $$v(S \cup \{i\}) = v(S \cup \{j\}) \tag{2.74}$$

   for all $S \subseteq N \setminus \{i, j\}$, then

   $$\phi_i(v) = \phi_j(v). \tag{2.75}$$

   In our setting this means that if two fitness function perform identical in that they make the same marginal contributions to every portfolio, then they receive the same Shapley value.

3. **Linearity**
   For any two coalitional values v and w we define $(v + w)(C) = v(C) + w(C)$, then:

   $$\phi_i(v + w) = \phi_i(v) + \phi_i(w) \tag{2.76}$$

   for every $i \in N$. Also, for any real number a,

   $$\phi_i(av) = a\, \phi_i(v) \tag{2.77}$$

   This means that the Shapley value is invariant with respect to scaling of the success rate. In practice, scaling of the success rate might occur by changing the parameters of the EA(e.g. more generations lead to higher success rates). However, this will not effect the ranking of the fitness functions.

4. **Null Player**
   If player i satisfies

   $$v(S \cup \{i\}) = v(S) \tag{2.78}$$

   for all $S \subseteq N \setminus \{i\}$, then

   $$\phi_i(v) = 0. \tag{2.79}$$

   This means that if a fitness function does not impact the performance of any subset then it receives a Shapley value of 0.

In addition to these four properties the Shapley value satisfies various other properties specified in (Wiese, 2005). As computing the exact Shapley values is intractable for 100 fitness functions ($100! \approx 9.3 \cdot 10^{157}$), we approximate the Shapley values by random sampling of 100 000 permutations.

Example

As a small example we consider a task where we want to select 2 out of 3 fitness functions. Consider the three fitness functions

$$F = \{f_1, f_2, f_3\}$$

and four target signals

$$T = \{t_1, t_2, t_3, t_4\}.$$

The individual successes are given in Table 2.8, where a 1 stands for a successful run and a 0 denotes an unsuccessful run.

Table 2.8: Successes of the three fitness functions.

	t_1	t_2	t_3	t_4
f_1	1	1	1	0
f_2	0	1	1	0
f_3	0	0	0	1

Coalition Success Rates

$$\begin{aligned} s(\{f_1\}) = \frac{3}{4}, \quad s(\{f_2\}) = \frac{2}{4}, \quad s(\{f_3\}) = \frac{1}{4}, \\ s(\{f_1, f_2\}) = \frac{3}{4}, \quad s(\{f_1, f_3\}) = 1, \quad s(\{f_2, f_3\}) = \frac{3}{4}, \\ s(\{f_1, f_2, f_3\}) = 1. \end{aligned}$$

Standalone Contributions

The standalone contribution of a fitness function f_i is equivalent to the success rate $s(\{f_i\})$. From the perspective of a classification task we could interprete this as the accuracy of a fitness function.

$$\text{contr}_{st}(f_1) = \frac{3}{4}, \quad \text{contr}_{st}(f_2) = \frac{2}{4}, \quad \text{contr}_{st}(f_3) = \frac{1}{4}.$$

Ranking:

$$f_1 \succ f_2 \succ f_3.$$

Selecting two functions would yield $\{f_1, f_2\}$.

Marginal Contributions

We can think of the marginal contribution of a fitness function f_i as the number of targets that were correctly approximated by f_i and incorrectly approximated by all other fitness function. In this example fitness functions f_1 and f_3 each have one correctly classified target that was missed by all other fitness functions while f_2 has none. The resulting value is then divided by the total

number of targets, however, scaling is irrelevant since we are only interested in a ranking between the fitness functions.

$$\begin{aligned}
\text{contr}_{ma}(f_1, F) &= s(F) - s(\{f_2, f_3\}) = 1 - \frac{3}{4} = \frac{1}{4}, \\
\text{contr}_{ma}(f_2, F) &= s(F) - s(\{f_1, f_3\}) = 1 - 1 = 0, \\
\text{contr}_{ma}(f_3, F) &= s(F) - s(\{f_1, f_2\}) = 1 - \frac{3}{4} = \frac{1}{4}.
\end{aligned}$$

Ranking:

$$f_1 \sim f_3 \succ f_2.$$

Selecting two functions would yield $\{f_1, f_3\}$.

Shapley Values
We consider the permutations $\pi \in \Pi^3$ in the following order:

$$\pi_1 = (1,2,3), \quad \pi_2 = (1,3,2), \quad \pi_3 = (2,1,3), \quad \pi_4 = (2,3,1), \quad \pi_5 = (3,1,2), \quad \pi_6 = (3,2,1)$$

For readability reasons the factor $\frac{1}{4}$ was written in front of the bracket. This factor is present in all success rates since the total number of targets is 4.

$$\begin{aligned}
\phi_1 &= \frac{1}{3!} \cdot \frac{1}{4} \cdot (3+3+1+1+3+1) = \frac{1}{6} \cdot \frac{12}{4} = \frac{1}{2} \\
\phi_2 &= \frac{1}{3!} \cdot \frac{1}{4} \cdot (0+0+2+2+0+2) = \frac{1}{6} \cdot \frac{6}{4} = \frac{1}{4} \\
\phi_3 &= \frac{1}{3!} \cdot \frac{1}{4} \cdot (1+1+1+1+1+1) = \frac{1}{6} \cdot \frac{6}{4} = \frac{1}{4}
\end{aligned}$$

Ranking:

$$f_1 \succ f_2 \sim f_3.$$

There are two equal selections of fitness functions: $\{f_1, f_2\}$ or $\{f_1, f_3\}$.

Chapter 3

Single-Objective Experiment

For this experiment 100 polyphonic audio mixes were randomly generated from our sample library each containing between one and five samples following the distribution $[0.1, 0.3, 0.3, 0.2, 0.1]$. The actual distribution in the dataset can be found in Table 3.1. The EA was applied to reconstruct these target signals. Since the algorithm has access to the same library used to generate the targets, exact reconstruction of the signals was theoretically possible, allowing a precise evaluation of the ability to identify the global optimum.

Samples	Frequency
1	11
2	37
3	31
4	12
5	9

Table 3.1: Distribution of the number of samples in the single onset dataset (mean = 2.71).

Each of the 100 fitness functions was tested separately. The population size was set to $\mu = 10$, following Dettmer (2023). The success rates after 5000 generations are illustrated in Figure 3.1.

The results indicate that combinations involving the STFT with Jensen–Shannon divergence, as well as STFT with Kullback–Leibler divergence, achieve the highest success rates of 70 %. In contrast, Tonnetz, MFCC and the spectral combination features performs poorly, with near-zero successful reconstructions. One possible explanation might related to the number of dimensions in the audio features. While STFT has a default of 1025 frequency bins per frame ($1 + \frac{n_fft}{2}$), MFCC has only 20 coefficients per frame. Tonnetz has 6 dimensions per time frame and the spectral features have a total number of 11 dimensions (centroid: 1, bandwidth: 1, contrast: 7, rolloff: 1, flatness: 1). This strong dimensionality reduction may lead to a loss of detailed information, making accurate reconstruction significantly more difficult and potentially explaining the lower success rates.

These results suggest that default parameter settings-particularly regarding the number of dimensions per time frame-may not be optimal for this task. Therefore, systematically tuning the parameters of the audio features, such as the number of coefficients or frequency resolution, represents a promising direction for future research.

Using the STFT with Kullback–Leibler divergence as the fitness function, we analysed the influence of the number of samples on reconstruction performance. The results reveal a clear negative correlation: success rates decrease steadily as the number of samples increases, dropping from 100% for a single sample to 0% for five samples (Figure 3.2). As one would expect, the number of samples significantly raises the complexity of the reconstruction task, making it progressively harder for the algorithm to find accurate solutions.

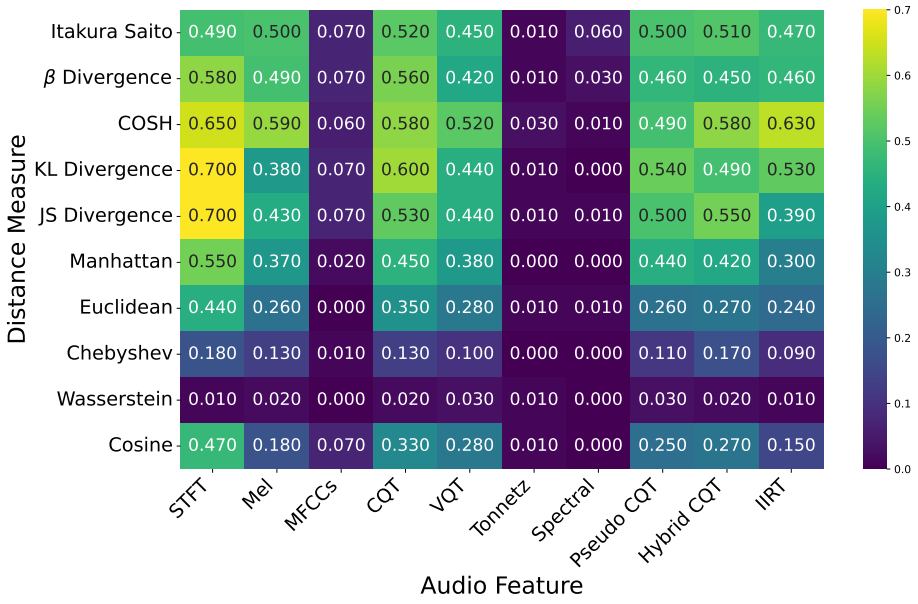


Figure 3.1: Fitness function success rate.

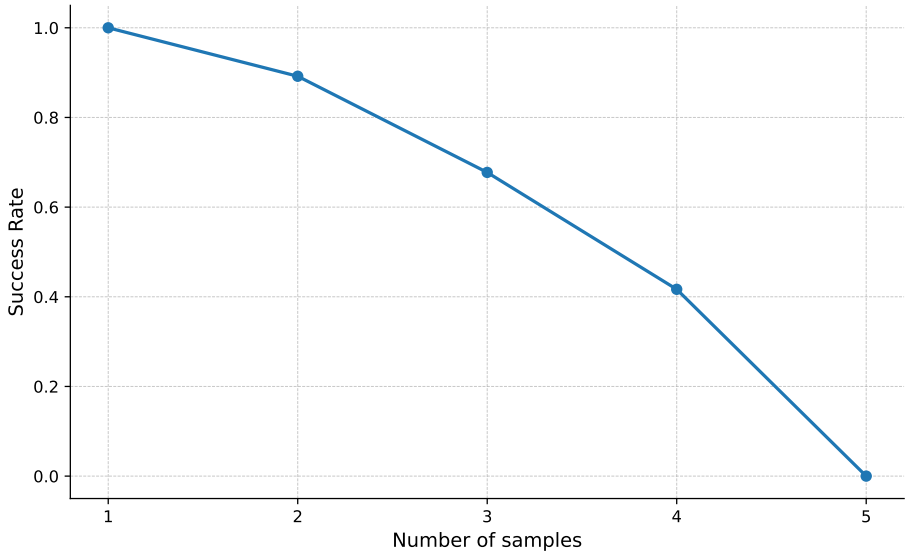


Figure 3.2: STFT + KL-divergence: success rate versus sample number.

3.1 Fitness Function Selection

In order to select suitable combinations of fitness functions for the MOEA, it is necessary to analyse the relationships between the different evaluation metrics. In particular, we are interested in identifying whether certain fitness functions behave complementary. For example, if a fitness function performs well on instrument detection but poorly on pitch detection, it would be beneficial to combine it with another fitness function that performs well on the pitch detection task.

To investigate this, we consider two instrument-based distances (Jaccard and McAdams) and four pitch-based (chord) distances (Jaccard, average Hausdorff Ginsel, average Hausdorff and L_2^{88}). For each fitness function the metrics were averaged across all 100 single onsets. Figure 3.3 shows a strong linear relationship between all considered metrics. The pairwise correlations between the metrics are consistently high, with the lowest pairwise correlation exceeding 0.9. This indicates that fitness functions that perform well with respect to one metric also perform well across all others. Consequently, there is no observable trade-off between pitch-related and instrument-related objectives.

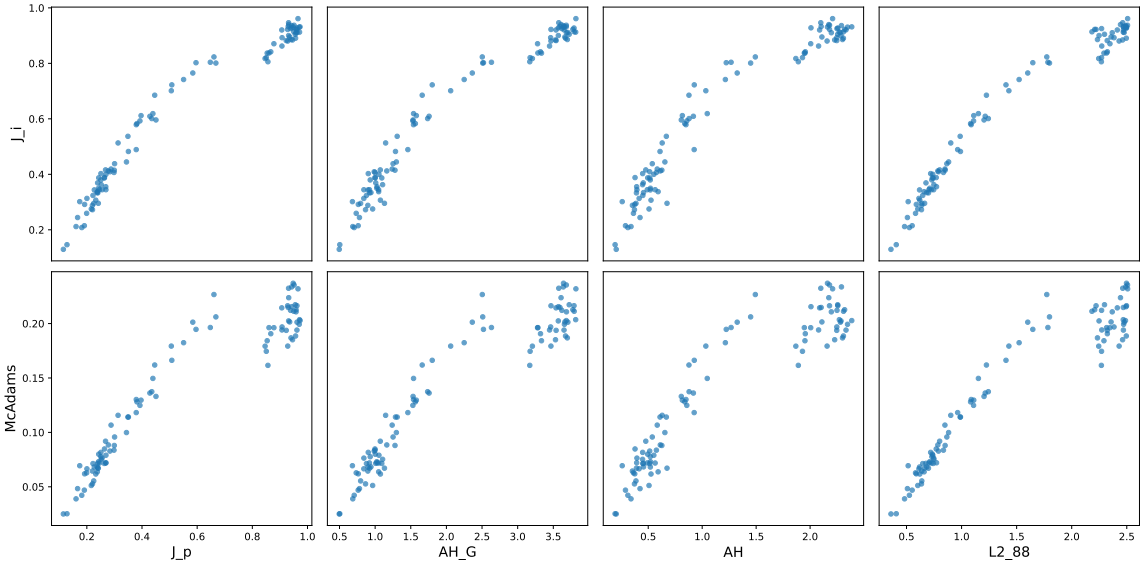


Figure 3.3: Scatter matrix of fitness functions using instrument distances (Jaccard and McAdams) on the y-axis and chord distances (Jaccard, average Hausdorff Ginsel, Average Hausdorff and L_2^{88}) on the x-axis.

To further examine this observation, we focus on a subset of high-performing fitness functions. Specifically, we restrict the analysis to solutions with both instrument-level and pitch-level Jaccard distances below 0.5.

The results further confirm that there is no trade-off between instrument and pitch detection (Figure 3.4). The strong performance of the STFT audio feature which was already observed in Figure 3.1 becomes evident once again. In particular, STFT with Kullback–Leibler divergence and STFT with Jensen–Shannon divergence performs best. This pattern is observed not only for the Jaccard-based measures but also across all other distance metrics considered in this thesis.

To analyse relationships between audio features, values across all distance measures and evaluation metrics are aggregated. Each audio feature is represented by a vector capturing its performance paired with each of the ten distance measures and across all evaluation metrics. Pairwise Pearson correlations are then computed between these vectors, resulting in a correlation matrix that quantifies similarity between audio features. This matrix is visualized as a heatmap shown in Figure D.1.

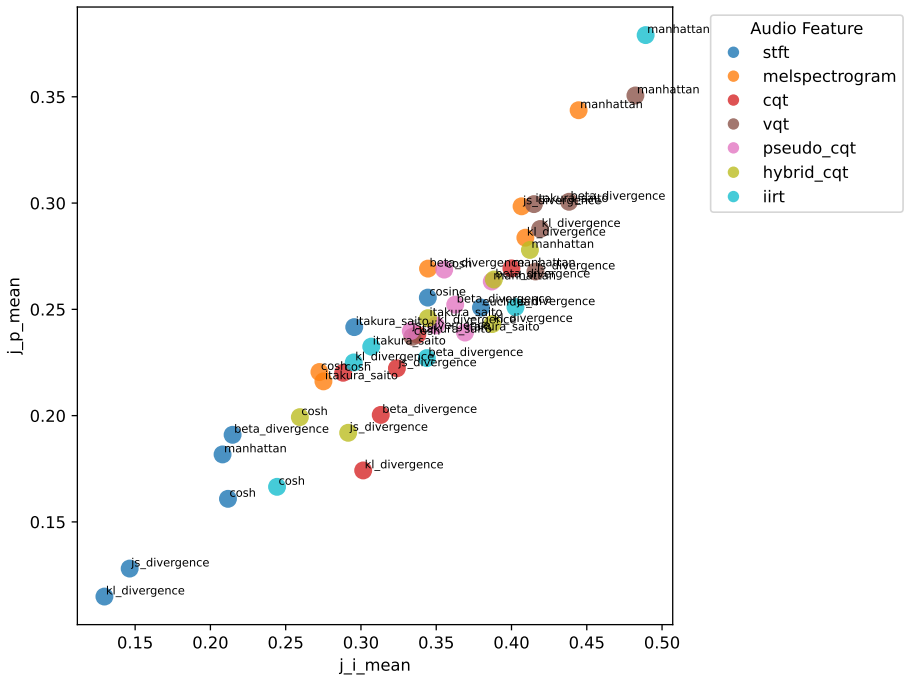


Figure 3.4: Instrument-level versus pitch-level Jaccard distance for fitness functions with $J_i \leq 0.5$ and $J_p \leq 0.5$

Rather than revealing complementary behaviour, the correlation matrix indicates the presence of two distinct groups of audio features with high intra-group correlation. One group consists of consistently well-performing features (STFT, CQT, hybrid CQT, IIRT, Melspectrogram, VQT, pseudo CQT), while the other group consists of features with comparatively poor performance (MFCC, Tonnetz, Spectral).

This separation is also reflected in the scatter matrix (Figure 3.3), where the well-performing features form a cluster in the lower-left region of the plots, corresponding to low distances across all metrics. In contrast, the poorly performing features appear as a separate cluster in the upper-right region, indicating consistently higher distances.

As there are no significant complementarities at the level of the average metrics across all onsets, a more detailed analysis is required. In particular, it is possible that different fitness functions yield similar overall values but differ in terms of which individual onsets are correctly reconstructed. We therefore proceed to analyse performance at the level of individual onsets to determine whether such differences exist despite the high global correlation between the metrics.

Figure 3.5 provides a more fine-grained view of fitness function performance at the level of individual onsets. A more detailed comparison of all fitness functions regarding all 100 single onsets is presented in C.1. The combination of STFT with Kullback–Leibler divergence, which is one of the two best-performing fitness functions, successfully reconstructs 70 out of 100 target onsets.

Importantly, among the 30 onsets for which this fitness function fails, 24 are successfully reconstructed by at least one of the remaining fitness functions. This implies that, when considering the set of all fitness functions F, a total of 94 out of 100 onsets can be reconstructed successfully, i.e., $s(F) = 0.94$. These results indicate that, despite the lack of complementariness at the level of aggregate metrics, there exists a form of complementariness at the level of individual onsets. Different fitness functions succeed on different subsets of onsets, suggesting that their strengths are not entirely redundant.

This observation motivates the use of Shapley values as a measure of complementariness. Each fitness function in the MOEA can be interpreted as a contributor to a portfolio and the overall

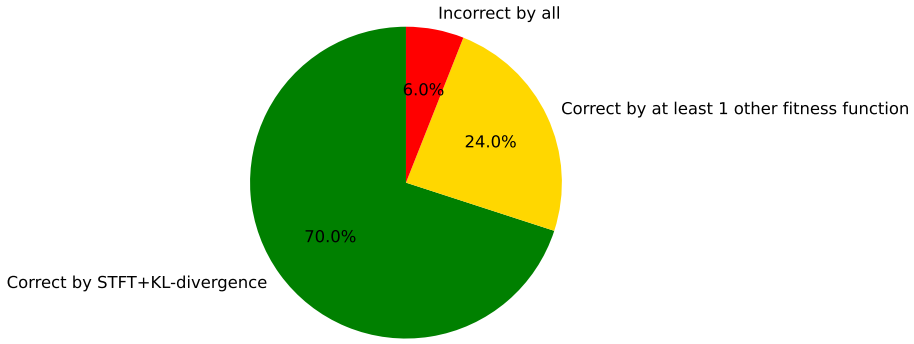


Figure 3.5: Fitness function success rate in comparison to STFT + KL-divergence.

performance is determined by their combined ability to reconstruct onsets. This is analogous to the setting in Fréchette et al. (2016), where a portfolio of SAT solvers is evaluated across problem instances and Shapley values are used to quantify the contribution of each solver to the overall performance. Similarly, we approximate Shapley values to assess the contribution of each fitness function with respect to the set of all fitness functions, in order to select those with the highest contribution for the MOEA.

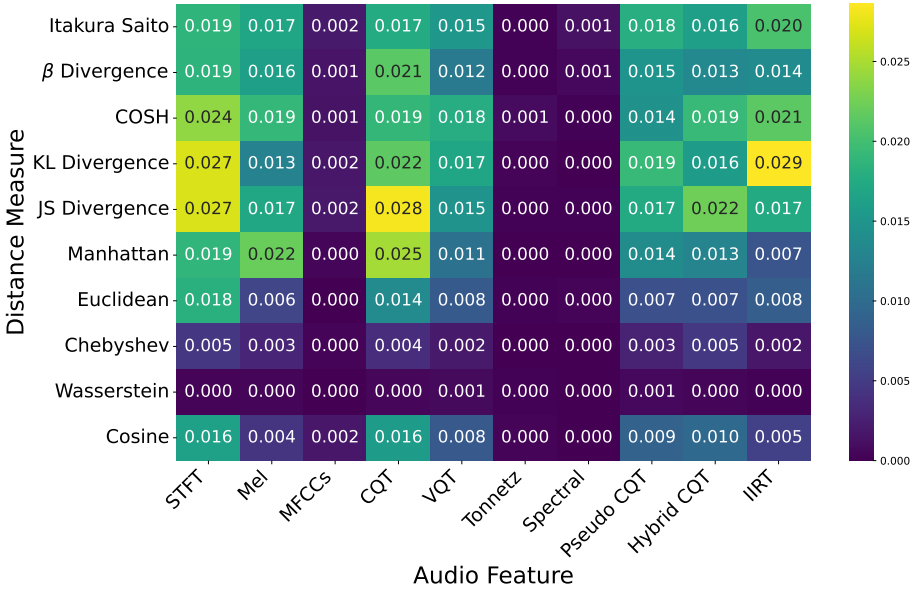


Figure 3.6: Shapley values of the fitness function.

The Shapley value heatmap (Figure 3.6) closely resembles the success rate heatmap (Figure 3.1). This is expected, as the individual success rates of the fitness functions contribute directly to the Shapley value computation (see Section 2.3.5). However, several notable differences can be observed. While STFT combined with Kullback–Leibler divergence and Jensen–Shannon divergence achieves the highest success rates, these fitness functions rank only third and fourth with respect to Shapley value. In contrast, IIRT combined with Kullback–Leibler divergence achieves the highest Shapley value despite ranking only 14$^{\text{th}}$ in terms of success rate. Similarly, CQT combined with Jensen–Shannon divergence attains the second-highest Shapley value while ranking fifth in success rate.

These discrepancies highlight that the Shapley value captures not only individual performance but also the marginal contribution of a fitness function within the set of all fitness functions. In

particular, fitness functions with moderate standalone performance can still provide substantial value if they successfully reconstruct onsets that are missed by others.

The ten best fitness functions ranked by success rate and by Shapley value are shown in Tables 3.2 and 3.3, respectively.

Table 3.2: Top 10 fitness functions by success rate.

Feature	Distance	Success Rate
STFT	KL	0.70
STFT	JS	0.70
STFT	COSH	0.65
IIRT	COSH	0.63
CQT	KL	0.60
Melspectrogram	COSH	0.59
Hybrid CQT	COSH	0.58
CQT	COSH	0.58
STFT	Beta	0.58
CQT	Beta	0.56

Table 3.3: Top 10 fitness functions by Shapley value.

Feature	Distance	Shapley
IIRT	KL	0.02860
CQT	JS	0.02821
STFT	JS	0.02699
STFT	KL	0.02697
CQT	Manhattan	0.02471
STFT	COSH	0.02383
Hybrid CQT	JS	0.02232
Melspectrogram	Manhattan	0.02206
CQT	KL	0.02151
IIRT	COSH	0.02136

Chapter 4

MOEA Experiments

To evaluate the MOEA, we conducted two experiments. First, different configurations of the MOEA were evaluated on the relatively simple task of approximating single polyphonic onsets. Subsequently, the MOEA was tested on the more challenging task of approximating full musical pieces.

For this more realistic setting, we use 20 artificially generated musical pieces with precise annotations from the AAM dataset (Ostermann et al., 2023). The selection of pieces follows the experimental setup of Dettmer (2023).

4.1 Single Onset Approximation

In this experiment, we investigate three aspects of the MOEA. First, we compare different algorithms for the environmental selection phase. As introduced in Section 2.2.5, two algorithms are considered:

1. **NSGA-II**: This algorithm uses non-dominated sorting to rank individuals and employs a crowding distance to break ties.
2. **SMS-EMOA**: This algorithm selects individuals based on their hypervolume contribution.

For both algorithms, default parameters are used.

Second, we compare three strategies for selecting combinations of fitness functions to be optimized simultaneously:

1. **Random selection**: n fitness functions are randomly chosen from the 50 best fitness functions based on success rate. This strategy serves as a baseline.
2. **Standalone contribution selection**: Fitness functions are ranked by their success rates (i.e., standalone contribution), and the top n are selected. The ten best fitness functions are listed in Table 3.2.
3. **Shapley value selection**: The top n fitness functions are selected based on their approximated Shapley values, capturing their complementary contributions. The ten best fitness functions are listed in Table 3.3.

Finally, we vary the number of objectives from 2 to 5. For example, SMS-EMOA with standalone contribution selection and four objectives simultaneously optimizes the fitness functions STFT +

KL-divergence, STFT + JS-divergence, STFT + Cosh and IIRT + Cosh. Increasing the number of objectives from four to five while keeping all other settings fixed results in the addition of CQT + KL-divergence to the set of fitness functions.

For this configuration, the convergence behaviour is shown in Figure 4.1. It can be observed that all fitness functions are optimized simultaneously, despite differences in their scales.

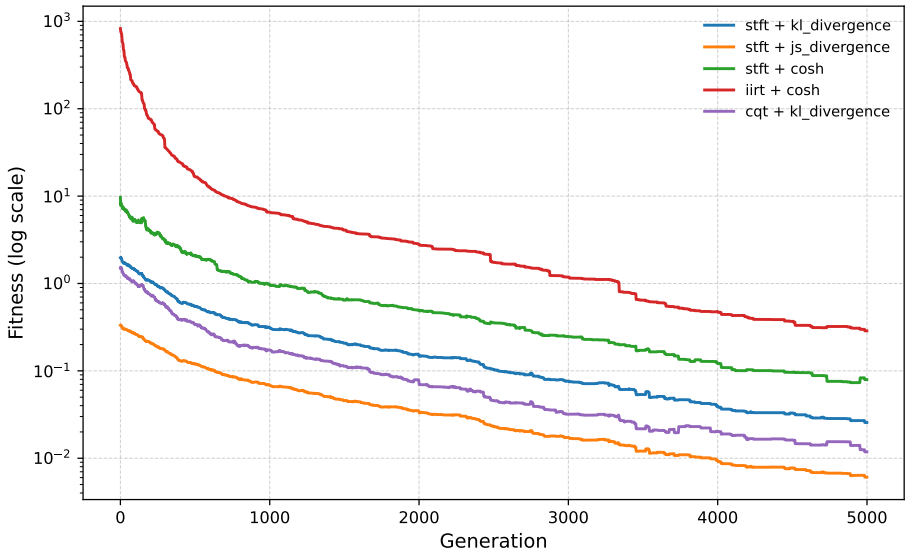


Figure 4.1: SMS-EMOA fitness convergence for 5 objectives on a logarithmic scale.

For each combination of algorithm, selection strategy and number of objectives, ten runs are performed to account for randomness. In each run, the initial population is fixed to ensure a fair comparison. To maintain comparability with the single-objective baseline, the same hyperparameters as in the single-objective experiment (Section 3) are used, including a population size of 10.

As shown in Figure 4.2, selection based on standalone contribution achieves the highest average success rate across all numbers of objectives and for both algorithms. Although Shapley-value-based selection performs slightly worse on average, it consistently outperforms random selection.

Comparing the algorithms, SMS-EMOA performs slightly better than NSGA-II overall. In general, the success rate increases with the number of objectives for both algorithms when considering two to four objectives. However, when adding the fifth objective, the success rate of SMS-EMOA increases further, whereas the performance of NSGA-II decreases across all three selection strategies.

The highest average success rate of 79% is achieved by SMS-EMOA with four or five objectives using standalone contribution selection, exceeding the maximum success rate of 70% obtained by the single-objective EA.

To assess whether this improvement is statistically significant, hypothesis testing is conducted. The hypotheses are defined as:

- H_0: $\mu \leq 0.7$
- H_1: $\mu > 0.7$

where μ denotes the average success rate. For each combination of algorithm, selection strategy and number of objectives, a one-sided t-test is performed. The p-value is defined as

$$p = P(T \geq t_{\text{obs}} \mid H_0), \tag{4.1}$$

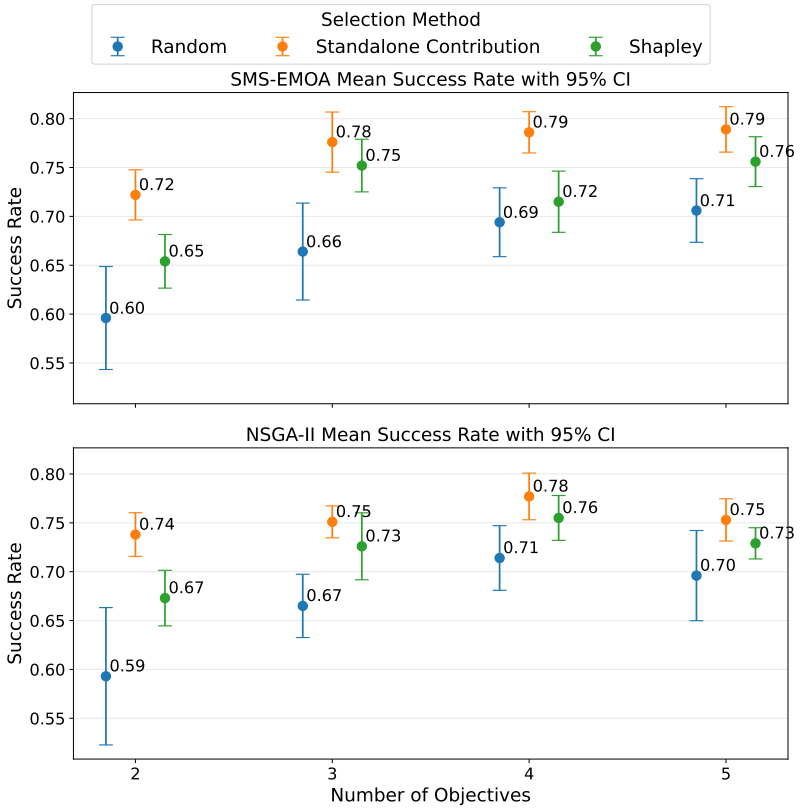


Figure 4.2: Average success rate across all 100 onsets for NSGA-II and SMS-EMOA

where T denotes the test statistic under the null hypothesis and t_{obs} is the observed value of the test statistic.

Intuitively, this means that the p-value measures how likely it is to observe a success rate at least as high as the one obtained, assuming that there is no true improvement over the single-objective EA (i.e., assuming that H_0 holds). Consequently, a small p-value indicates that such an improvement would be unlikely to occur by chance alone, providing strong evidence against H_0 and in favour of the tested configuration achieving a higher success rate.

In practice, a significance level of $\alpha = 0.05$ is commonly used. This means that a p-value below this threshold is considered statistically significant and indicates an improvement over the single-objective baseline.

Table 4.1 shows the corresponding p-values and significance decisions. The strong impact of the objective selection strategy is evident. Standalone contribution selection yields statistically significant improvements over the single-objective baseline in all cases. In contrast, Shapley-value-based selection achieves significant improvements for half of the configurations. Moreover, simply increasing the number of objectives without a principled selection strategy does not lead to improvements, as random selection never results in statistically significant gains.

Table 4.1: Hypothesis test results comparing success rate against single-objective baseline (0.7).

Algorithm	Selection	Objectives	Mean success rate	p-value	Significant
NSGA-II	Random	2	0.593	0.996335	×
NSGA-II	Random	3	0.665	0.981455	×
NSGA-II	Random	4	0.714	0.181660	×
NSGA-II	Random	5	0.696	0.575562	×
NSGA-II	Standalone	2	0.738	0.001948	✓
NSGA-II	Standalone	3	0.751	0.000029	✓
NSGA-II	Standalone	4	0.777	0.000023	✓
NSGA-II	Standalone	5	0.753	0.000178	✓
NSGA-II	Shapley	2	0.673	0.969921	×
NSGA-II	Shapley	3	0.726	0.060066	×
NSGA-II	Shapley	4	0.755	0.000210	✓
NSGA-II	Shapley	5	0.729	0.001327	✓
SMS-EMOA	Random	2	0.596	0.999216	×
SMS-EMOA	Random	3	0.664	0.932539	×
SMS-EMOA	Random	4	0.694	0.645470	×
SMS-EMOA	Random	5	0.706	0.343254	×
SMS-EMOA	Standalone	2	0.722	0.042075	✓
SMS-EMOA	Standalone	3	0.776	0.000169	✓
SMS-EMOA	Standalone	4	0.786	0.000004	✓
SMS-EMOA	Standalone	5	0.789	0.000006	✓
SMS-EMOA	Shapley	2	0.654	0.997867	×
SMS-EMOA	Shapley	3	0.752	0.000902	✓
SMS-EMOA	Shapley	4	0.715	0.153385	×
SMS-EMOA	Shapley	5	0.756	0.000387	✓

4.2 Full Piece Approximation

Since the MOEA significantly improved the reconstruction success rates in the single onset experiment, it is of interest to investigate whether this observation generalizes to the approximation of full pieces. For this experiment, 20 pieces from the AAM dataset were selected, and the population size was set to 300 following the experiments of Dettmer (2023).

Based on the results of the single onset approximation, the setup for approximating full pieces is adjusted. Since standalone contribution is the most effective strategy for selecting objectives, only this strategy is considered in this experiment. Furthermore, we observed a general trend that increasing the number of objectives improves performance. Therefore, the maximum number of objectives is increased to ten.

Similar to the single onset approximation, ten runs are performed for each combination of algorithm and number of objectives. For the single-objective baseline, ten runs are performed for each of the two best-performing configurations from the single onset experiment: STFT + KL-divergence and STFT + JS-divergence.

In contrast to the single onset approximation, it is no longer possible for the algorithm to reconstruct all onsets perfectly. This is mainly due to the following limitations of the algorithm:

1. **Population size**: The musical pieces consist of between 368 and 935 onsets (Table E.1). Although many onsets are repeated, the number of unique onsets still ranges from 62 to 308 per piece, which remains too large for a population size of 300.
2. **Maximum number of samples**: The algorithm is limited to a maximum of 5 samples per individual, while the musical pieces contain up to 9 samples per onset (Table E.1).

While both the population size and the number of samples could be increased, the goal of this work is not to achieve a perfect reconstruction of all onsets. Instead, the objective is to obtain a general approximation of the instruments and pitches present in a piece within a reasonable runtime, which can be used as features for downstream tasks such as genre classification or content-based music recommendation.

For the evaluation, GD (Equation 2.68) and IGD (Equation 2.69) are computed with respect to all onsets in a given piece. As the inner distance $d(x, x^*)$, the Jaccard distance on the instrument–pitch level is used, which is sufficient for a general comparison between the single-objective EA and the MOEA. Although more informative distance measures were introduced in Section 2.3.2 and Section 2.3.3, they are not used here due to their higher computational complexity, which arises from their combinatorial nature. Evaluating GD and IGD with these distances may provide further insights and is left for future work.

To interpret the results, note that GD behaves similarly to precision, whereas IGD behaves similarly to recall. A low GD indicates that certain parts of the piece are approximated very precisely, whereas a low IGD indicates that all parts of the piece are approximated reasonably well. Consequently, there is an inherent trade-off between these two metrics. For the task of approximating an entire piece with a limited number of individuals, IGD is the more relevant metric, as the goal is to cover all parts of the piece. For both metrics, lower values indicate better performance.

Figure 4.3 shows that both algorithms behave similarly up to four objectives, while SMS-EMOA achieves lower GD values for larger numbers of objectives. This difference becomes more pronounced as the number of objectives increases. A possible explanation for this behaviour is related to population diversity and is discussed later in this section.

Another notable observation concerns the effect of specific objectives. The GD increases when the fourth objective (IIRT + COSH-distance) is added and decreases again after adding the fifth objective (CQT + KL-divergence). This suggests that the inclusion of IIRT + COSH-distance has a negative impact on precision, which can be compensated by adding more informative objectives.

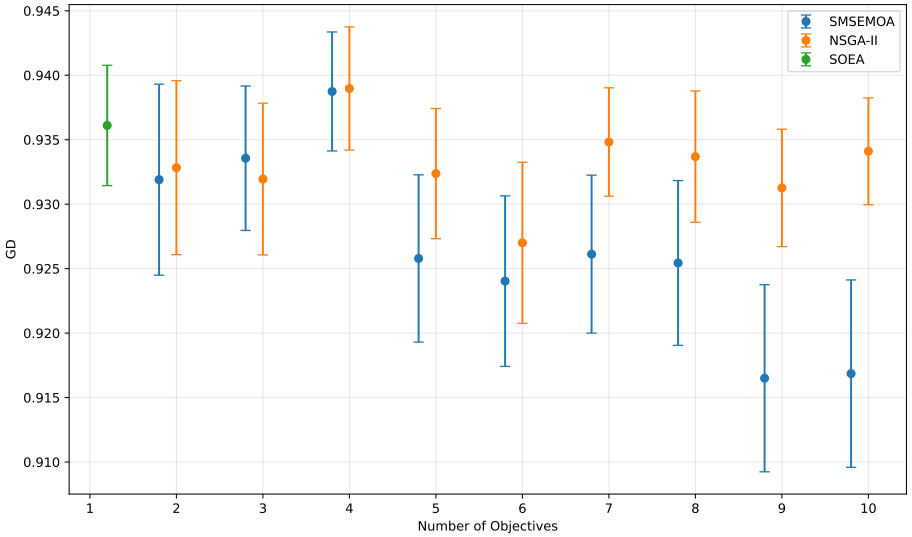


Figure 4.3: Average GD across all pieces for the single-objective evolutionary algorithm (SOEA), NSGA-II and SMS-EMOA

Statistical significance is assessed using a one-sided Welch t-test with $\alpha = 0.05$. The hypothesis are:

- H_0: The mean GD of the MOEA is not better than the single-objective baseline
- H_1: The mean GD of the MOEA is better than the single-objective baseline

The corresponding p-values and significance decisions are presented in Table 4.2.

Table 4.2: GD: Hypothesis test results comparing MOEA against single-objective baseline

Algorithm	Objectives	Mean GD	p-value	Significant
NSGA-II	2	0.932830	0.216029	×
NSGA-II	3	0.931945	0.137868	×
NSGA-II	4	0.938974	0.800843	×
NSGA-II	5	0.932373	0.142790	×
NSGA-II	6	0.927004	0.011014	✓
NSGA-II	7	0.934821	0.343683	×
NSGA-II	8	0.933688	0.245526	×
NSGA-II	9	0.931263	0.072149	×
NSGA-II	10	0.934102	0.263734	×
SMSEMOA	2	0.931901	0.172345	×
SMSEMOA	3	0.933564	0.246352	×
SMSEMOA	4	0.938743	0.785327	×
SMSEMOA	5	0.925791	0.005691	✓
SMSEMOA	6	0.924031	0.001747	✓
SMSEMOA	7	0.926123	0.005508	✓
SMSEMOA	8	0.925441	0.004137	✓
SMSEMOA	9	0.916501	0.000005	✓
SMSEMOA	10	0.916856	0.000007	✓

For SMS-EMOA with more than four objectives a significant improvement over the single-objective baseline can be observed. NSGA-II only achieves significant improvements for 6 objectives. The best configuration with respect to GD is SMS-EMOA with 9 or 10 objectives, reducing the average

GD from approximately 0.936 in the single-objective case to 0.917. While this value may still appear relatively high, it should be noted that perfect reconstruction is not achievable and that the Jaccard distance provides only a basic measure of similarity.

For IGD, the results are presented in Figure 4.4. Similar to GD, both algorithms behave similarly up to five objectives. For higher numbers of objectives, NSGA-II performs better. This is consistent with the observed trade-off between GD and IGD.

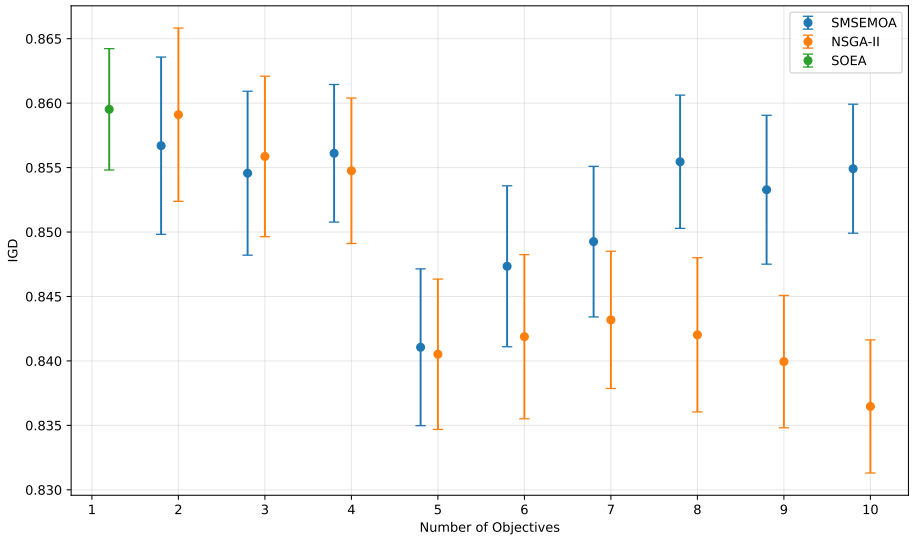


Figure 4.4: Average IGD across all pieces for the single-objective evolutionary algorithm (SOEA), NSGA-II and SMS-EMOA.

A substantial decrease in IGD is observed when adding the fifth objective (CQT + KL-divergence). For SMS-EMOA, this configuration yields the best IGD performance among those evaluated. For NSGA-II, a lower IGD is only achieved when increasing the number of objectives to nine. This indicates that CQT + KL-divergence is particularly beneficial for approximating full pieces.

Table 4.3: IGD: Hypothesis test results comparing MOEA against single-objective baseline

Algorithm	Objectives	Mean IGD	p-value	Significant
NSGA-II	2	0.859105	0.460236	×
NSGA-II	3	0.855864	0.178341	×
NSGA-II	4	0.854752	0.100913	×
NSGA-II	5	0.840517	$4.24 \cdot 10^{-7}$	✓
NSGA-II	6	0.841881	$7.26 \cdot 10^{-6}$	✓
NSGA-II	7	0.843184	$3.75 \cdot 10^{-6}$	✓
NSGA-II	8	0.842026	$3.85 \cdot 10^{-6}$	✓
NSGA-II	9	0.839947	$2.52 \cdot 10^{-8}$	✓
NSGA-II	10	0.836468	$9.99 \cdot 10^{-11}$	✓
SMSEMOA	2	0.856700	0.252588	×
SMSEMOA	3	0.854565	0.108993	×
SMSEMOA	4	0.856110	0.172838	×
SMSEMOA	5	0.841060	$1.53 \cdot 10^{-6}$	✓
SMSEMOA	6	0.847344	0.001147	✓
SMSEMOA	7	0.849254	0.003649	✓
SMSEMOA	8	0.855453	0.126195	×
SMSEMOA	9	0.853282	0.049871	✓
SMSEMOA	10	0.854913	0.093600	×

For IGD we again perform hypothesis testing to confirm the significance of the improvements over the single-objective baseline (Table 4.3). The formulation of H_0 and H_1 is similar to the hypothesis testing for GD. The single-objective baseline achieved a IGD of approximately 0.860.

Both algorithms again show a clear improvement over the single-objective baseline. NSGA-II is significantly better than the single-objective baseline for more than four objectives while SMS-EMOA is significantly better for four out of ten configurations The best configuration with respect to IGD is NSGA-II with 10 objectives, reducing the average IGD from approximately 0.860 to 0.836.

A possible explanation for the observed behaviour of GD and IGD lies in the population diversity. To illustrate this, consider two extreme cases. In the first case, all individuals are identical and represent a single onset. This results in minimal diversity, a GD of zero and an IGD close to one. In the second case, the population contains a very large number of highly diverse individuals covering the entire label space. This leads to maximal diversity, an IGD close to zero and a GD close to one. In practice, both extremes are undesirable and a balance between diversity and precision is required.

From this example it becomes very clear that higher population diversity is expected to result in a lower (better) IGD value and a higher (worse) GD value. To quantify population diversity, we again use the Jaccard distance. If two individuals are very similar, they have a low Jaccard distance. Therefore, we compute the average pairwise Jaccard distance across all pairs of individuals in the population to measure overall similarity. This average pairwise Jaccard distance, used as a measure of population diversity, is illustrated in Figure 4.5. A higher value indicates greater diversity.

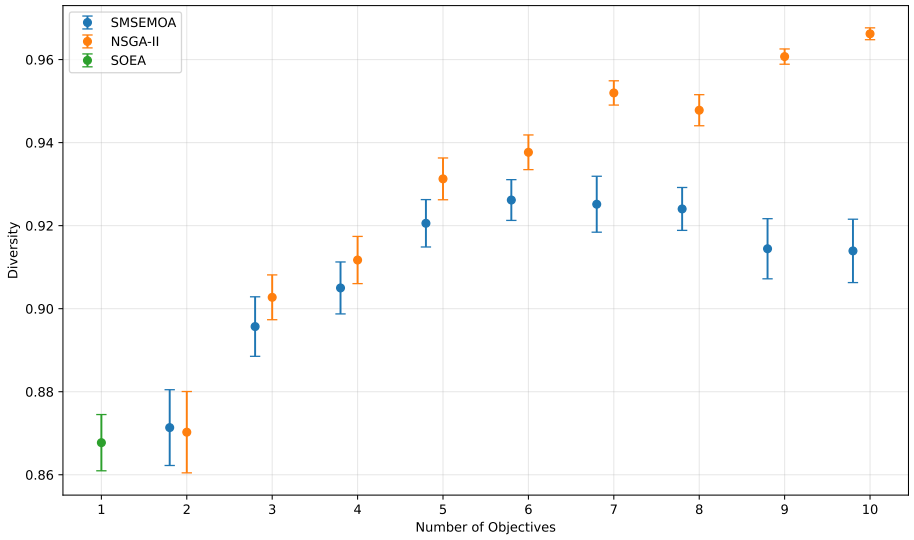


Figure 4.5: Diversity by number of objectives.

As expected, increasing the number of objectives leads to higher diversity, particularly for NSGA-II. In contrast, the diversity of SMS-EMOA peaks at around six to seven objectives and then decreases. This behaviour mirrors the observed trends in GD and IGD. For SMS-EMOA, decreasing diversity is associated with lower GD and higher IGD, whereas NSGA-II shows a more continuous increase in diversity and a corresponding decrease in IGD.

However, diversity alone does not fully explain the improvements achieved by the MOEA. For example, SMS-EMOA with 10 objectives exhibits both higher diversity and better GD compared to configurations with fewer objectives. This indicates that both diversity and precision can improve simultaneously, reflecting an overall enhancement in approximation quality. A similar observation holds for NSGA-II, which achieves improvements in both GD and IGD compared to the single-objective baseline. Figures F.1 and F.2 show substantial variation in difficulty across pieces and differences in the best-performing configurations. A detailed analysis of these differences is left for future work.

Chapter 5

Discussion

For the first experiment of this thesis, we constructed and analysed fitness functions in the single-objective EA. To this end, ten audio features were combined with ten distance measures, resulting in a total of 100 candidate fitness functions. In order to keep the scope manageable, all features and distance measures were used with their default parameter settings. While this allows for a broad comparison, it also highlights a possible direction for future work, namely the systematic tuning of feature-specific parameters.

As a first step, the fitness functions were evaluated within the original single-objective EA on a simplified task, namely the reconstruction of single onsets. This setup provided a controlled environment to assess the relative performance of different feature-distance combinations and to derive insights for the subsequent multi-objective formulation. In this experiment, the combination of STFT with COSH-distance, previously identified as strong (Ginsel, 2021), was outperformed by STFT combined with Kullback-Leibler divergence and Jensen–Shannon divergence, achieving success rates of up to 70 % compared to 65 % for STFT with COSH-distance. In contrast, features such as MFCC, Tonnetz and spectral combinations performed poorly, which may be related to their lower dimensionality.

Significant differences were also observed among distance measures. The Chebychev distance performed particularly poorly, which can be explained by its definition as the maximum absolute difference across all entries of a vector. This effectively only considers a single dimension from high-dimensional feature vectors and leads to a substantial loss of information. The poor performance of the Wasserstein distance, on the other hand, could not be conclusively explained and remains an open question.

Interestingly, no meaningful trade-off between instrument and pitch detection was observed. Fitness functions that performed well for pitch detection also performed well for instrument detection. For this comparatively simple task of approximating single onsets more advanced perceptual evaluation metrics did not provide additional insights beyond simple set-based measures such as the Jaccard distance. Since solutions were entirely correct many times, Jaccard-distance appears to be well suited for this task. However, for more complex tasks, where exact approximations are rare, the more sophisticated metrics introduced in this work are expected to become more informative.

To analyse the contribution of individual fitness functions, Shapley values were computed based on single runs. While this provided useful indications, the results may be influenced by stochastic effects inherent to EAs. A more robust analysis would require multiple runs, which was not feasible within the scope of this thesis due to computational constraints. Using the Shapley value in order to select complementary objectives for the MOEA showed only limited success while still being better than selecting random objectives.

The method for selecting objectives appears to be very important, as there are big differences between the three selection strategies that were evaluated. Selection based on the standalone success

rate of fitness functions proved highly effective. Using this approach, the success rate increased significantly from 70 % in the single-objective baseline to an average of 79 % in the multi-objective setting. Two multi-objective algorithms were considered, namely NSGA-II and SMS-EMOA. In the single-onset experiment, SMS-EMOA outperformed NSGA-II, while NSGA-II still outperformed the single-objective baseline. Furthermore, results suggest that increasing the number of objectives can be beneficial. Therefore, investigating higher number of objectives could be a direction for future work. However, SMS-EMOA and NSGA-II scale poorly with the number of objectives. Thus, it might be necessary to use algorithms that are specifically designed for many-objective problems such as NSGA-III (Deb & Jain, 2013).

For the more complex task of approximating full musical pieces, a significant improvement over the single-objective baseline was again observed. The evaluation was based on GD and IGD, using the Jaccard distance as the underlying metric. Although more informative distance measures were introduced earlier in this work, they were not used in this evaluation due to their high computational complexity. Incorporating these metrics remains an important direction for future research.

When comparing NSGA-II and SMS-EMOA on this task, distinct differences emerged. SMS-EMOA achieved better results in terms of GD, indicating more precise approximations, while NSGA-II performed better with respect to IGD, suggesting a more diverse and comprehensive coverage of the whole piece. This observation is consistent with the measured population diversity: NSGA-II maintained significantly higher diversity for larger numbers of objectives, which correlates with improved IGD but slightly worse GD. Overall, both algorithms outperformed the single-objective EA in terms of GD and IGD when using ten objectives, demonstrating the effectiveness of the multi-objective approach.

One question that remains is why NSGA-II achieves a higher population diversity than SMS-EMOA. The only difference between the two algorithms is the selection mechanism used for individuals belonging to the same Pareto front. NSGA-II uses crowding distance, which explicitly promotes a uniform distribution of solutions by favoring individuals located in sparsely populated regions of the objective space. In contrast, SMS-EMOA uses hypervolume contribution for survivor selection. In high-dimensional spaces, it is possible that extreme solutions dominate disproportionately large hypervolume regions extending towards the reference point. Consequently, SMS-EMOA might prefer extreme trade-off solutions while removing interior solutions, which could lead to population clustering and reduced overall diversity. To confirm this hypothesis, further experiments and analysis of the distribution of individuals in the objective space would be required, which is beyond the scope of this thesis.

Overall, The findings also suggest potential applications beyond reconstruction. Due to the increased diversity of populations generated by the MOEA, they may be particularly well suited for feature extraction tasks. Previous work by Dettmer (2023) indicates that a broader distribution of instruments and pitches can improve genre classification. Based on this, the proposed multi-objective approach could enhance classification performance, which represents a promising direction for future work.

Finally, it is important to note that all algorithmic parameters were adopted from the literature for the single-objective case. However, the behaviour of the multi-objective algorithms differs significantly, particularly in terms of population diversity and convergence dynamics. This suggests that dedicated hyperparameter optimization could further improve performance and should be explored in future research.

Chapter 6

Conclusions and Future Work

The central contribution of this thesis is the extension of the EA proposed by Vatolkin (2020) to a multi-objective setting, enabling the simultaneous optimisation of multiple fitness functions. 100 fitness functions, each consisting of an audio feature and a distance measure, were evaluated on a dataset of 100 polyphonic chords. The single-objective experiment revealed that STFT-based representations combined with Kullback-Leibler- or Jensen-Shannon divergence performed best achieving a success rate of 0.7.

Selecting meaningful subsets of fitness functions that are optimised simultaneously in the multi-objective case was a key challenge. For this task, we introduced three strategies:

- Random selection
- Standalone contribution selection
- Shapley value selection

For the MOEA we used two algorithms: NSGA-II and SMS-EMOA. In the single onset experiment we used between two and five objectives.

In the single onset approximation, the best single-objective configuration achieved a success rate of 70 %, while the proposed MOEA reached up to 79 %, demonstrating that jointly optimising multiple fitness functions improves reconstruction performance. In this experiment, SMSEMOA outperformed NSGA-II. Among the objective selection strategies, ranking by standalone contribution yielded the strongest results. Shapley-value-based selection still outperformed random selection and provides a principled framework for analysing objective complementarity. Overall, increasing the number of objectives improved the success rate.

In order to get insights on the performance on full musical pieces the MOEA was tested on the AAM dataset. When comparing NSGA-II and SMS-EMOA on this task, distinct differences became evident. SMS-EMOA achieved lower GD values (0.917), indicating more precise approximations and significantly outperforming the single-objective setting which achieved a GD of 0.936. On the other hand, NSGA-II performed better with respect to IGD (0.836), suggesting a more diverse and comprehensive coverage of the whole piece compared to the single-objective setting which achieved an IGD of 0.860. This observation is consistent with the measured population diversity: NSGA-II maintained a significantly higher diversity for larger numbers of objectives, which correlates with improved IGD but slightly worse GD. Overall, both algorithms outperformed the single-objective EA in terms of GD and IGD when using ten objectives, demonstrating the effectiveness of the multi-objective approach.

In our discussion of these results, we made several recommendations for future work. This includes systematic tuning of parameters for the audio feature and for the MOEA, larger objective

sets or alternative multi-objective algorithms. Using the introduced chord and instrument distance for evaluation of the full piece approximation and investigating the performance on single pieces could give further insights into the strength and weaknesses of the algorithms. Another direction for further research could be feature extraction from the MOEA for downstream tasks such as genre-classification. As mentioned in previous work (Dettmer et al., 2024), exploration of other mutation methods and extending the individual's dimensions with aspects such as loudness or the addition of audio effects could be promising directions future work.

Overall, the results indicate that multi-objective evolutionary optimisation is a promising direction for joint instrument and pitch detection. By using various audio features and distance measures, the proposed approach improves reconstruction accuracy while maintaining the flexibility and gradient-free nature of evolutionary search. The full implementation of this work is publicly available on GitHub[1] to ensure reproducibility and facilitate future research.

[1] https://github.com/MalteSchorken/MasterThesis

References

Bäck, T., Fogel, D. B., & Michalewicz, Z. (2000). *Evolutionary computation 1: Basic algorithms and operators*. Taylor & Francis.

Bartz-Beielstein, T., Branke, J., Mehnen, J., & Mersmann, O. (2014). Evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(3), 178–195.

Benetos, E., Dixon, S., Duan, Z., & Ewert, S. (2019). Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, *36*(1), 20–30.

Beume, N., Naujoks, B., & Emmerich, M. (2007). Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, *181*(3), 1653–1669.

Bhattarai, B., & Lee, J. (2023). A comprehensive review on music transcription. *Applied Sciences*, *13*(21), 11882.

Burgoyne, J. A., Fujinaga, I., & Downie, J. S. (2015). Music information retrieval. *A new companion to digital humanities*, 213–228.

Chang, S., Benetos, E., Kirchhoff, H., & Dixon, S. (2024). Yourmt3+: Multi-instrument music transcription with enhanced transformer architectures and cross-dataset stem augmentation. *Proceedings of the 2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1–6.

Cheuk, K. W., Choi, K., Kong, Q., Li, B., Won, M., Hung, A., Wang, J.-C., & Herremans, D. (2022). Jointist: Joint learning for multi-instrument transcription and its applications. *arXiv preprint arXiv:2206.10805*.

Chew, E. (2000). *Towards a mathematical model of tonality* [Doctoral dissertation, Massachusetts Institute of Technology].

Cook, L. M. (2003). The peppered moth and industrial melanism: Evolution of a natural selection case study. *Heredity*, *91*, 376–381.

Darwin, C. (1859). *On the origin of species by means of natural selection*. John Murray.

Das, S., Maity, S., Qu, B.-Y., & Suganthan, P. N. (2011). Real-parameter evolutionary multimodal optimization—a survey of the state-of-the-art. *Swarm and Evolutionary Computation*, *1*(2), 71–88.

Dash, S. K., Solanki, S. S., & Chakraborty, S. (2024). Deep convolutional neural networks for predominant instrument recognition in polyphonic music using discrete wavelet transform. *Circuits, Systems, and Signal Processing*, *43*(7), 4239–4271.

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(4), 357–366.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Deb, K., & Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE transactions on evolutionary computation*, *18*(4), 577–601.

de León, P. J. P., Iñesta, J. M., Calvo-Zaragoza, J., & Rizo, D. (2016). Data-based melody generation through multi-objective evolutionary computation. *Journal of Mathematics and Music*, *10*(2), 173–192.

Dettmer, J. (2023). *Expanding an evolutionary algorithm for the synthesis of polyphonic music* [Institute of Neuroinformatics]. Ruhr University Bochum.

Dettmer, J., Vatolkin, I., & Glasmachers, T. (2024). Weighted initialisation of evolutionary instrument and pitch detection in polyphonic music. *Proceedings of the International Conference on Artificial Intelligence in Music, Sound, Art and Design*, 114–129.

Downie, J. S. (2004). The scientific evaluation of music information retrieval systems: Foundations and future. *Computer Music Journal*, *28*(2), 12–23.

Ebeling, M. (2007). *Verschmelzung und neuronale autokorrelation als grundlage einer konsonanztheorie*. Peter Lang.

Elghamrawy, S. M., & Ibrahim, S. E. (2021). Audio signal processing and musical instrument detection using deep learning techniques. *Proceedings of the 9th International Japan-Africa Conference on Electronics, Communications, and Computations*, 146–149.

Fréchette, A., Kotthoff, L., Michalak, T., Rahwan, T., Hoos, H., & Leyton-Brown, K. (2016). Using the shapley value to analyze algorithm portfolios. *Proceedings of the AAAI Conference on Artificial Intelligence*, *30*(1).

Freitas, A., & Guimaraes, F. (2011). Melody harmonization in evolutionary music using multiobjective genetic algorithms. *Proceedings of the Sound and Music Computing Conference*.

Fujita, O. (2013). Metrics based on average distance between sets. *Japan Journal of Industrial and Applied Mathematics*, *30*(1), 1–19.

Ginsel, P. (2021). *Abstandsmaße zur evolutionären klangapproximation auf audiodaten* [Faculty of Computer Science, Algorithm Engineering (LS 11)]. Technical University of Dortmund.

Gray, A., & Markel, J. (1976). Distance measures for speech processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *24*(5), 380–391.

Grey, J. M. (1977). Multidimensional perceptual scaling of musical timbres. *Journal of the Acoustical Society of America*.

Han, Y., Kim, J., & Lee, K. (2017). Deep convolutional neural networks for predominant instrument recognition in polyphonic music. *Proceedings of the IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(1), 208–221.

Harte, C., Sandler, M., & Gasser, M. (2006). Detecting harmonic change in musical audio. *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, 21–26.

Hurajová, K., & Hutník, O. (2024). Towards measuring the distances of chords of different cardinalities. In T. Noll, M. Montiel, F. Gómez, O. C. Hamido, J. L. Besada, & J. O. Martins (Eds.), *Mathematics and computation in music* (pp. 345–359). Springer Nature Switzerland.

International Federation of the Phonographic Industry. (2025). *Global music report 2025: State of the industry*. IFPI.

Itakura, F., & Saito, S. (1968). Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*.

Jeong, J. H., & Ahn, C. W. (2015). Automatic evolutionary music composition based on multi-objective genetic algorithm. *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems - Volume 2*, 105–115.

Kostka, S., Payne, D., & Almén, B. (2018). *Tonal harmony* (8th ed.). McGraw-Hill.

Koutini, K., Schlüter, J., Eghbal-zadeh, H., & Widmer, G. (2022). Efficient Training of Audio Transformers with Patchout. *Interspeech*, 2753–2757.

Majidi, M., & Toroghi, R. M. (2023). A combination of multi-objective genetic algorithm and deep learning for music harmony generation. *Multimedia Tools and Applications*, *82*(2), 2419–2435.

McAdams, S., Winsberg, S., Donnadieu, S., Soete, G. D., & Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, *58*(3), 177–192.

McAdams, S. (2013). Musical timbre perception. *The psychology of music*, *3*(2), 177–192.

McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., & Nieto, O. (2015). Librosa: Audio and music signal analysis in python. *Proceedings of the 14th Python in Science Conference (SciPy)*, 18–24.

Menchaca-Mendez, A., & Coello Coello, C. A. (2015). Gd-moea: A new multi-objective evolutionary algorithm based on the generational distance indicator. *International conference on evolutionary multi-criterion optimization*, 156–170.

Mokoena, N., & Obagbuwa, I. C. (2025). An analysis of artificial intelligence automation in digital music streaming platforms for improving consumer subscription responses: A review. *Frontiers in Artificial Intelligence*, *7*.

Müller, M. (2015). *Fundamentals of music processing*. Springer.

Ostermann, F., Vatolkin, I., & Ebeling, M. (2023). Aam: A dataset of artificial audio multitracks for diverse music information retrieval tasks. *EURASIP Journal on Audio, Speech, and Music Processing*, *2023*(1), 13.

Petrie, M., & Halliday, T. (1994). Experimental and natural changes in the peacock's (pavo cristatus) train can affect mating success. *Nature*, *371*, 598–599.

Qin, L., Zhu, Y., Liu, S., Zhang, X., & Zhao, Y. (2025). The shapley value in data science: Advances in computation, extensions, and applications. *Mathematics*, *13*(10), 1581.

Rechenberg, I. (1971). *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution* [PhD Thesis, Department of Process Engineering]. Technische Universität Berlin.

Rocher, T., Robine, M., Hanna, P., & Desainte-Catherine, M. (2010). A survey of chord distances with comparison for chord analysis. *Proceedings of the International Computer Music Conference (ICMC)*, 187–190.

Schörkhuber, C., & Klapuri, A. (2010). Constant-q transform toolbox for music processing. *Proceedings of the 7th Sound and Music Computing Conference*, 3–64.

Schörkhuber, C., Klapuri, A., Holighaus, N., & Dörfler, M. (2014). A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. *Proceedings of the Audio Engineering Society 53rd International Conference: Semantic Audio*.

Schwefel, H.-P. (1975). *Evolutionsstrategie und numerische optimierung* [Dr.-Ing. Dissertation, Fachbereich Verfahrenstechnik]. Technische Universität Berlin.

Shapley, L. S. (1951). *Notes on the n-person game ii: The value of an n-person game* (tech. rep.). RAND Corporation.

Shapley, L. S. (1953). A value for n-person games. In H. W. Kuhn & A. W. Tucker (Eds.), *Contributions to the theory of games* (pp. 307–317, Vol. 28). Princeton University Press.

Sloss, A. N., & Gustafson, S. (2020). 2019 evolutionary algorithms review. In W. Banzhaf, E. Goodman, L. Sheneman, L. Trujillo, & B. Worzel (Eds.), *Genetic programming theory and practice xvii* (pp. 307–344). Springer.

Smith, T. B. (1993). Disruptive selection and the genetic basis of bill size polymorphism in the african finch pyrenestes. *Nature*, *363*, 618–620.

Sun, Y., Yen, G. G., & Yi, Z. (2018). Igd indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, *23*(2), 173–187.

Tian, Y., Si, L., Zhang, X., Cheng, R., He, C., Tan, K. C., & Jin, Y. (2021). Evolutionary large-scale multi-objective optimization: A survey. *ACM Computing Surveys (CSUR)*, *54*(8), 1–34.

Vatolkin, I. (2020). Evolutionary approximation of instrumental texture in polyphonic audio recordings. *Proceedings of the 2020 IEEE Congress on Evolutionary Computation*, 1–8.

Vatolkin, I., Preuß, M., Rudolph, G., Eichhoff, M., & Weihs, C. (2012). Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures. *Soft Computing*, *16*(12), 2027–2047.

Wei, B., & Gibson, J. D. (2001). *Comparison of distance measures in discrete spectral modeling* [Master's thesis, Southern Methodist University, Dallas, Texas].

Weihs, C., Jannach, D., Vatolkin, I., & Rudolph, G. (2016). *Music data analysis: Foundations and applications*. CRC Press.

Wiese, H. (2005). *Kooperative spieltheorie*. Oldenbourg Wissenschaftsverlag.

Wu, H.-C. (2021). Using shapley values and genetic algorithms to solve multiobjective optimization problems. *Symmetry*, *13*(11), 2021.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and evolutionary computation*, *1*(1), 32–49.

Zwicker, E., & Fastl, H. (2013). *Psychoacoustics: Facts and models*. Springer.

Appendix A

Statistics about the Sample Library

InstrumentName	NumberofStyles	MinimumPitch	MaximumPitch
AcousticGuitar	12	e2	e5
Balalaika	1	a2	f5
Bandura	1	c2	g6
BanjoFramus	1	a2	d6
Banjolin	1	e3	a5
Bass	2	c1	cis4
Bassoon	2	gis1	d5
Bawu	1	b3	f5
BouzoukiSakis	1	c3	b5
Cello	9	c2	gis5
CeylonGuitar	1	c2	c4
Clarinet	2	c3	fis6
Contrabassoon	1	ais0	c4
Cumbus	1	c2	c5
DallapeAccordion	1	b1	b6
Dilruba	1	c3	g5
Domra	1	c3	b5
Drums	12	(36)	(64)
DungDkarTrumpet	1	e3	a5
EgyptianFiddle	1	e2	e5
ElectricBass	7	b0	e4
ElectricGuitar	9	e2	e5
ElectricPiano	6	c1	g7
Erhu	1	a3	c6
Flute	8	a3	e7
Fujara	1	c2	c7
Horn	4	g1	b5
JinghuOperaviolin	1	a4	a6
Kantele	1	c1	c6
Melodica	1	a3	c6
MorinKhuurViolin	1	d3	c6
Oboe	2	a3	f6
Organ	6	c1	c6
Oud	1	f2	f5
Panflute	1	f2	dis5
Piano	11	a0	c8
Pinkillo	1	c5	c7
PivanaFlute	1	c5	c7
Saxophone	5	c2	f6
ScaleChangerHarmonium	1	g1	c5
Shakuhachi	1	a3	e6
Sitar	1	c3	g5
Tampura	1	f2	ais3
Tanbur	1	fis1	c5
Trombone	4	a0	g5
Trumpet	10	c3	f6
Tuba	3	gis0	e5
TurkeySaz	1	e2	e5
Ukulele	1	a3	b5
Viola	9	c3	f6
Violin	10	g3	e7

Table A.1: Instrument ranges and styles

Appendix B

Instrument-Style Distances

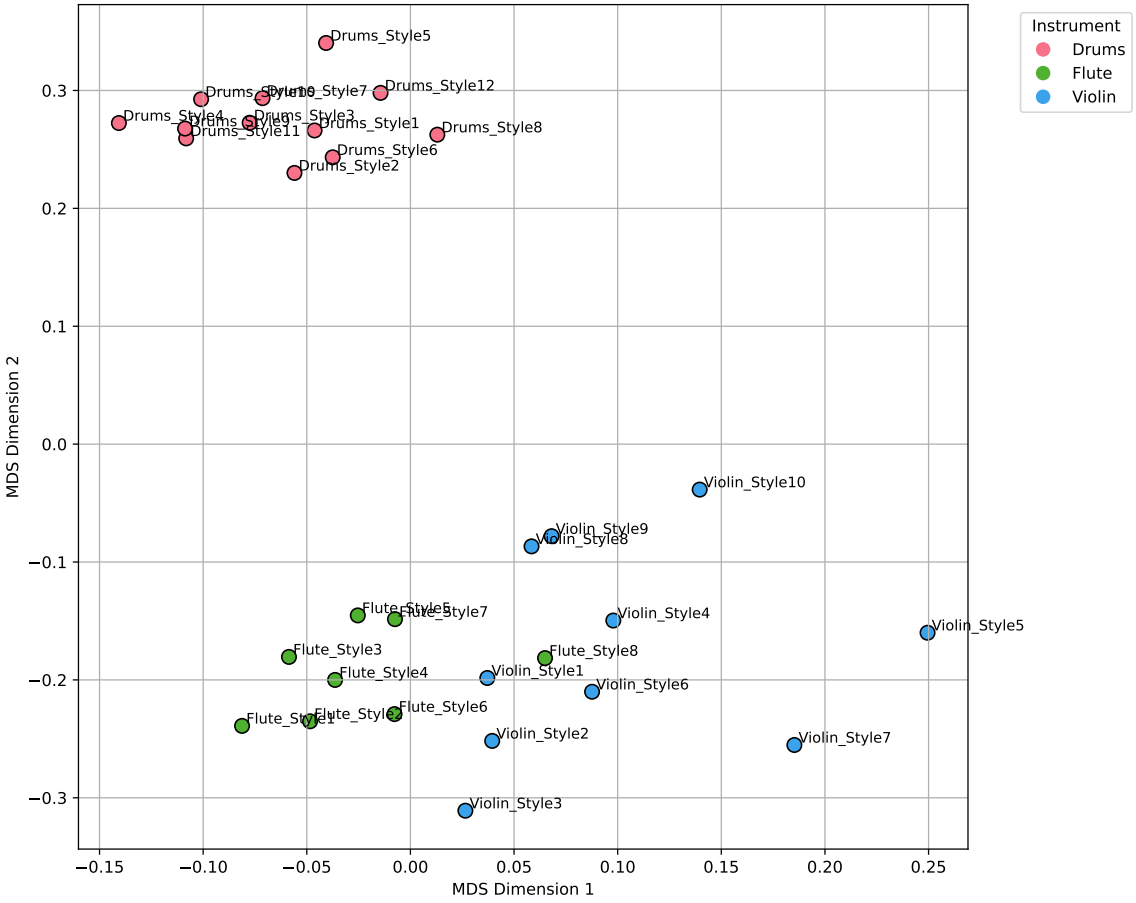


Figure B.1: MDS projection of instrument distances for drums, flute and violin with all styles.

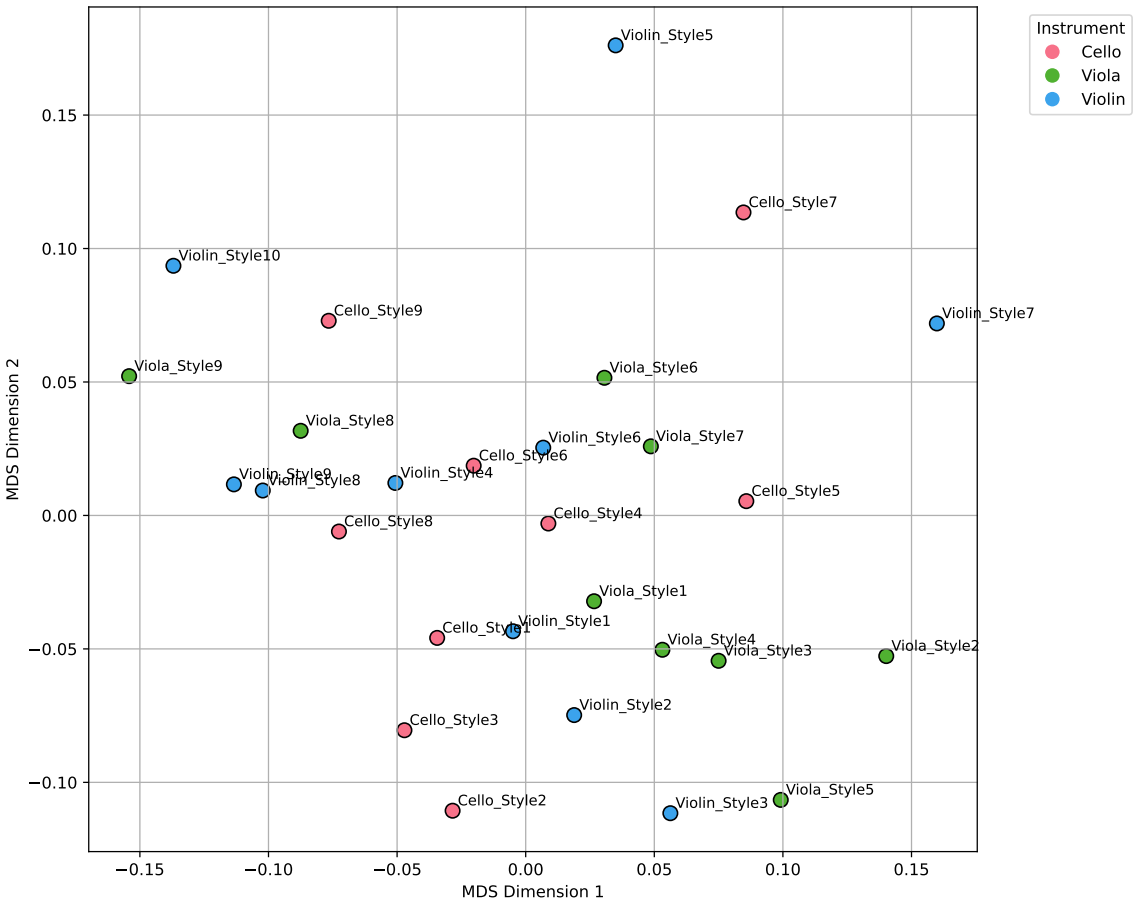


Figure B.2: MDS projection of instrument distances for violin, viola and cello with all styles.

Appendix C

Single-Objective Experiment Full Results



Figure C.1: Success matrix for all 100 fitness functions and all 100 chords.

Appendix D

Audio Feature Correlation

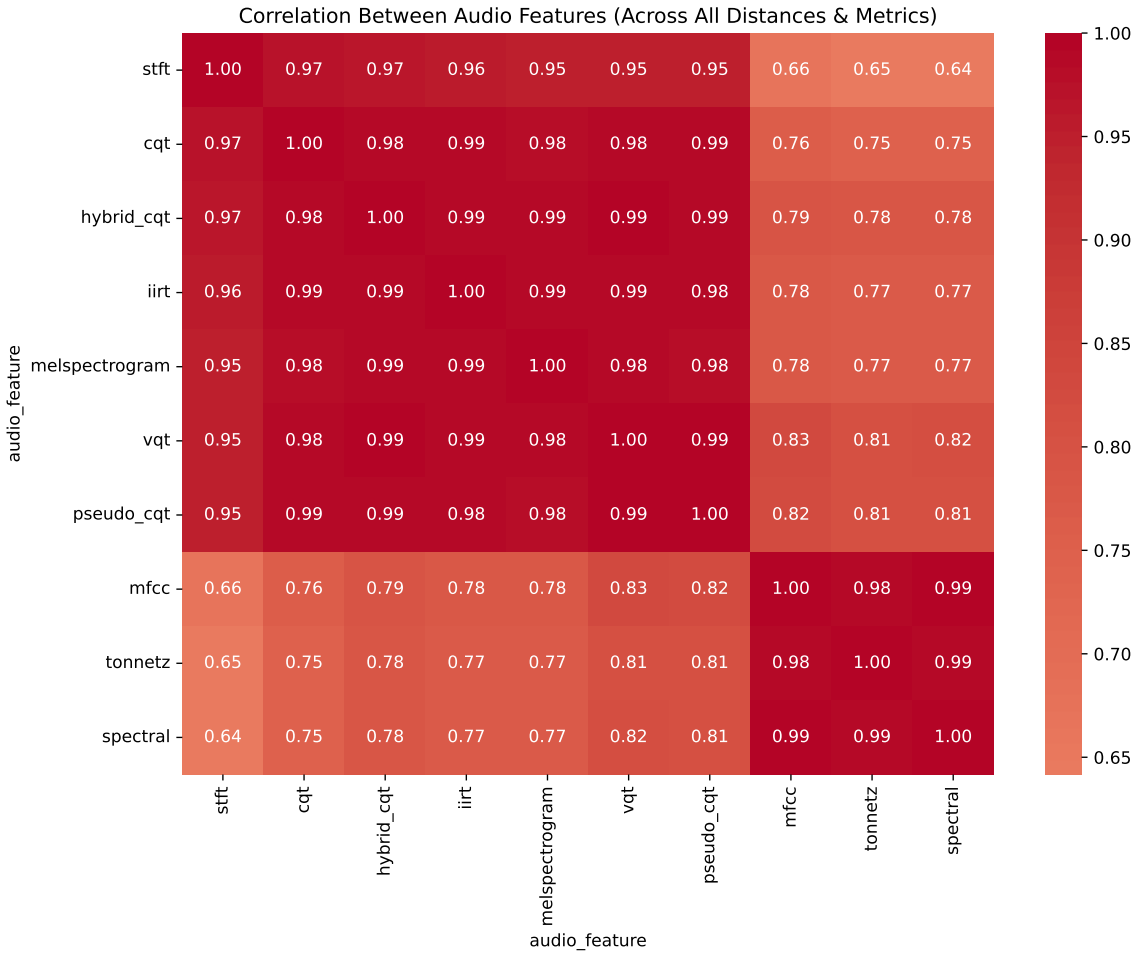


Figure D.1: Correlation matrix of audio features.

Appendix E

Statistics about the Tiny AAM Dataset

Piece	# Onsets	# Unique Onsets	Max Samples per Onset
0001	368	120	4
0080	498	136	8
0192	392	62	6
0620	806	167	9
0758	581	132	8
0989	583	146	8
1014	667	155	8
1050	569	161	7
1545	504	245	7
1711	464	126	8
1941	601	233	8
2269	468	299	8
2395	638	174	8
2462	369	122	7
2602	684	227	7
2720	640	285	9
2828	381	153	8
2841	935	235	9
2990	739	308	8
3000	369	133	8

Table E.1: Overview of dataset statistics.

Appendix F

Approximation Results by Piece

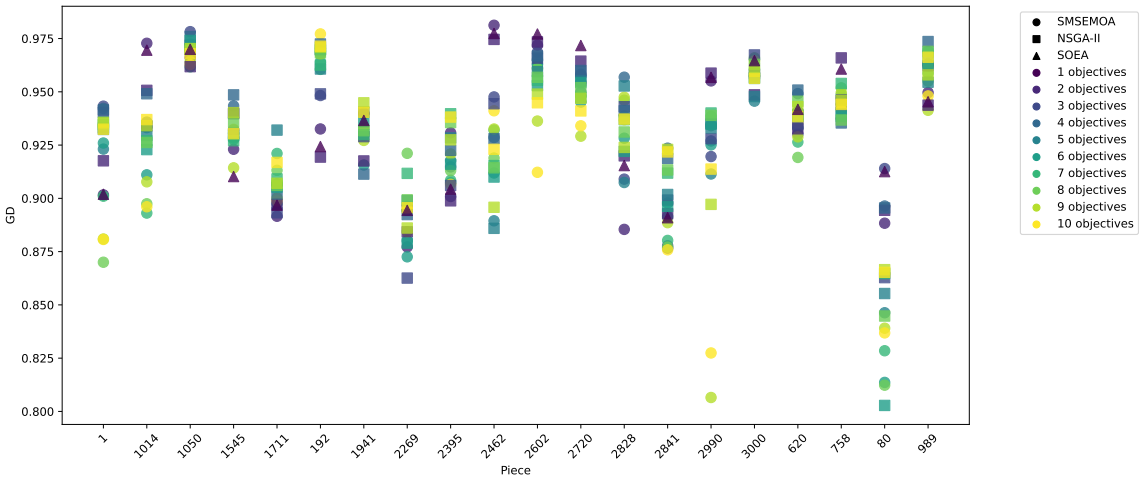


Figure F.1: Average GD by piece and algorithm.

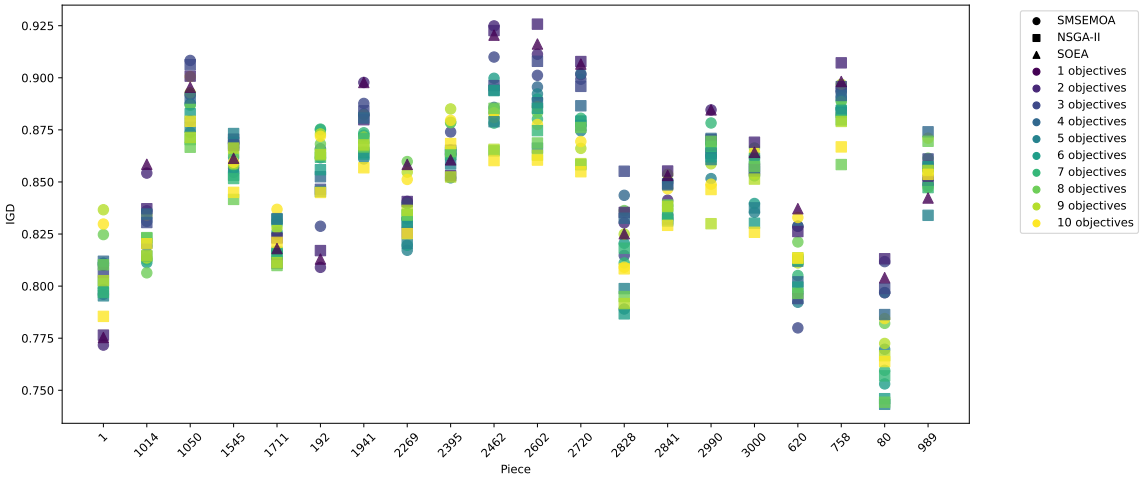


Figure F.2: Average IGD by piece and algorithm.