Imitative Computer-Aided Musical Orchestration with Biologically Inspired Algorithms

Marcelo Caetano, Carmine Cella

Imitative Computer-Aided Musical Orchestration with Biologically Inspired Algorithms

Marcelo Caetano and Carmine E. Cella

1 Introduction

Musical orchestration is an empirical art form based on tradition and heritage whose lack of formalism hinders the development of assistive computational tools. Computer-aided musical orchestration (CAMO) systems aim to assist the composer in several steps of the orchestration procedure. Particularly, *imitative* CAMO focuses on instrumentation by aiding the composer in creating timbral mixtures as instrument combinations. Imitative CAMO allows composers to specify a reference sound and replicate it with a predetermined orchestra [51]. Therefore, the aim of imitative CAMO is to find a combination of musical instrument sounds that perceptually approximates a reference sound when played together. However, the complexity of timbre perception and the combinatorial explosion of all possible musical instrument sound combinations make imitative CAMO a very challenging problem.

This chapter covers the theoretical background, the basic concepts, and algorithms involved in imitative CAMO. Specifically, this chapter describes the computational formalization of imitative CAMO and the motivation to use algorithms inspired by biological systems to tackle the complexity of timbral mixtures and the subjective nature of music composition. First, we present a brief review of timbre perception to motivate the use of the computer in musical orchestration. Then, we review several approaches to CAMO found in the literature. Next, we review CAMO systems that rely on the biologically inspired algorithms designated genetic algorithms (GA) and artificial immune systems (AIS), which are used to search for orchestrations via single-objective optimization (SOO) or multi-objective optimization (MOO). We discuss several aspects related to the different biologically inspired algorithms and optimization strategies focusing on the compositional perspective of orchestration. Finally, we conclude with future perspectives of CAMO.

1.1 Musical Orchestration

Traditionally, orchestration manuals regard musical orchestration as the process of writing music for the orchestra [65]. Orchestration has always been one of the most difficult disciplines to explain and convey [51]. The gap between the symbols in the score and their acoustic realization involves many steps that are difficult to quantify and some of these steps are oftentimes unpredictable. More than any other component of music composition, orchestration is an empirical activity essentially based on tradition and heritage. Even contemporary manuals of orchestration approach orchestration as an art form rather than a systematic procedure that can be captured by an algorithm. The lack of formalism in orchestration practice has been a major hindrance to the development of assistive computational tools.

Broadly speaking, orchestration is understood as "the art of blending instrument timbres together" [63]. Initially, orchestration was simply the assignment of instruments to pre-composed parts of the score, which was dictated largely by the availability of resources, such as what instruments and how many of each are available in the orchestra [47, 42]. Later on, composers started regarding orchestration as an integral part of the compositional process whereby the musical ideas themselves are expressed [47, 69]. Compositional experimentation in orchestration arises from the increasing tendency to specify instrument combinations to achieve desired effects, resulting in the contemporary use of timbral mixtures [55, 69]. Orchestration remains an empirical activity largely due to the difficulty to formalize the required knowledge [47, 63, 51].

In the past twenty years or so, composers have felt the need for a more systematic approach to orchestration to gain more control over timbral mixtures. Research in music writing pushed composers very far in imagining possible timbres resulting from extended instrumental techniques. Timbral mixtures have become more and more complex and predicting their *sound quality* while writing the score requires a great deal of experience and experimentation. In such a context, a tool to help simulate the result of timbral mixtures became a necessity. While other parameters of musical writing such as harmony and rhythm have been supported by computer-assisted techniques since the beginning of computer music [8], only recently did orchestration benefit from such tools because of its high complexity, requiring knowledge and understanding of both mathematical formalization and musical writing.

The concept of timbre lies at the core of musical orchestration [65, 47, 63, 51, 6, 7] because music and, consequently, musical instruments are strongly associated with timbre [65, 53, 73]. Musical orchestration uses the principle of instrumental combinations to obtain a desired effect. The orchestrator must have thorough knowledge of the individual instruments allied with a mental conception of their timbres. Additionally, the effects resulting from different instrumental combinations must be learned, such as balance of tone, mixed tone colors, and clarity in texture [65]. In this chapter, we will consider the specific example of *imitative orchestration*, where the aim is to find a combination of musical instrument sounds that, when played together, blends into a new timbre that perceptually approximates a given reference timbre. Imitative orchestration requires a great deal of knowledge about timbre, from

the timbre of isolated musical instruments to timbral mixtures. Unfortunately, timbre is a complex perceptual phenomenon that is not well understood enough to this day. In fact, nowadays timbre is considered the last frontier of auditory science [71]. Therefore, this chapter will provide a brief overview of timbre research to illustrate the complexity of (imitative) musical orchestration.

1.2 Musical Timbre

Historically, timbre was viewed as the perceptual quality of sounds that allows listeners to tell the difference between different musical instruments and ultimately recognize the instrument (or, more generally, the sound source). However, the term *timbre* can be misleading [55] because it has different meanings when it is used in psycho-acoustics, in music, in audio processing, and in other disciplines. Siedenburg *et al.* [73] recently wrote that, "Roughly defined, timbre is thought of as any property other than pitch, duration, and loudness that allows two sounds to be distinguished." Indeed, the complexity that the term timbre encompasses is mainly because [55] "[timbre] covers many parameters of perception that are not accounted for by pitch, loudness, spatial position, duration, and various environmental characteristics such as room reverberation." Similarly to *pitch* and *loudness*, timbre is a *perceptual* attribute [72], so timbre research commonly attempts to characterize quantitatively the ways in which sounds are perceived to differ [55].

1.2.1 The Helmholtz Theory of Timbre

In the nineteenth century, Hermann von Helmholtz published his seminal work in hearing science and musical acoustics [43] in which he used Fourier analysis to study musical instrument sounds. Helmholtz concluded that Fourier's theorem closely described both the acoustics of sound production and the physiological underpinnings of sound perception [73]. Regarding timbre, Helmholtz stated that [43] "the quality of the musical portion of a compound tone depends solely on the number and relative strength of its partial simple tones, and in no respect on their difference of phase." Thus, Helmholtz posited that the spectral shape is the acoustic feature that captures the timbre of the sound. However, his conclusions apply mainly to the steady state portion of musical instrument sounds because he assumed that the "musical tones" are completely stationary, neglecting the attack and decay portions of musical instrument sounds, as well as any temporal variations occurring during the course of the sound such as those found in the *glissando*, *sforzando*, and *vibrato* playing techniques. Later studies [67, 52] revealed the importance of temporal variations such as the attack time and spectral fluctuations in the recognition of these musical instruments. The sound quality captured by the spectral shape alone became known as *sound color* [74].

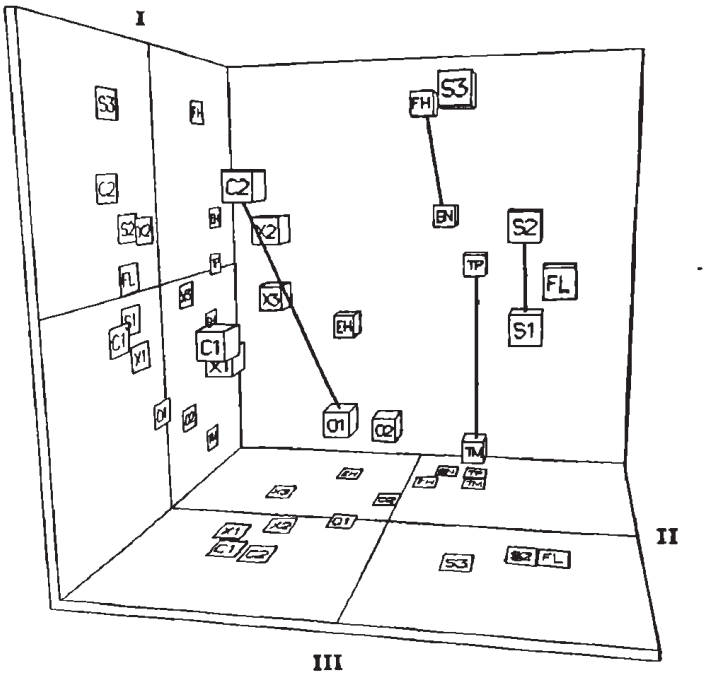


Fig. 1: Grey's [40] MDS timbre space. Each point represents a musical instrument sound, such that similar timbres are close together and dissimilar timbres are farther apart. Reprinted with permission from [40]. Copyright 1977, Acoustic Society of America.

1.2.2 Timbre Spaces

Some of the most successful attempts to study timbre perception quantitatively have resulted from multidimensional scaling (MDS) of dissimilarity ratings between pairs of musical instrument sounds [40, 56]. MDS generates a spatial configuration with points representing the musical instruments where the distances between the points reflect the dissimilarity ratings. This representation, called a *timbre space* (see Figure 1), places similar timbres closer together and dissimilar timbres farther apart. The musical instrument sounds used in MDS studies are equalized in pitch, loudness, and duration to ensure that the listeners focus on differences due to other perceptual attributes. Similarly, the sounds are presented over loudspeakers or headphones to remove differences due to spatial position. MDS timbre spaces [40, 50, 56, 11, 55] assume that the dimensions of timbre perception arising from the model are continuous and common to all the sounds presented. Additionally, there is the underlying assumption that all the listeners use the same perceptual dimensions to compare the timbres [55].

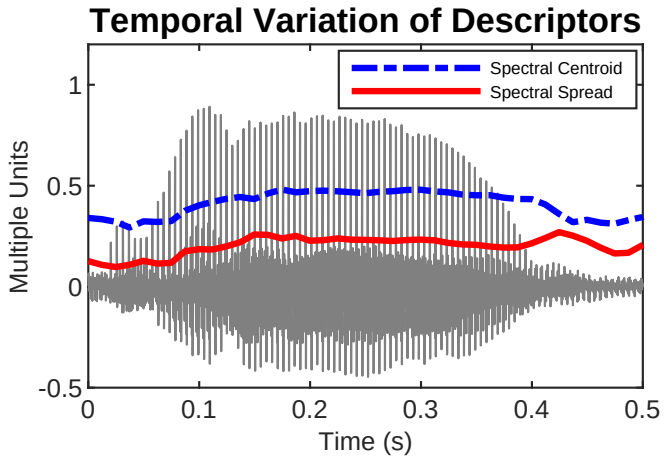


Fig. 2: Temporal variation of descriptors of timbre. The figure shows the temporal variation of the spectral centroid and of the spectral spread on top of the waveform of a trumpet note from which the descriptors were extracted.

1.2.3 Acoustic Correlates of Timbre Spaces

In MDS timbre studies, listeners typically use more than one dimension to rate the dissimilarity between pairs of sounds. This means that the sounds cannot be arranged along a single scale that reflects their pairwise dissimilarity (contrary to pitch, for example, where the sounds can be ordered from low to high). The resulting MDS timbre space commonly has two or three dimensions. Ultimately, the goal of MDS timbre studies is to unveil the psychological dimensions of timbre perception and associate them with the dimensions of the timbre space. Consequently, MDS timbre studies usually propose explanations for the dimensions of the timbre space found. Grey [40] qualitatively interpreted the three dimensions of his timbre space (see Figure 1) as (I) the distribution of spectral energy, (II) attack synchronicity of the partials, and (III) spectral balance during the attack. Later, researchers started to calculate acoustic descriptors from the sounds used in the MDS study and correlate these with the dimensions of the timbre space found [41, 49, 56], giving rise to *acoustic correlates of timbre spaces* [56, 55] also known as *descriptors of timbre* [64, 14]. From the plethora of descriptors proposed [64], the most ubiquitous correlates derived from musical instrument sounds include *spectral centroid*, the *logarithm of the attack time*, *spectral flux*, and *spectral irregularity* [55]. Nowadays, these descriptors of timbre are widely used in many computational tasks involving timbre [14], notably CAMO. Figure 2 illustrates the temporal variation of two descriptors of timbre for a relatively stable trumpet note. See [14] for details on the extraction of descriptors of timbre from audio and [64] for details on audio content descriptors in general. The role descriptors of timbre play in contemporary CAMO

systems will be explored in more detail throughout this chapter. But first, Sec. 1.2.4 summarizes conceptually the contemporary view of timbre.

1.2.4 The Contemporary View of Timbre

Today, we understand timbre from two distinct viewpoints, namely a sensory quality and a categorical contributor to sound source identification. Timbre as a multidimensional sensory quality is associated with timbre spaces, illustrated in Figure 1, whose dimensions can be either continuous (e.g., brightness) or categorical (e.g., the pinched offset of the harpsichord). From this viewpoint, two sounds can be declared qualitatively dissimilar independently from any association with their sources. In turn, timbre is also the primary perceptual vehicle for the recognition and tracking over time of the identity of a sound source, and thus involves the absolute categorization of a sound (into musical instruments, for example). This viewpoint sees timbre as a collection of auditory sensory descriptors that contributes to the inference of sound sources and events [72]. Further adding to its complex nature, timbre functions on different scales of detail [72] such that timbral differences do not always correspond to differences in sound sources [9] and timbres from sound-producing objects of the same type but different make may differ substantially enough to affect quality judgments [70]. The complexity of timbre perception plays a major role in the difficulty to formalize musical orchestration and also motivates the use of CAMO systems.

1.3 Musical Orchestration with the Aid of the Computer

The development of computational tools that aid the composer in exploring the virtually infinite possibilities resulting from the combinations of musical instruments gave rise to CAMO [66, 44, 20, 21, 69, 18, 22]. Imitative CAMO tools typically automate the search for instrument combinations that perceptually approximate a reference timbre commonly represented by a reference sound [51]. The combinations found can be subsequently included in the score and later played by orchestras in live performances [63]. However, most CAMO tools allow the composer to preview the result of the combinations found using musical instrument sounds from prerecorded databases, which has been deemed an appropriate rendition of the timbre of the instrument combinations [48].

Descriptors of timbre play a key role in several steps of recent CAMO systems [15, 57, 27], namely timbre description of isolated sounds, timbre description of combinations of musical instrument sounds, and timbre similarity between instrument combinations and the reference sound. The timbre of both the reference sound and of the isolated musical instrument sounds is represented with a descriptor vector comprising a subset of the traditional descriptors of timbre [14]. Each instrument combination corresponds to a vector of descriptors that captures the timbral result

of playing the instruments together [47]. So, the descriptor vector of an instrument combination is estimated from the descriptor vectors of the isolated sounds used in the combination [22, 36]. Timbre similarity between the reference sound and the instrument combination can be estimated as distances in timbre spaces [55], which are calculated as weighed distances between the corresponding descriptor vectors [18]. Smaller distances indicate a higher degree of timbral similarity [18] with the reference, so the instrument combinations with the smallest distances are returned as proposed orchestrations for a given reference sound.

The resulting instrument combinations found to orchestrate a given reference sound will depend on which descriptors are included in the descriptor vector. For example, spectral shape descriptors focus on approximating the distribution of spectral energy of the reference sound. However, early CAMO systems did not use descriptors of timbre at all, commonly resorting to the use of spectral information. In Sec. 2, we will delve deeper into the historical development of CAMO focusing mainly on the conceptual approach adopted to solve the problem of musical orchestration.

2 State of the Art

This section presents the state of the art of CAMO grouped into "early approaches", "generative approaches", and "machine learning". Section 2.1 presents the first CAMO systems proposed in the literature that commonly used subtractive spectral matching to find orchestrations. Next, Sec. 2.2 focuses on CAMO systems that search for orchestrations with the aid of biologically inspired algorithms. Finally, Sec. 2.3 covers CAMO systems based on machine learning.

2.1 Early Approaches

Early CAMO systems adopted a top-down approach [66, 44, 69] that consists of spectral analysis and subtractive spectral matching. These works commonly keep a database of spectral peaks from musical instruments that will be used to match the reference spectrum. The algorithm iteratively subtracts the spectral peaks of the best match from the reference spectrum aiming to minimize the residual spectral energy in the least squares sense. The iterative procedure requires little computational power, but the greedy algorithm restricts the exploration of the solution space, often resulting in suboptimal solutions because it only fits the best match per iteration [19].

Psenicka [66] describes SPORCH (SPectral ORCHestration) as "a program designed to analyze a recorded sound and output a list of instruments, pitches, and dynamic levels that, when played together, create a sonority whose timbre and quality approximate that of the analyzed sound." SPORCH keeps a database of spectral peaks of musical instrument sounds and uses subtractive spectral matching and least squares to return one orchestration per run. Hummel [44] approximates the spectral

envelope of phonemes as a combination of the spectral envelopes of musical instrument sounds. The method also uses a greedy iterative spectral subtraction procedure. The spectral peaks are not considered when computing the similarity between reference and candidate sounds, disregarding pitch among other perceptual qualities. Rose and Hetrik [69] use singular value decomposition (SVD) to perform spectral decomposition and spectral matching. SVD decomposes the reference spectrum as a weighted sum of the instruments present in the database, where the weights reflect the match. Besides the drawbacks from the previous approaches, SVD can be computationally intensive even for relatively small databases. Additionally, SVD sometimes returns combinations that are unplayable such as multiple simultaneous notes on the same violin, requiring an additional procedure to specify constraints on the database that reflect the physical constraints of musical instruments and of the orchestra.

2.2 Generative Approaches

The top-down approach neglects the exploration of timbral mixtures by relying on spectral matching, which does not capture the multi-dimensional nature of timbre. Carpentier *et al.* [20, 21, 76, 18, 22] adopted a bottom-up approach that relies on timbre similarity and evolutionary computation to search for instrument combinations that approximate the reference. The bottom-up approach represents a paradigm shift toward *generative CAMO* [21, 18, 33, 1, 15], where the timbre of instrument combinations is compared with the timbre of the reference sound via descriptors of timbre. Currently, there are two generative CAMO frameworks, the Orch* family of CAMO systems based on GA [18, 22, 33, 27], and CAMO-AIS [1, 15], which uses an artificial immune system (AIS). Orch* comprises three CAMO systems, namely *Orchidée* [20, 21, 76, 18, 22], Orchids [33, 32], and Orchidea [27]. Both Orch* and CAMO-AIS rely on algorithms inspired by biological systems that use a population of individuals to search for a solution in the vast pool of possible instrument combinations following *Orchidée*, the first generative CAMO system to be proposed.

2.2.1 Orch*

Orchidée searches for combinations of musical instrument sounds as a constrained combinatorial optimization problem. Carpentier *et al.* [20, 21, 76, 18, 22] formulate CAMO as a binary allocation knapsack problem where the aim is to find a combination of musical instruments that maximizes the timbral similarity with the reference constrained by the capacity of the orchestra (i.e., the database). *Orchidée* explores the vast space of possible instrument combinations with a GA that optimizes a fitness function which encodes timbral similarity between the candidate instrument combinations and the reference sound. Specifically, *Orchidée* uses the well-known multi-objective genetic local search (MOGLS) optimization algorithm [45] to return

multiple instrument combinations in parallel that are nearly Pareto optimal. Section 4.2 explains multi-objective optimization (MOO) in more detail, whereas Sec. 4 explores the use of biologically inspired algorithms.

Orchids was born out of a compositional drawback of *Orchidée*, namely *static orchestrations*. The problem is that static orchestrations do not take into account temporal variations in the reference sound. Static orchestrations can be understood with the aid of Figure 2, which shows the temporal variation of two descriptors of timbre calculated at equal steps. *Orchidée* uses descriptor vectors with the average value of the descriptors across time. A timbre-similarity measure based on temporal averages is appropriate when orchestrating reference sounds that do not present much temporal variation, such as stable musical notes sung or played on musical instruments [18, 15]. However, reference sounds such as an elephant trumpeting require taking the temporal variation of descriptors into consideration. Esling *et al.* [33, 32] developed Orchids with the ability to perform dynamic orchestrations by representing the temporal variation of descriptors of timbre. Orchids uses a multi-objective time series matching algorithm [31] capable of coping with the temporal and multidimensional nature of timbre. Orchids also uses MOO to return a set of efficient solutions rather than a single best solution.

Orchidea [27], the third generation of the Orch* family, expands *Orchidée* and Orchids toward macro-scale dynamic orchestration. Orchidea was conceived to be a full-fledged framework that helps composers in all the steps of the compositional process. Most of its design focuses on usability and on the integrability of the proposed solutions into a compositional workflow. In particular, Orchidea handles the temporal dimension of the reference sound differently from Orchids. While Orchids focuses on the micro-temporal scale of low-level descriptors, Orchidea shifts attention to the macro-scale of musical onsets, providing a more accessible approach for the users. Section 5.4 provides further information about dynamic orchestrations with Orchidea.

2.2.2 CAMO-AIS

CAMO-AIS addresses a different drawback of the Orch* family, namely *diversity of orchestrations*. Diversity has been identified as an important property that can provide the composer with multiple alternatives given the highly subjective nature of musical orchestration combined with the complexity of timbre perception [19]. Theoretically, the use of MOO allows to find many orchestrations (see Sec. 4.2 for more details). However, in practice, the orchestrations returned by *Orchidée*, for example, were all very similar to one another, commonly differing by only one musical instrument sound [15]. Caetano *et al.* [1, 15] proposed to use an AIS called opt-aiNet to search for combinations of musical instrument sounds that minimize the distance to a reference sound encoded in a fitness function. CAMO-AIS relies on single-objective optimization (SOO) and the multi-modal ability of opt-aiNet to find multiple solutions in parallel. Opt-aiNet was developed to maximize diversity in the solution set, which results in alternative orchestrations for the same reference sound

that are different among themselves. The companion webpage for CAMO-AIS [12] has several sound examples that compare orchestrations returned by *Orchidée* and CAMO-AIS for their diversity and perceptual similarity with the reference.

2.3 Machine Learning

Recently, Antoine *et al.* [4, 6, 5, 57] proposed the interactive CAMO system i-Berlioz to address what is considered to be a hindrance to the compositional workflow of *Orchidée* and Orchids, namely the multiple orchestrations returned by these CAMO systems [51]. They argue that the process of listening to multiple orchestrations to select one can be tedious, ineffective, and time-consuming, especially when the user has a particular sound quality in mind [57]. Instead, they propose to *narrow down* the orchestrations returned by i-Berlioz with constraints, making i-Berlioz conceptually opposed to the principle of maximum diversity in CAMO-AIS. i-Berlioz [57] suggests combinations of musical instruments to produce timbres specified by the user by means of verbal descriptors. Currently, five semantic descriptors of timbre are supported, namely "breathiness", "brightness", "dullness", "roughness", and "warmth". A support vector machine classifier is trained to match instrument combinations to the semantic descriptions. Additionally, i-Berlioz is also capable of performing dynamic orchestrations [57].

3 Imitative Computer-Aided Musical Orchestration

The purpose of this section is to lay the groundwork for a formalization of imitative CAMO focusing on generative systems that use biologically inspired algorithms to search for orchestrations. The end of this section points out the technical difficulties involved in finding orchestrations that perceptually approximate a given reference sound. Then, Sec. 4 presents the solutions adopted to circumvent the difficulties in this formalization of imitative CAMO from a conceptual standpoint.

There are several bio-inspired generative CAMO algorithms (*Orchidée*, Orchids, Orchidea, and CAMO-AIS), each of which frames CAMO differently. Therefore, it would be impractical and rather confusing to try to exhaustively describe all of them. Instead, this section will focus on CAMO-AIS [1, 15], which closely follows the framework proposed by Carpentier *et al.* [20, 21, 17, 18, 22, 19]. Section 4 will explore the main differences between CAMO-AIS and *Orchidée*, especially the differences between the optimization method adopted by each and the consequence in terms of diversity of orchestrations. It is out of the scope of this chapter to provide a detailed explanation of either the bio-inspired algorithms (GA and AIS) or the optimization methods (SOO and MOO). See the references in the respective sections for further details.

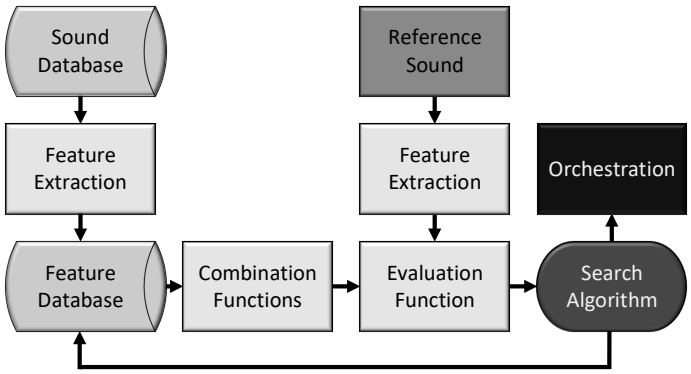


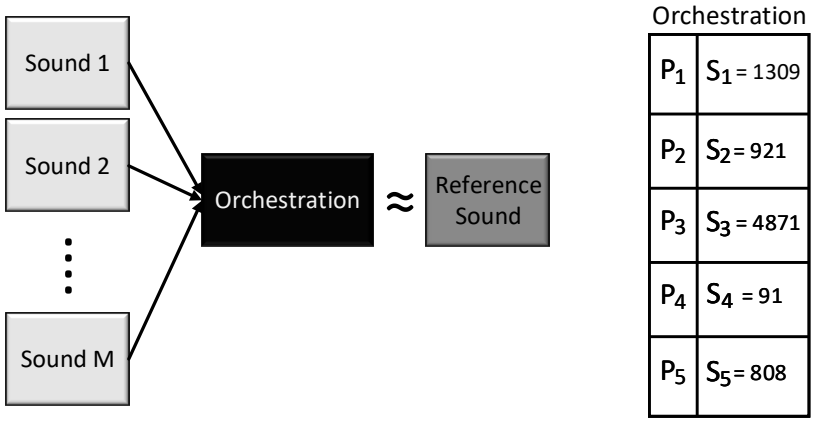
Fig. 3: Overview of CAMO-AIS. The figure illustrates the different components of the framework. Reprinted from [15] with permission from Elsevier.

3.1 Overview

Figure 3 shows an overview of CAMO-AIS. The sound database is used to build a feature database, which consists of descriptors of pitch, loudness, and timbre calculated for all sounds prior to the search for orchestrations. The same descriptors are calculated for the reference sound being orchestrated. The combination functions estimate the descriptors of a sound combination from the descriptors of the individual sounds. The evaluation function uses these descriptors to estimate the similarity between combinations of descriptors from sounds in the database and those of the reference sound. The search algorithm opt-aiNet is used to search for combinations that approximate the reference sound, called orchestrations.

3.2 Representation

Figure 4a illustrates an orchestration as a combination of sounds from the sound database that approximates the reference sound when played together. Figure 4b shows the representation used by CAMO-AIS, in which an orchestration has M players $p(m)$, and each player is allocated a sound $s(n) \in S$, where $n = [1, \ldots, N]$ is the index in the database S, which has N sounds in total. Thus an orchestration is a combination of sounds $c(m, n) = \{s_1(n), \ldots, s_M(n)\},\ \forall\ s_m(n) \in S$. Figure 4b shows $c(m, n)$ represented as a list, but the order of players $p(m)$ does not matter for the orchestration. Each sound $s_m(n)$ corresponds to a specific note of a given instrument played with a dynamic level, and $s_m(n) = 0$ indicates that player $p(m)$ was allocated no instrument.



(a) Orchestration as a sound combination. (b) Representation.

Fig. 4: Representation of orchestrations. Part (a) illustrates the orchestration as a combination of sounds that approximates the reference. Part (b) shows the internal representation of each orchestration in CAMO-AIS. Reprinted from [15] with permission from Elsevier.

3.3 Audio Descriptor Extraction

Timbre perception excludes pitch, loudness, and duration (see Sec. 1.2). Therefore, we consider pitch, loudness, and duration separately from timbre dimensions. The descriptors used are fundamental frequency f_0 (pitch), frequency f and amplitude a of the contribution spectral peaks A, loudness λ, spectral centroid μ, and spectral spread σ. The fundamental frequency f_0 of all sounds $s(n)$ in the database is estimated with Swipe [16]. The spectral centroid μ captures brightness while the spectral spread σ correlates with the third dimension of MDS timbre spaces [40, 50, 56, 11]. All the descriptors are calculated over short-term frames and averaged across all frames.

3.3.1 Contribution Spectral Peaks

The spectral energy that sound $s(m)$ contributes to an orchestration is determined by the *contribution-spectral-peak* vector $\mathbf{A}_m(k)$. In what follows, only peaks whose spectral energy (amplitude squared) is at most 35 dB below the maximum level (i.e., 0 dB) are used and all other peaks are discarded. These peaks are stored as a vector with the pairs $\{a(k), f(k)\}$ for each sound $s(m)$, where k is the index of the peak. The *contribution spectral peaks* $\mathbf{A}_m(k)$ are the spectral peaks from the *candidate* sound $s(m)$ that are common to the spectral peaks of the *reference* sound r. Eq. (1) shows the calculation of $\mathbf{A}_m(k)$ as

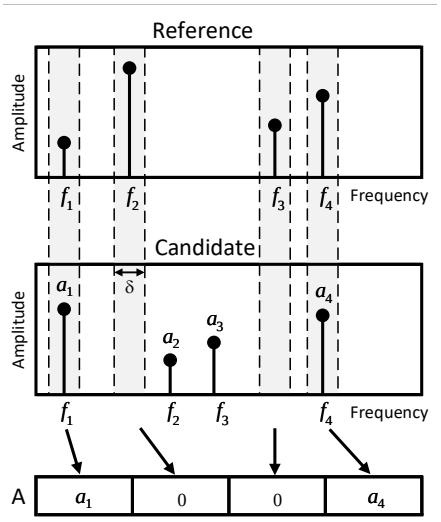


Fig. 5: Contribution spectral peaks $\mathbf{A}_m(k)$. The figure shows the representation of the contribution spectral peaks of a candidate sound. Reprinted from [15] with permission from Elsevier.

$$\mathbf{A}_m(k) = \begin{cases} a_s(k) & \text{if } (1+\delta)^{-1} \leq f_s(k)/f_r(k) \leq 1+\delta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $a_s(k)$ is the amplitude and $f_s(k)$ is the frequency of the spectral peak of the *candidate* sound, and $f_r(k)$ is the frequency of the *reference* sound.

Figure 5 illustrates the computation of spectral peak similarity between the *reference* sound and a *candidate* sound. Spectral peaks are represented as spikes with amplitude $a(k)$ at frequency $f(k)$. The frequencies $f_r(k)$ of the peaks of the *reference* sound are used as reference. Whenever the *candidate* sound contains a peak in a region δ around $f_r(k)$, the amplitude $a(k)$ of the peak at frequency $f_s(k)$ of the *candidate* sound is kept at position k of the contribution spectral peaks vector $\mathbf{A}_m(k)$.

3.3.2 Loudness

Loudness λ is calculated as

$$\lambda = 20\log_{10}\left(\sum_k a(k)\right), \tag{2}$$

where $a(k)$ are the amplitudes at frequencies $f(k)$.

3.3.3 Spectral Centroid

The spectral centroid μ is calculated as

$$\mu = \sum_k f(k) \frac{|a(k)|^2}{\sum_k |a(k)|^2}. \tag{3}$$

3.3.4 Spectral Spread

The spectral spread σ is calculated as

$$\sigma = \sum_k (f(k) - \mu)^2 \frac{|a(k)|^2}{\sum_k |a(k)|^2}. \tag{4}$$

3.4 Pre-Processing

Prior to the search for orchestrations of a given reference sound r, the entire sound database S is reduced to a subset S^r of sounds that will be effectively used to orchestrate r. All the sounds whose contribution spectral peaks vector $\mathbf{A}_m(k)$ is all zeros are eliminated because these do not contribute spectral energy to the orchestration. Similarly, all the sounds whose f_0 is lower than f_0^r are eliminated because these add spectral energy outside of the region of interest and have a negative impact on the final result. Partials with frequencies higher than all frequencies in r are not considered because these are in the high-frequency range and typically have negligible spectral energy.

3.5 Combination Functions

The sounds $s(n)$ in an orchestration $c(m, n)$ should approximate the reference r when played together. Therefore, the combination functions estimate the values of the spectral descriptors of $c(m, n)$ from the descriptors of the isolated sounds $s(n)$ normalized by the RMS energy $e(m)$ [22]. The combination functions for the spectral centroid μ, spectral spread σ, and loudness λ are given respectively by

$$\mu_c = \frac{\sum_m^M e(m)\,\mu(m)}{\sum_m^M e(m)}, \tag{5}$$

$$\sigma_c = \sqrt{\frac{\sum_{m}^{M} e(m)\left(\sigma^2(m) + \mu^2(m)\right)}{\sum_{m}^{M} e(m)} - \mu_c^2}, \tag{6}$$

$$\lambda_c = 20\log_{10}\left(\sum_{m}^{M}\frac{1}{K}\sum_{k}^{K} a(m,k)\right). \tag{7}$$

The estimation of the contribution spectral peaks of the combination $\mathbf{A}_c$ uses the contribution vectors $\mathbf{A}_s$ of the sounds $s(n)$ in $c(m,n)$ as

$$\mathbf{A}_r = \left\{\max_{k\in K}[\mathbf{A}(m,1)], \max_{k\in K}[\mathbf{A}(m,2)], \cdots, \max_{k\in K}[\mathbf{A}(m,N)]\right\}. \tag{8}$$

3.6 Distance Functions

Equation (13) shows the calculation of the fitness value F of an orchestration as the weighed sum of distances D_j. Each distance D_j in eq. (13) measures the difference between the descriptors from the reference sound r and the candidate orchestration $c_q(m,n)$, where q is the index of the orchestration among all the candidates for r, as follows

$$D_\mu = \frac{|\mu(c_q) - \mu(r)|}{\mu(r)}, \tag{9}$$

$$D_\sigma = \frac{|\sigma(c_q) - \sigma(r)|}{\sigma(r)}, \tag{10}$$

$$D_\lambda = \frac{|\lambda(c_q) - \lambda(r)|}{\lambda(r)}. \tag{11}$$

The distance between the contribution vector of the reference sound $\mathbf{A}_r$ and the contribution vector of the orchestration $\mathbf{A}_c$ is calculated as

$$D_{\mathbf{A}} = 1 - \cos(\mathbf{A}_r, \mathbf{A}_c). \tag{12}$$

3.7 Calculating the Fitness of Orchestrations

The fitness of an orchestration is an objective measure of the timbral distance between the orchestration and the reference. Since each descriptor used has an independent distance function D_j associated, the total fitness F is defined as the weighed combination of D_j expressed as

$$F\left(\alpha_j\right) = \sum_j \alpha_j D_j \quad \text{with} \quad \sum_j \alpha_j = 1 \quad \text{and} \quad 0 \leq \alpha_j \leq 1. \tag{13}$$

where j is the index of each feature, α_j are the weights, and D_j are the distance functions. The fitness value $F\left(\alpha_j\right)$ of a candidate orchestration calculated with eq. (13) depends on the values of the weights α_j. Choosing numerical values for α_j subject to $\sum_j \alpha_j = 1,\ 0 \leq \alpha_j \leq 1$ allows to compare numerically the fitness F of different orchestrations. Optimization of F following the numerical choice of α_j is known as SOO, which constrains the solutions found to that particular combination of weights. CAMO-AIS [1, 15] uses SOO to minimize F, whereas *Orchidée* [22] uses MOO. Sec. 4 will discuss the difference between SOO and MOO from the perspetive of CAMO.

4 Computer-Aided Musical Orchestration with Bio-Inspired Algorithms

The goal of imitative CAMO is to find a combination c of M musical instrument sounds s from a database S that perceptually approximates a given reference sound r. Section 3 formalized imitative CAMO as a function optimization problem, where the goal is to minimize the fitness F in eq.(13). However, minimization of F is not a trivial task because it is an *inverse problem* and because F is a combination of *multiple objectives*. The formulation of CAMO as an inverse problem requires searching for orchestrations, so Sec. 4.1 discusses the need for bio-inspired algorithms to perform the search. Finding an orchestration requires minimizing the multiple distances encoded in the fitness function, and Sec. 4.2 discusses the use of SOO and MOO to do it.

4.1 Searching for Orchestrations for a Reference Sound

Calculating the fitness F of an orchestration with eq. (13) requires multiple steps shown in Figure 3. Mathematically, the fitness function $\mathcal{F}$ measures the distance between a reference sound r and a combination $c\,(m,n)$ of M sounds from the database S as $F = \mathcal{F}\,(c,r)$. Thus, minimizing $\mathcal{F}$ can be expressed as

$$\min_{c(m,n)} \mathcal{F}\,(c,r), \quad c\,(m,n) = \{s_1\,(n), \ldots, s_M\,(n)\} \in S^r \subseteq \mathbb{N}^M, \tag{14}$$

which is read as "find the combination $c\,(m,n)$ of M sounds $s\,(n)$ from the database S^r that minimizes the distance F to r". This mathematical formulation of CAMO is known as an *inverse problem* in the optimization literature because $\mathcal{F}$ only allows to calculate the distance F given the combination c and the reference r. There is no inverse $\mathcal{F}^{-1}$ to retrieve which c corresponds to a specific F. So we cannot simply

set a desired value for F and retrieve the orchestration(s) that correspond to it. In practice, we must search for the combination c that results in the minimum distance F.

At first, it might seem trivial to search for the orchestration that minimizes $\mathcal{F}$. For example, exhaustive search will simply try all possible combinations and return the one with minimum distance F. However, the combinatorial nature of CAMO means that this *brute-force* approach will suffer from the growth in complexity as the size of the database S increases known as *combinatorial explosion*. Depending on the size of the database, exhaustive search can take from a few minutes to longer than the age of the universe! In computational complexity, combinatorial optimization problems are said to be in NP. It is easy to check if a candidate is indeed an answer to a problem in NP, but it is really difficult to find any answer [38]. See the Clay Mathematics Institute webpage about the P vs NP problem [28] for further information. Carpentier *et al.* [20, 21, 76, 18, 22] formalized imitative CAMO as a binary allocation knapsack problem, which was proved to be NP-complete [46]. Thus, *heuristic* search strategies are typically used to find approximate solutions to imitative CAMO. Biologically inspired algorithms such as GA and AIS are popular choices because they use clever search heuristics to check promising instrument combinations.

4.1.1 Genetic Algorithms

GA use an abstraction of biological evolution to provide computer systems with the mechanisms of adaptation and learning [30]. Evolution can also be seen as a method for designing innovative solutions to complex problems. Thus the GA evolves a population of candidate solutions represented as *chromosomes* by means of the genetic operators of mutation, crossover, and selection [59, 37]. A fitness function evaluates the quality of each individual of the population. The fittest individuals are selected to generate offspring by exchanging genetic material (crossover). Then the offspring undergo mutation and only the fittest offspring are selected for the next generation.

The *search space* comprises the collection of all potential solutions resulting from the representation adopted. A measure of "distance" between candidate solutions allows to define the neighborhood of regions in the search space as well as the *fitness landscape*, which is a representation of the fitness of all the individuals in the search space. A smooth fitness landscape is akin to a continuous function where "neighboring" candidate solutions have similar fitness value. Combinatorial optimization problems typically do not feature continuous fitness landscapes, adding to their difficulty. The mutation operator is responsible for exploitation of the search space by introducing small random perturbations that search the neighborhood of promising regions. The crossover operator performs exploration of the search space under the assumption that high-quality "parents" from different regions in the search space will produce high-quality "offspring" candidate solutions. Finally, the selection operator is responsible for implementing the principle of *survival of the fittest* by

only allowing the fittest individuals to generate offspring and be passed on to the next generation. So, GA work by discovering, emphasizing, and recombining good *building blocks* of solutions in a highly parallel fashion. Adaptation in a GA results from the trade-off between the exploration of new regions of the search space and the exploitation of the current promising regions (e.g., local optima). In fact, the parallel nature of the search can be interpreted as the GA allocating resources (i.e., candidate solutions) to regions of the search space based on an estimate of the relative performance of competing regions.

GA have become popular to solve hard combinatorial optimization problems, such as imitative CAMO. In fact, GA are particularly suited to find solutions in complex domains, such as music [10, 60, 58] and the arts [78, 68]. See also the online proceedings of the EvoMUSART conference [75] and the EvoStar web page [34]. In CAMO, the timbre arising from instrument combinations is unknown *a priori* and the orchestrations proposed by the GA might contain surprising combinations of musical instruments not contained in traditional orchestrations manuals. However, GA also present several drawbacks, such as slow convergence and loss of diversity [59]. The next section will introduce AIS and focus on how the characteristic of *maintenance of diversity* can be used in CAMO.

4.1.2 Artificial Immune Systems

AIS are inspired by theoretical immunology and immune functions applied to solve real-world problems [25, 29]. The biological immune system features many properties that can be useful in several branches of science [25, 39], engineering [77, 29], and the arts [13, 61, 62], including robustness, pattern recognition, fault and noise tolerance, learning, self-organization, feature extraction, and diversity. Additionally, the immune system is self-organizing, highly distributed, adaptable to dynamic and complex environments, and it displays cognitive properties such as a decentralized control and memory [77], akin to neural networks. The (vertebrate) immune system is incredibly complex and not yet fully understood [39]. However, several mechanisms of the adaptive immune system have served as inspiration for AIS [25, 39], such as negative selection, the immune network theory, and clonal selection, among others. Thus, it can be said that AIS use abstractions of immunological processes to endue algorithms with some of its properties. Consequently, AIS is an umbrella term that encompasses several different algorithms [25, 77, 39].

CAMO-AIS uses opt-aiNet [24], an AIS for multi-modal optimization that draws inspiration from the immunological principles of clonal selection, affinity maturation, and the immune network theory [25, 77, 39]. Clonal selection commonly serves as inspiration for search and optimization, whereas the immune network theory is commonly associated with learning [39]. Clonal selection algorithms [23] present a strong resemblance to GA without crossover, but their notion of affinity and their significantly higher mutation rate (i.e., hypermutation) distinguish them from similar adaptive algorithms [25, 39]. In opt-aiNet, hypermutation contributes to diversity [25, 77] and affinity maturation adds learning and adaptation. Addition-

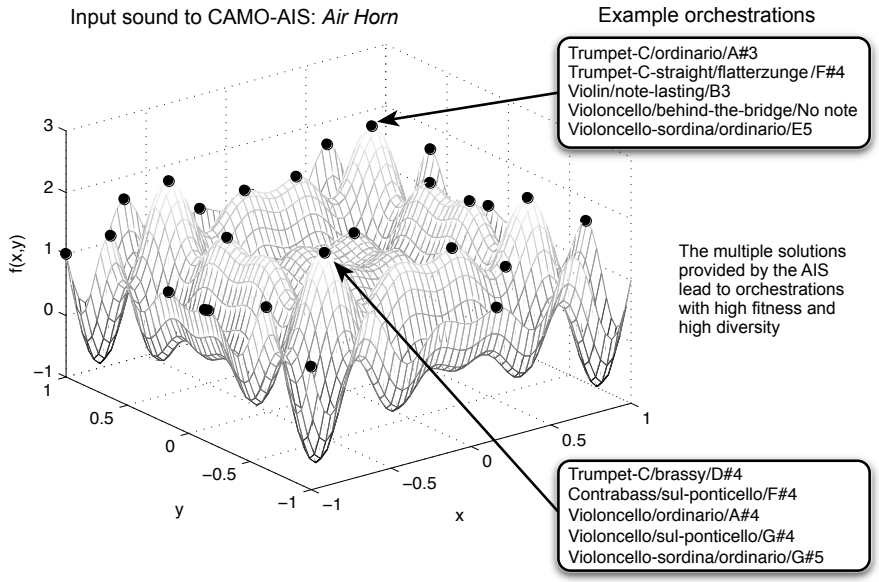


Fig. 6: Illustration of multi-modal function optimization in CAMO. The figure shows an objective function with multiple optima. The black dots represent multiple orchestrations returned by CAMO-AIS. Two example orchestrations for the reference sound *air horn* are given following the convention *instrument/playing technique/note*. Access the CAMO-AIS webpage [12] to listen to these orchestrations among several other examples. Reprinted from [15] with permission from Elsevier.

ally, the affinity measure is used in a suppression stage that is instrumental to the characteristic maintenance of diversity of opt-aiNet [24].

4.1.3 Maintenance of Diversity in opt-aiNet

Opt-aiNet was developed to solve multi-modal optimization problems [25, 77, 39], which exhibit *local optima* in addition to a *global optimum*. A local optimum is better than its neighbors but worse than the global optimum. Figure 6 shows a (continuous) multi-modal function with global and local optima represented by the peaks. Standard optimization methods such as GA commonly only return one solution (i.e., one black dot) corresponding to one local optimum of the fitness function. The property of maintenance of diversity allows opt-aiNet to find and return multiple local optima in parallel.

Local optima can be very interesting for CAMO given the subjective nature of orchestration. Composers are seldom interested in the "best" solution to eq. (14) in a mathematical sense. A set of multiple orchestrations to choose from is potentially more interesting from a compositional point of view. A CAMO algorithm that is

capable of proposing multiple orchestrations in parallel that resemble the reference sound differently can be valuable. However, finding local optima of a multi-modal fitness function with SOO is not the only method to propose multiple orchestrations for a reference sound. MOO also allows to find multiple orchestrations in parallel. Section 4.2 illustrates the difference between the SOO and the MOO approaches in CAMO. Then, Sec. 5 discusses the differences between these approaches, emphasizing the advantages and disadvantages of each.

4.2 Finding Orchestrations for a Reference Sound

Equation (13) defined the fitness $F\left(\alpha_j\right)$ of an orchestration as the weighed sum of the individual distances D_j calculated for each descriptor. It is important to note that the value of $F\left(\alpha_j\right)$ depends on the weights α_j. The SOO approach consists in choosing numerical values for α_j and finding one or more orchestrations corresponding to that particular combination of weights, whereas the MOO [79] approach returns multiple solutions corresponding to different values of the weights α_j. CAMO-AIS uses SOO and the multi-modal ability of opt-aiNet to find multiple local optima that maximize diversity in the feature space. *Orchidée* uses the well-known multi-objective genetic local search (MOGLS) optimization algorithm [45] to generate a pool of orchestrations by approximating the Pareto frontier [79].

Figure 7 shows the search space, the feature space, and the objective space to illustrate the difference between SOO and MOO conceptually. Each point in the search space is an orchestration represented as an instrument combination that has a corresponding position in the feature space. The middle panel in Figure 7 shows the reference sound (black dot) in the feature space among the orchestrations (grey dots) to illustrate the calculation of the distances D_j between the orchestrations and the reference sound. Finally, the objective space is obtained by associating a dimension to each distance D_j. The weights α_j map the distances D_j from the feature space to the objective space, where each point corresponds to a fitness value F of an orchestration.

The main difference between SOO and MOO lies in the objective space. In SOO, the weights α_j are fixed, so the objective space is one-dimensional (i.e., a line) and the fitness values F depend exclusively on the distances D_j. Therefore, minimizing F requires finding an orchestration whose distances D_j are as small as possible (i.e., as close as possible to the origin). In MOO, the weights α_j are not defined beforehand, so each point in the objective space (corresponding to a specific value F) depends on the values of both D_j and α_j. Each orchestration occupies a fixed point in the feature space, and so does the reference. Therefore, the distances D_j are also fixed for each orchestration. However, different weights α_j will map the same orchestration in the feature space to different points in the objective space, as illustrated in Figure 7. Consequently, in MOO, each orchestration corresponds to multiple points in the objective space with varying values of F. Thus, minimizing the fitness function $\mathcal{F}\left(\alpha_j\right)$ requires finding both the orchestration with distances D_j

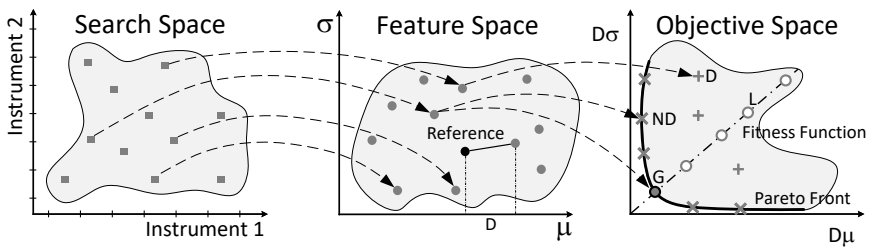


Fig. 7: Illustration of the different spaces in CAMO. The left-hand panel shows the search space, the middle panel shows the feature space, and the right-hand panel shows the objective space. Each point in the search space is an instrument combination (orchestration) that has a corresponding position in the feature space. The reference sound can also be seen in the feature space. The distances D between points in the feature space and the reference sound are calculated in the feature space. The weights α_j map points in the feature space to the objective space. Reprinted from [15] with permission from Elsevier.

and the combination of weights α_j for that specific orchestration that are as close to the origin as possible. So, there is more than one possible direction from which to minimize $\mathcal{F}\left(\alpha_j\right)$. In fact, there are multiple minima of $\mathcal{F}\left(\alpha_j\right)$ corresponding to different combinations of weights α_j. The set of all minima of $F\left(\alpha_j\right)$ is called *Pareto front*, illustrated in Figure 7 as the thick border in the objective space.

Figure 7 illustrates the objective space with non-dominated solutions (ND) represented as "X" and dominated solutions (D) represented as "+". Solutions along the Pareto front are called non-dominated (ND) because there is no other solution whose value of $\mathcal{F}\left(\alpha_j\right)$ is closer to the origin. The SOO fitness function in the objective space can also be seen as a straight line containing the global optimum (G) illustrated as the filled "O" and the local optima (L) illustrated as the empty "O". Note that dominated solutions D can coincide with local optima L and, in turn, non-dominated solutions ND can coincide with the global optimum G. Thus CAMO-AIS returns solutions L that were discarded by *Orchidée* because there is a solution G closer to the reference in the same direction in the objective space (i.e., specified by the same α_j). Section 5 will explore further the consequence of the different approaches by CAMO-AIS and *Orchidée*.

5 Discussion

This section discusses aspects of the Orch* family and of CAMO-AIS. Firstly, the differences between the SOO and MOO approaches are examined more closely. Then, the attention shifts to the difference between dynamic orchestrations with Orchids and with Orchidea.

5.1 Perceptual Considerations

Conceptually, two values have important perceptual and aesthetic consequences in CAMO, namely the fitness value F and the weights α_j in eq. (13). F is the objective measure of distance between the orchestrations and the reference. Therefore, minimizing F is conceptually similar to maximizing the timbral similarity, so F is inversely proportional to the perceptual similarity with the reference. In theory, a smaller F indicates a higher degree of similarity.

The weights α_j allow to emphasize the relative importance of each descriptor in the orchestrations returned. For example, a relatively high value of α_μ for the spectral centroid distance D_μ would penalize more severely orchestrations whose D_μ is higher. Consequently, the focus would be on matching *brightness* because it is the perceptual counterpart of the spectral centroid [56]. Therefore, α_j can be interpreted as specifying the *perceptual direction* from which an orchestration approaches the reference. In other words, the weights α_j control the perceptual dimension(s) of the similarity between the orchestration and the reference.

The main differences between the orchestrations by CAMO-AIS and by *Orchidée* result from the use of SOO and MOO respectively. The MOO approach by *Orchidée* returns orchestrations that approximate the Pareto front, which is where the orchestrations with lowest F are in the objective space. However, each point on the Pareto front corresponds to a different combination of α_j. Consequently, the orchestrations along the Pareto front approach the reference sound from different perceptual directions. Therefore, each orchestration returned by *Orchidée* is the most similar to the reference sound according to different criteria emphasized by the different α_j. *Orchidée* prioritizes the objective similarity of Pareto optimal orchestrations over the perceptual similarity controlled by α_j. Consequently, the composer using *Orchidée* implicitly chooses a different perceptual direction by selecting an orchestration among the pool of solutions returned.

On the other hand, CAMO-AIS returns solutions that always approach the reference in the same direction, emphasizing the same perceptual similarities. Ultimately, α_j in CAMO-AIS are an aesthetic choice by the composer to determine the perceptual direction to search for orchestrations, allowing the composer to interactively explore the vast space of compositional possibilities. The trade-off is that, in theory, the timbral similarity between the orchestrations returned by the CAMO system and the reference decreases. CAMO-AIS returns orchestrations that correspond to local optima of the SOO fitness function, so the objective distance is not the smallest possible. Caetano *et al.* [15] compared the orchestrations returned by CAMO-AIS and *Orchidée* in terms of diversity and perceptual similarity with the reference. They showed that CAMO-AIS returns orchestrations with more diversity than *Orchidée* yet the systems did not differ in perceptual similarity with the reference. Therefore, CAMO-AIS provides more options to the composer without loss of perceptual similarity. Sec. 5.2 delves deeper into diversity of orchestrations.

5.2 Diversity of Orchestrations in CAMO-AIS

Some authors [51, 57] argue that CAMO systems that return multiple orchestrations present the composer with the challenge of choosing which one(s) to use. Instead, they suggest that there is a "best" solution to the imitative CAMO problem when it is posed correctly [51]. However, the CAMO framework described in this chapter does not narrow down the search space enough to admit only one solution. The descriptors of timbre used do not result in an exhaustive description such that multiple sounds would potentially match these descriptor values. In CAMO, this redundancy in the description of timbre translates as multiple instrument combinations approximating the reference timbral description.

Caetano *et al.* [15] argue that having multiple orchestrations provides aesthetic alternatives for the composer. The composer is rarely interested in a single combination (i.e., an orchestration) that optimizes some objective measure(s) with a reference sound [19]. Often, the composer uses CAMO tools to explore the problem space and find instrument combinations that would be missed by the empirical methods found in traditional orchestration manuals [63, 51]. The reference sound guides the search toward interesting regions of the search space and α_j fine-tune the relative importance of perceptual dimensions of timbre similarity encoded in the fitness function.

CAMO systems that return only one orchestration seldom meet the requirements of the highly subjective and creative nature of music composition [19]. Very often, the composer will use subjective criteria not encoded in the objective measure(s) guiding the search to choose one or more orchestrations of interest. In that case, diversity provides the composer with multiple choices when orchestrating a reference sound, expanding the creative possibilities of CAMO beyond what the composer initially imagined. From that perspective, a CAMO algorithm should be capable of returning several orchestrations that are all similar to the reference sound yet dissimilar among themselves, representing different alternative orchestrations for that reference sound. Thus, diversity of orchestrations allows the exploration of different musical ideas [15].

5.3 Dynamic Orchestrations with Orchids

Orchids was the first CAMO algorithm to allow dynamic orchestrations. However, the approach proposed by Esling *et al.* [33, 31, 32] has both technical and usability issues resulting from the time series matching algorithm used for dynamic orchestration. Since both issues are related, we will discuss the technical aspect first and then the usability problem related to it. The time series matching algorithm at the heart of Orchids [32] matches the *shape* of the temporal variation of the descriptors used. The algorithm includes two pre-processing steps prior to matching, namely *descriptor range normalization* and *dynamic time warping* (DTW). Normalization works along the axis of the descriptor (e.g., Hz for the spectral centroid), whereas DTW equalizes

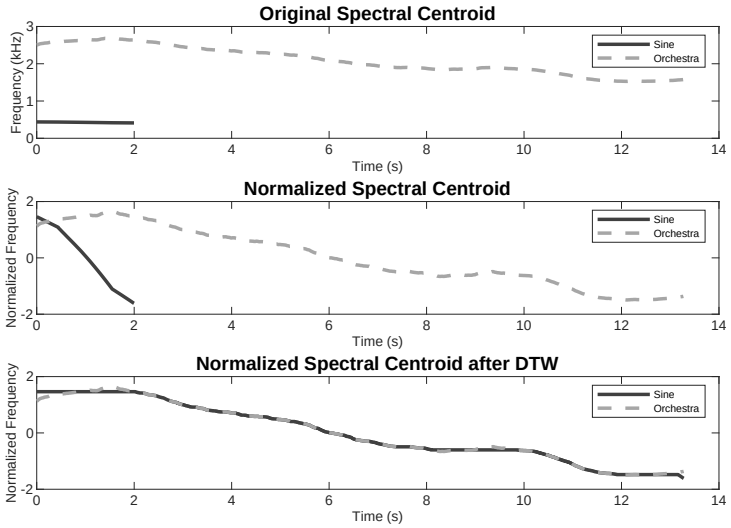


Fig. 8: The figure represents the spectral centroid of two radically different sounds. The top panel shows the original spectral centroids in kHz, the middle panel shows the spectral centroids after frequency range normalization, and the bottom panel shows the time-warped normalized spectral centroids. This kind of processing alters considerably the shape of the two spectral centroid curves, creating a mathematical match that is not representative of the perceptual similarity.

the sound duration along the temporal axis. Thus, both the range of descriptor values and the absolute duration associated with the original sounds are lost. These pre-processing steps have the undesired side effect of matching shapes in the normalized descriptor space that would not be considered similar in the original descriptor space. In practice, perceptually different sounds may be matched by Orchids.

Two radically different sounds were used in the example shown in Figure 8 to illustrate the issue. The first sound is a two-second pure sine wave whose frequency varies from 440 Hz down to 400 Hz (in total, a 40 Hz frequency range variation). The second sound is a 14-second long orchestral recording in which strings perform a downward *glissando* whose range is about 200 Hz. The top panel of Figure 8 shows the temporal variation of the spectral centroid of the original sounds, the sine wave is shown with a solid line and the strings with a dashed line. Note how the shapes differ radically because of the absolute values of both frequency and time. The middle panel shows the result of normalizing the range of descriptors, where the sine wave and the orchestral *glissando* now occupy the same normalized frequency range. Finally, the bottom panel of Figure 8 shows the result of applying DTW to both curves. Now, the shape of the two time series of descriptors is very similar and the algorithms behind Orchids would match them even if the two sounds are perceptually very different. Additionally, matching fast decays in energy, long

downward *glissandos*, slowly amplitude modulated sounds or fast *vibratos* requires instrument sounds played with these techniques in the musical instrument sound database. This could result in an exponentially increasing size of the database with problems in scalability.

Finally, from the perspective of the composer, using time series of descriptors places the focus on the micro-temporal scale and on the low-level aesthetic, perceptual, and musical aspects that this scale implies. Instead of thinking about musical elements such as chords, notes or musical scales, the user has to deal with time series of spectral descriptors that are difficult to relate to orchestration problems. Users who had musical training but no technical background reported having difficulty interpreting orchestration results and this difficulty naturally led to usability issues. While several composers showed interest in micro-temporal dynamic orchestration, a large share of the community did not manage to use this idea efficiently.

5.4 Dynamic Orchestrations with Orchidea

In Orchidea, dynamic orchestrations focus on a different temporal scale when compared with Orchids. Orchidea shifts the focus from the micro-temporal scale of milliseconds typical of time series of descriptors to the more musically meaningful temporal scale of musical notes. Orchidea breaks up the reference sound into a sequence of *events* that are orchestrated separately. First, Orchidea uses a two-stage optimization process in a high-dimensional descriptor space. Finally, Orchidea ensures continuation of the final orchestrations.

The main steps in Orchidea can be summarized as *segmentation*, *embedding*, *optimization*, and *continuation*. *Segmentation* determines the most important *musical events* in the reference sound with a *novelty*-based segmentation algorithm [35] that generates *sub-references* that are subsequently optimised separately. *Embedding* represents both the set of sub-references and the database of musical instrument sounds in a high-dimensional descriptor space. *Optimization* comprises a *preliminary* step followed by *refinement*. Stochastic matching pursuit performs the *preliminary* estimation of the orchestrations, followed by *refinement* with a GA performing SOO. Finally, a *continuation* model is applied on the selected solution for each sub-reference to minimize the number of changes for each instrument. Continuation is intended to improve the *voicing* of each player in the orchestra and to implement the orchestration principle of *dovetailing*: different instruments change notes at different times in order to maximize the blending of the orchestral colors (see pages 467-472 in [3]).

An interesting aspect of Orchidea is how it estimates the descriptors of instrument combinations. Given the high number of instrumental combinations generated during the optimization, it is impractical to synthesise a new audio file and then compute descriptors for every combination of sounds. Previous members of the Orch* family and CAMO-AIS estimate the descriptors of each candidate instrument combination using a simple energy-weighted linear combination (see Sec. 3.5), even if these

descriptors are not themselves linear [22]. Orchidea takes a different approach and estimates the new descriptors using a non-linear long short-term memory (LSTM) deep neural network (DNN). While the training phase of the predictor is time consuming, the estimation is very fast since it has a low complexity [36]. Refer to the Orchidea companion webpage [26] to listen to sound examples, download the system, and watch tutorial videos.

6 Conclusions

Musical orchestration remains one the most elusive aspects of musical composition to develop computer-assisted techniques for due to its highly empirical approach combined with the complexity of timbre perception. A major consequence of this lack or formalization is that computer-aided musical orchestration (CAMO) is still in its infancy relative to other aspects of musical composition, such as harmony and rhythm. This chapter focused on *imitative* CAMO methods aimed at helping composers find instrument combinations that replicate a given reference sound. Imitative CAMO is formalized as the search for a combination of instrument sounds from a database that minimizes the timbral distance captured by descriptors of timbre. Biologically inspired algorithms such as genetic algorithms (GA) and artificial immune systems (AIS) are commonly used to minimize a single-objective or multi-objective fitness function that encodes timbral similarity between the candidate orchestrations and the reference.

Several aspects of imitative CAMO deserve further investigation, such as orchestrating time-varying reference sounds with dynamic orchestrations [27] and proposing orchestrations that feature diversity [15]. Similarly, future research effort should be devoted to improving specific steps such as timbre similarity measures or the timbre of instrument combinations [36]. However, this formulation of imitative CAMO, albeit powerful, stems from a conceptual framework first laid out over a decade ago [21, 20, 17]. In particular, the current framework of imitative CAMO addresses musical orchestration from the narrow scope of instrumentation via timbre matching [51]. Recent developments in machine learning and computational intelligence have the potential to lead to a paradigm shift in CAMO that breaks free from the constraints of imitative CAMO into the next generation of CAMO systems that will address orchestration as a whole. For example, Piston [65] mentions *background and accompaniment* as well as *voice leading and counterpoint*, whereas Maresz [51] argues that "high-level orchestration is the art of combining simultaneous yet different sound layers." Each layer relies on specific musical parameters to provide an identity depending on the musical context. This high-level approach to orchestration would require a formalization that includes *descriptors of orchestral qualities* rather than descriptors of timbre. Currently, little is known about the timbre of instrumental music [54] to propel CAMO into full-fledged orchestration systems. Initiatives such as the ACTOR project [2] are currently investigating musical orchestration from

multiple perspectives to take the first steps in the exciting yet relatively unexplored world of computer-aided musical orchestration.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 831852 (MORPH)

References

1. Abreu, J., Caetano, M., Penha, R.: Computer-aided musical orchestration using an artificial immune system. In: C. Johnson, V. Ciesielski, J. Correia, P. Machado (eds.) Evolutionary and Biologically Inspired Music, Sound, Art and Design, pp. 1–16. Springer International Publishing (2016)
2. ACTOR: Actor project web page. `https://www.actorproject.org/` (2020). Accessed: 2020-06-18
3. Adler, S.: The study of orchestration. W. W. Norton and Company, London and New York (1989)
4. Antoine, A., Miranda, E.R.: Towards intelligent orchestration systems. In: 11th International Symposium on Computer Music Multidisciplinary Research (CMMR), pp. 671–681. Plymouth, UK (2015)
5. Antoine, A., Miranda, E.R.: Musical acoustics, timbre, and computer-aided orchestration challenges. In: Proceedings of the 2017 International Symposium on Musical Acoustics, pp. 151–154. Montreal, Canada (2017)
6. Antoine, A., Miranda, E.R.: A perceptually oriented approach for automatic classification of timbre content of orchestral excerpts. The Journal of the Acoustical Society of America **141**(5), 3723–3723 (2017). DOI 10.1121/1.4988156
7. Antoine, A., Miranda, E.R.: Predicting timbral and perceptual characteristics of orchestral instrument combinations. The Journal of the Acoustical Society of America **143**(3), 1747–1747 (2018). DOI 10.1121/1.5035706
8. Assayag, G., Rueda, C., Laurson, M., Agon, C., Delerue, O.: Computer-assisted composition at IRCAM: From PatchWork to OpenMusic. Computer Music Journal **23**(3), 59–72 (1999)
9. Barthet, M., Depalle, P., Kronland-Martinet, R., Ystad, S.: Acoustical correlates of timbre and expressiveness in clarinet performance. Music Perception: An Interdisciplinary Journal **28**(2), 135–154 (2010). DOI 10.1525/mp.2010.28.2.135
10. Biles, J.: GenJam: A Genetic Algorithm for Generating Jazz Solos. In: Proceedings of the International Computer Music Conference, pp. 131–131. International Computer Music Association (1994)
11. Caclin, A., McAdams, S., Smith, B., Winsberg, S.: Acoustic Correlates of Timbre Space Dimensions: A Confirmatory Study Using Synthetic Tones. The Journal of the Acoustical Society of America **118**(1), 471–482 (2005)
12. Caetano, M.: CAMO-AIS web page. `http://camo.prism.cnrs.fr/` (2019). Accessed: 2020-06-18
13. Caetano, M., Manzolli, J., Von Zuben, F.J.: Application of an artificial immune system in a compositional timbre design technique. In: C. Jacob, M.L. Pilat, P.J. Bentley, J.I. Timmis (eds.) Artificial Immune Systems, pp. 389–403. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
14. Caetano, M., Saitis, C., Siedenburg, K.: Audio content descriptors of timbre. In: K. Siedenburg, C. Saitis, S. McAdams, A.N. Popper, R.R. Fay (eds.) Timbre: Acoustics, Perception, and Cognition, pp. 297–333. Springer International Publishing, Cham (2019). DOI 10.1007/978-3-030-14832-4_11

15. Caetano, M., Zacharakis, A., Barbancho, I., Tardón, L.J.: Leveraging diversity in computer-aided musical orchestration with an artificial immune system for multi-modal optimization. Swarm and Evolutionary Computation **50**, 100484 (2019). DOI https://doi.org/10.1016/j.swevo.2018.12.010
16. Camacho, A., Harris, J.: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music. Journal of the Acoustical Society of America **124**(3), 1638–1652 (2008)
17. Carpentier, G.: Approche Computationnelle de L'Orchestration Musciale-Optimisation Multicritère sous Contraintes de Combinaisons Instrumentales dans de Grandes Banques de Sons. Ph.D. thesis, Université Pierre et Marie Curie-Paris VI (2008)
18. Carpentier, G., Assayag, G., Saint-James, E.: Solving the Musical Orchestration Problem using Multiobjective Constrained Optimization with a Genetic Local Search Approach. Journal of Heuristics **16**(5), 681–714 (2010)
19. Carpentier, G., Daubresse, E., Garcia Vitoria, M., Sakai, K., Villanueva, F.: Automatic orchestration in practice. Computer Music Journal **36**(3), 24–42 (2012). DOI 10.1162/COMJ_a_00136
20. Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E.: Imitative and Generative Orchestrations Using Pre-Analysed Sound Databases. In: Proceedings of the Sound and Music Computing Conference, pp. 115–122 (2006)
21. Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E.: An Evolutionary Approach to Computer-Aided Orchestration. In: M. Giacobini (ed.) Applications of Evolutionary Computing, *Lecture Notes in Computer Science*, vol. 4448, pp. 488–497. Springer Berlin Heidelberg (2007)
22. Carpentier, G., Tardieu, D., Harvey, J., Assayag, G., Saint-James, E.: Predicting Timbre Features of Instrument Sound Combinations: Application to Automatic Orchestration. Journal of New Music Research **39**(1), 47–61 (2010)
23. de Castro, L., Von Zuben, F.: Learning and Optimization Using the Clonal Selection Principle. Evolutionary Computation, IEEE Transactions on **6**(3), 239–251 (2002)
24. de Castro, L.N., Timmis, J.: An artificial immune network for multimodal function optimization. In: Proceedings of the Congress on Evolutionary Computation, CEC'02., vol. 1, pp. 699–704. IEEE (2002)
25. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A new Computational Intelligence Approach. Springer (2002)
26. Cella, C.E.: Orchidea web page. `www.orch-idea.org` (2020). Accessed: 2020-06-18
27. Cella, C.E., Esling, P.: Open-source modular toolbox for computer-aided orchestration. In: Proceedings of Timbre 2018: Timbre Is a Many-Splendored Thing, pp. 93–94. Montreal, Canada (2018)
28. Clay-Mathematics-Institute: P vs NP Problem. `https://www.claymath.org/millennium-problems/p-vs-np-problem` (2020). Accessed: 2020-06-18
29. Dasgupta, D.: Advances in artificial immune systems. IEEE Computational Intelligence Magazine **1**(4), 40–49 (2006). DOI 10.1109/MCI.2006.329705
30. De Jong, K.: Learning with genetic algorithms: An overview. Machine Learning **3**(2-3), 121–138 (1988). DOI 10.1007/BF00113894
31. Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. **45**(1), 1–34 (2012). DOI 10.1145/2379776.2379788
32. Esling, P., Agon, C.: Multiobjective time series matching for audio classification and retrieval. IEEE Transactions on Audio, Speech, and Language Processing **21**(10), 2057–2072 (2013). DOI 10.1109/TASL.2013.2265086
33. Esling, P., Carpentier, G., Agon, C.: Dynamic Musical Orchestration Using Genetic Algorithms and a Spectro-Temporal Description of Musical Instruments, *Lecture Notes in Computer Science*, vol. 6025, pp. 371–380. Springer Berlin Heidelberg (2010)
34. EvoStar: EvoStar web page. `www.evostar.org` (2019). Accessed: 2020-06-18
35. Foote, J., Cooper, M.L.: Media segmentation using self-similarity decomposition. In: Proceedings of SPIE 5021, Storage and Retrieval for Media Databases 2003, January 10, 2003; doi: 10.1117/12.476302 (2003)

36. Gillick, J., Cella, C.E., Bamman, D.: Estimating unobserved audio features for target-based orchestration. In: Proceedings of the 20th International Society for Music Information Retrieval Conference, pp. 192–199. Delft, the Netherlands (2019)
37. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Professional, Reading, MA (1989)
38. Goldreich, O.: P, NP, and NP-Completeness: The Basics of Computational Complexity. Cambridge University Press, New York, NY (2010)
39. Greensmith, J., Whitbrook, A., Aickelin, U.: Artificial immune systems. In: M. Gendreau, J.Y. Potvin (eds.) Handbook of Metaheuristics, *International Series in Operations Research & Management Science*, vol. 146, pp. 421–448. Springer US, Boston, MA (2010). DOI 10.1007/978-1-4419-1665-5_14
40. Grey, J.: Multidimensional Perceptual Scaling of Musical Timbres. The Journal of the Acoustical Society of America **61**(5), 1270–1277 (1977)
41. Grey, J., Gordon, J.: Perceptual Effects of Spectral Modifications on Musical Timbres. The Journal of the Acoustical Society of America **63**(5), 1493–1500 (1978)
42. Handelman, E., Sigler, A., Donna, D.: Automatic Orchestration for Automatic Composition. In: 1st International Workshop on Musical Metacreation (MUME 2012), pp. 43–48. AAAI (2012)
43. Helmholtz, H.: On the Sensations of Tone as a Physiological Basis for the Theory of Music. Longmans, Green, and Co., London, New York (1895)
44. Hummel, T.: Simulation of Human Voice Timbre by Orchestration of Acoustic Music Instruments. In: Proceedings of the International Computer Music Conference (ICMC), p. 185 (2005)
45. Jaszkiewicz, A.: Genetic local search for multiple objective combinatorial optimization. European Journal of Operational Research **1**(137), 50–71 (2002)
46. Karp, R.M.: Reducibility among combinatorial problems. In: R.E. Miller, J.W. Thatcher (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press, New York (1972)
47. Kendall, R.A., Carterette, E.C.: Identification and blend of timbres as a basis for orchestration. Contemporary Music Review **9**(1-2), 51–67 (1993). DOI 10.1080/07494469300640341
48. Kopiez, R., Wolf, A., Platz, F., Mons, J.: Replacing the orchestra? - The discernibility of sample library and live orchestra sounds. PLOS ONE **11**(7), 1–12 (2016). DOI 10.1371/journal.pone.0158324
49. Krimphoff, J., McAdams, S., Winsberg, S.: Caractérisation du Timbre des Sons Complexes.II. Analyses Acoustiques et Quantification Psychophysique. J. Phys. IV France **04**(C5), 625–628 (1994)
50. Krumhansl, C.L.: Why is Musical Timbre so Hard to Understand? Structure and perception of electroacoustic sound and music **9**, 43–53 (1989)
51. Maresz, Y.: On computer-assisted orchestration. Contemporary Music Review **32**(1), 99–109 (2013). DOI 10.1080/07494467.2013.774515
52. Mathews, M.V., Miller, J.E., Pierce, J.R., Tenney, J.: Computer study of violin tones. The Journal of the Acoustical Society of America **38**(5), 912–913 (1965). DOI 10.1121/1.1939649
53. McAdams, S.: Timbre as a structuring force in music. In: K. Siedenburg, C. Saitis, S. McAdams, A.N. Popper, R.R. Fay (eds.) Timbre: Acoustics, Perception, and Cognition, pp. 211–243. Springer International Publishing, Cham (2019). DOI 10.1007/978-3-030-14832-4_11
54. McAdams, S.: Timbre as a structuring force in music. In: K. Siedenburg, C. Saitis, S. McAdams, A.N. Popper, R.R. Fay (eds.) Timbre: Acoustics, Perception, and Cognition, pp. 211–243. Springer International Publishing, Cham (2019). DOI 10.1007/978-3-030-14832-4_8. URL `https://doi.org/10.1007/978-3-030-14832-4_8`
55. McAdams, S., Giordano, B.L.: The Perception of Musical Timbre. In: S. Hallam, I. Cross, M. Thaut (eds.) The Oxford Handbook of Music Psychology, pp. 72–80. Oxford University Press, New York, NY (2009)
56. McAdams, S., Winsberg, S., Donnadieu, S., De Soete, G., Krimphoff, J.: Perceptual Scaling of Synthesized Musical Timbres: Common Dimensions, Specificities, and Latent Subject Classes. Psychological Research **58**(3), 177–192 (1995)

57. Miranda, E.R., Antoine, A., Celerier, J.M., Desainte-Catherine, M.: i-Berlioz: Towards interactive computer-aided orchestration with temporal control. International Journal of Music Science, Technology and Art **1**(1), 15–23 (2019)
58. Miranda, E.R., Biles, J.A. (eds.): Evolutionary Computer Music. Springer, London (2007). DOI 10.1007/978-1-84628-600-1
59. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA (1996)
60. Moroni, A., Manzolli, J., Von Zuben, F., Gudwin, R.: Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition. Leonardo Music Journal **10**, 49–54 (2000)
61. Navarro, M., Caetano, M., Bernandes, G., de Castro, L., Corchado, J.: Automatic Generation of Chord Progressions with an Artificial Immune System. In: Proceedings of EVOMUSART 2015 (2015)
62. Navarro-Cáceres, M., Caetano, M., Bernardes, G., de Castro, L.N.: ChordAIS: An assistive system for the generation of chord progressions with an artificial immune system. Swarm and Evolutionary Computation **50**, 100543 (2019). DOI https://doi.org/10.1016/j.swevo.2019.05.012
63. Nouno, G., Cont, A., Carpentier, G., Harvey, J.: Making an Orchestra Speak. In: Sound and Music Computing. Porto, Portugal (2009)
64. Peeters, G., Giordano, B.L., Susini, P., Misdariis, N., McAdams, S.: The timbre toolbox: Extracting audio descriptors from musical signals. The Journal of the Acoustical Society of America **130**(5), 2902–2916 (2011). DOI 10.1121/1.3642604
65. Piston, W.: Orchestration. W. W. Norton & Company, London (1955)
66. Psenicka, D.: SPORCH: An Algorithm for Orchestration Based on Spectral Analyses of Recorded Sounds. In: Proceedings of International Computer Music Conference (ICMC), p. 184 (2003)
67. Risset, J.C.: Computer study of trumpet tones. The Journal of the Acoustical Society of America **38**(5), 912–912 (1965). DOI 10.1121/1.1939648
68. Romero, J., Machado, P. (eds.): The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music. Natural Computing Series. Springer, Berlin, Heidelberg (2007)
69. Rose, F., Hetrik, J.E.: Enhancing Orchestration Technique via Spectrally Based Linear Algebra Methods. Computer Music Journal **33**(1), 32–41 (2009)
70. Saitis, C., Giordano, B.L., Fritz, C., Scavone, G.P.: Perceptual evaluation of violins: A quantitative analysis of preference judgments by experienced players. The Journal of the Acoustical Society of America **132**(6), 4002–4012 (2012). DOI 10.1121/1.4765081
71. Siedenburg, K., Jones-Mollerup, K., McAdams, S.: Acoustic and categorical dissimilarity of musical timbre: Evidence from asymmetries between acoustic and chimeric sounds. Frontiers in Psychology **6**, 1977 (2016). DOI 10.3389/fpsyg.2015.01977
72. Siedenburg, K., McAdams, S.: Four distinctions for the auditory "wastebasket" of timbre. Frontiers in Psychology **8**, 1747 (2017). DOI 10.3389/fpsyg.2017.01747
73. Siedenburg, K., Saitis, C., McAdams, S.: The present, past, and future of timbre research. In: K. Siedenburg, C. Saitis, S. McAdams, A.N. Popper, R.R. Fay (eds.) Timbre: Acoustics, Perception, and Cognition, pp. 1–19. Springer International Publishing, Cham (2019). DOI 10.1007/978-3-030-14832-4_1
74. Slawson, W.: Sound Color. University of California Press, Berkeley (1985)
75. Springer: Proceedings of the International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar). `https://link.springer.com/conference/evomusart` (2020). Accessed: 2020-06-18
76. Tardieu, D., Rodet, X.: An Instrument Timbre Model for Computer Aided Orchestration. In: Applications of Signal Processing to Audio and Acoustics, 2007 IEEE Workshop, pp. 347–350. IEEE (2007)
77. Timmis, J., Knight, T., de Castro, L.N., Hart, E.: An overview of artificial immune systems. In: R. Paton, H. Bolouri, M. Holcombe, J.H. Parish, R. Tateson (eds.) Computation in Cells and Tissues: Perspectives and Tools of Thought, pp. 51–91. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). DOI 10.1007/978-3-662-06369-9_4
78. Todd, S., Latham, W.: Evolutionary Art and Computers. Academic Press, Inc., USA (1994)
79. Yang, X.S.: Multi-objective optimization. In: X.S. Yang (ed.) Nature-Inspired Optimization Algorithms, pp. 197–211. Elsevier, Oxford (2014). DOI 10.1016/B978-0-12-416743-8.00014-2