Computer-Aided Musical Orchestration Using an Artificial Immune System

José Abreu[1(✉)], Marcelo Caetano[2], and Rui Penha[1,2]

[1] Faculty of Engineering, University of Porto, Porto, Portugal
{ee10146,ruipenha}@fe.up.pt
[2] Sound and Music Computing Group, INESC TEC, Porto, Portugal
mcaetano@inesctec.pt

Abstract. The aim of computer-aided musical orchestration is to find a combination of musical instrument sounds that approximates a target sound. The difficulty arises from the complexity of timbre perception and the combinatorial explosion of all possible instrument mixtures. The estimation of perceptual similarities between sounds requires a model capable of capturing the multidimensional perception of timbre, among other perceptual qualities of sounds. In this work, we use an artificial immune system (AIS) called opt-aiNet to search for combinations of musical instrument sounds that minimize the distance to a target sound encoded in a fitness function. Opt-aiNet is capable of finding multiple solutions in parallel while preserving diversity, proposing alternative orchestrations for the same target sound that are different among themselves. We performed a listening test to evaluate the subjective similarity and diversity of the orchestrations.

1 Introduction

Orchestration refers to composing music for an orchestra [12]. Initially, orchestration was simply the assignment of instruments to pre-composed parts of the score, which was dictated largely by availability of resources, such as what instruments there are and how many of them [10,12]. Later on, composers started regarding orchestration as an integral part of the compositional process whereby the musical ideas themselves are expressed [18]. Compositional experimentation in orchestration arises from the increasing tendency to specify instrument combinations to achieve desired effects, resulting in the contemporary use of timbral combinations [15,18]. 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 computer-aided musical orchestration (CAMO) [3–6,11,17,18]. Most of these tools rely on searching for combinations of musical instrument sounds from pre-recorded datasets to approximate a given target sound. Early works [11,17,18] resorted to spectral analysis followed by subtractive spectral matching.

Psenicka [17] 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

C. Johnson et al. (Eds.): EvoMUSART 2016, LNCS 9596, pp. 1–16, 2016.
DOI: 10.1007/978-3-319-31008-4_1

and quality approximate that of the analyzed sound." The method keeps a database of spectral peaks estimated from either the steady state or the attack (for nonpercussive and percussive sounds, respectively) of musical instrument sounds organized according to pitch, dynamic level, and playing technique such as *staccato* and *vibrato*. The algorithm iteratively subtracts the spectral peaks of the best match from the target 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. Hummel [11] 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 target and candidate sounds, disregarding pitch among other perceptual qualities. Rose and Hetrik [18] use singular value decomposition (SVD) to perform spectral decomposition and spectral matching using a database of averaged DFTs of musical instrument sounds containing different pitches, dynamic levels, and playing techniques. SVD decomposes the target 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.

The concept of timbre lies at the core of musical orchestration. Yet, timbre perception is still only partially understood [1,9,13,15,16]. The term timbre encompasses auditory attributes, perceptual and musical issues, covering perceptual parameters not accounted for by pitch, loudness, spatial position, duration, among others [13,15]. Nowadays, timbre is regarded as both a multidimensional set of sensory attributes that quantitatively characterize the ways in which sounds are perceived to differ and the primary vehicle for sound source recognition and identification [15]. McAdams and Bruno [15] wrote that "instrumental combinations can give rise to new timbres if the sounds are perceived as blended, and timbre can play a role in creating and releasing musical tension." Consequently, the goal of CAMO is to find an instrument combination that best approximates the target timbre rather than the target spectrum [19].

To overcome the drawbacks of subtractive spectral matching, Carpentier and collaborators [3–6,19] search for a combination of musical instrument sounds whose timbral features best match those of the target sound. This approach requires a model of timbre perception to describe the timbre of isolated sounds, a method to estimate the timbral result of an instrument combination, and a measure of timbre similarity to compare the combinations and the target. Multidimensional scaling (MDS) of perceptual dissimilarity ratings [1,9,13,16] provides a set of auditory correlates of timbre perception that are widely used to model timbre perception of isolated musical instrument sounds. MDS spaces are obtained by equating distance measures to timbral (dis)similarity ratings.

In metric MDS spaces, the distance measure directly allows timbral comparison. Models of timbral combination [6,12] estimate these features for combinations of musical instrument sounds.

Carpentier and collaborators [3–6,19] consider the search for combinations of musical instrument sounds as a constrained combinatorial optimization problem [5]. They formulate CAMO as a variation of the knapsack problem where the aim is to find a combination of musical instruments that maximizes the timbral similarity with the target constrained by the capacity of the orchestra (i.e., the database). The binary allocation knapsack problem can be shown to be NP-complete so it cannot be solved in polynomial time. They explore the vast space of possible instrument combinations with a genetic algorithm (GA) that optimizes a fitness function which encodes timbral similarity between the candidate instrument combinations and the target sound. GAs are metaheuristics inspired by the Darwinian principle of *survival of the fittest*. The GA maintains a list of individuals that represent the possible combinations of instruments. These individuals evolve towards optimal solutions by means of crossover, mutation, and selection. Crossover and mutation are responsible for introducing variations in the current population and promoting the exploration and exploitation of the search space. Selection guarantees that the fittest individuals are passed to the next generation gradually converging to optimal regions of the search space. The major drawback of this approach arises from the loss of diversity inherent in the evolutionary search performed with GAs. In practice, the loss of diversity results in only one solution that commonly corresponds to a local optimum because GAs cannot guarantee to return the global optimum (i.e., the best solution). Moreover, running the GA multiple times with the same parameters commonly results in different solutions. Carpentier *et al.* [5] use a combination of local search and constraint strategies to circumvent the issues resulting from loss of diversity.

In this work, we use an artificial immune system (AIS) called opt-aiNet [7] to search for multiple combinations of musical instrument sounds whose timbral features match those of the target sound. Inspired by immunological principles, opt-aiNet returns multiple good quality solutions in parallel while preserving diversity. The intrinsic property of maintenance of diversity allows opt-aiNet to return all the optima (global and local) of the fitness function being optimized upon convergence, which translates as orchestrations that are all similar to the target yet different from one another. The AIS provides the composer with multiple choices when orchestrating a sound instead of searching for one solution constrained by choices defined *a priori*. Therefore, our work can expand the creative possibilities of CAMO beyond what the composer initially imagined.

The remainder of this paper is organized as follows. The next section presents an overview of our approach to CAMO. Then we describe the immunological approach to CAMO. Next we present the experiment we performed followed by the evaluation. The evaluation comprises similarity and diversity using objective measures and the subjective ratings from a listening test. We present and discuss the results, followed by conclusions and perspectives.

2 Computer-Aided Musical Orchestration (CAMO)

2.1 Overview

Figure 1 shows an overview of our approach. The sound database is used to build a feature database, which consists of acoustic features calculated for all sounds prior to the search for orchestrations. The same features are calculated for the target sound being orchestrated. The fitness function uses these features to estimate the similarity between combinations of features from sounds in the database and those of the target sound. The AIS is used to search for combinations that approximate the target sound, called orchestrations. Each orchestration is a list of sounds from the sound database, which contains sounds with various lengths. A phase vocoder is used to time-stretch or compress each sound from an orchestration to the average duration to ensure they all start and end at the same time when played together. The graphic interface (GUI) displays information about the solution set and allows the user to play the target sound and the orchestrations.

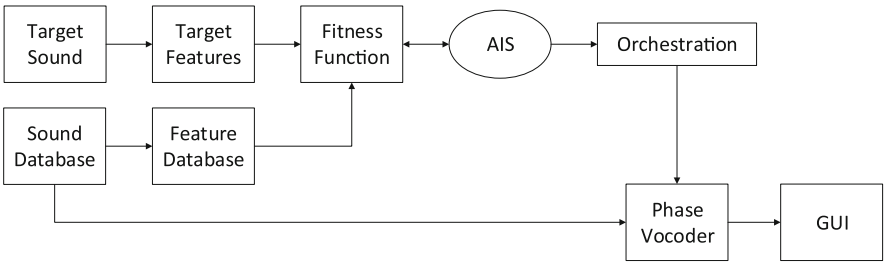


Fig. 1. Overview of the modules that compose the developed orchestration system

2.2 Sound Database

The sound database used in this work contains musical instrument sounds from the RWC Music Database [8] available to compose the orchestrations. In total, there are 1439 sounds from 13 instruments played with 3 dynamics, *forte*, *mezzo forte* and *piano*. The instruments are *violin*, *viola*, *cello*, *contrabass*, *trumpet*, *trombone*, *tuba*, *french horn*, *english horn*, *oboe*, *bassoon*, *clarinet*, and *flute*. For each file the values of the sound features described in the next section were computed and stored.

2.3 Feature Database

Traditionally, timbre is considered as the set of attributes whereby a listener can judge that two sounds are dissimilar using any criteria other than pitch, loudness,

or duration [15]. Therefore, we consider pitch, loudness, and duration separately from timbre dimensions. The features used are fundamental frequency f_0 (pitch), frequency and amplitude of the contribution spectral peaks P, loudness λ, spectral centroid μ, and spectral spread σ. The spectral centroid μ captures brightness while the spectral spread σ correlates with the third dimension of MDS timbre spaces [1,9,13,16]. The RMS energy was also calculated for each sound and the duration is equalized later with a phase vocoder. All musical instrument sounds used are sustained (i.e., nonpercussive) with attack times longer than 250 ms and duration of 1 s or more. The calculation of the features is performed for short-term frames between 250 ms and 750 ms and then averaged because the signal is considered stable in that region.

Fundamental Frequency. The f_0 of all sounds $s(i)$ in the database is estimated with Swipe [2].

Contribution Spectral Peaks. The spectral peaks considered are those whose spectral energy (amplitude squared) is at most 35 dB below the maximum. They are estimated with the MIR Toolbox [14] and stored as a vector with the pairs $\{a(n), f(n)\} \in s(i)$. The spectral peaks $a(n)$ are used to compute the *contribution* spectral peaks $P(i,n)$, which are the spectral peaks from the *selected* sound $s(i)$ that are common to the spectral peaks of the *target* sound s^T. Equation (1) shows the calculation of $P(i,n)$ as

$$P(i,n) = \begin{cases} a^i(n) & \text{if } (1+\delta)^{-1} \leq f^i(n)/f^T(n) \leq 1+\delta \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $a^i(n)$ is the amplitude and $f^i(n)$ is the frequency of the main spectral peak of the *selected* sound, and $f^T(n)$ is the frequency of the *target* sound. Figure 2 illustrates the computation of spectral peak similarity between the *target* sound and a *selected* sound. Spectral peaks are represented as spikes with amplitude $a(n)$ at frequency $f(n)$ where n is the index of the peak. The frequencies $f^T(n)$ of the peaks of the *target* sound are used as reference. Whenever the *selected* sound contains a peak in a region δ around $f^T(n)$, the amplitude $a^i(n)$ of the peak at frequency $f^i(n)$ of the *selected* sound is kept at position n of the contribution vector $P(n)$. In this work $\delta = 0.025$.

Loudness. Loudness $\lambda(i)$ is calculated as

$$\lambda(i) = 20\log_{10}\left(\sum_n a(n)\right), \tag{2}$$

where $a(n)$ are the amplitudes at frequencies $f(n)$ and i is the sound index.

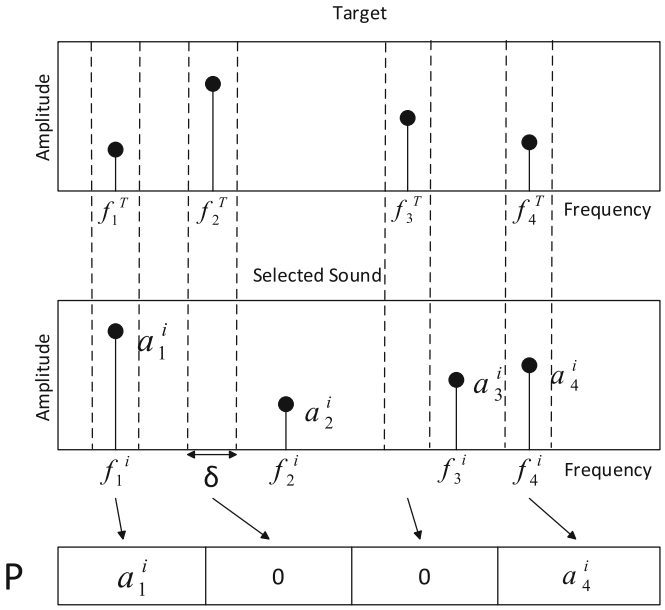


Fig. 2. Construction of the contribution vector $P(i,n)$. See text for explanation.

Spectral Centroid. The spectral centroid $\mu(i)$ is calculated as

$$\mu(i) = \sum_n f(n) \frac{|a(n)|^2}{\sum_n |a(n)|^2}. \tag{3}$$

Spectral Spread. The spectral spread $\sigma(i)$ is calculated as

$$\sigma(i) = \sum_n (f(n) - \mu)^2 \frac{|a(n)|^2}{\sum_n |a(n)|^2}. \tag{4}$$

2.4 Pre-processing

Prior to the search for orchestrations of a given target sound s^T, the entire sound database S is reduced to a subset S^T of sounds that will be effectively used to compose orchestrations for s^T. All the sounds whose contribution vector $P(i,n)$ is all zeros are eliminated because these do not have any contribution spectral peaks. Similarly, all the sounds whose f_0 is lower than f_0^T are eliminated because any partials lower than f_0^T have a negative impact on the final result. Partials that are higher than all $P^T(n)$ have a negligible effect and are not considered.

2.5 Representation

An orchestration is a list of sounds $S(i)$ that, when played together, should approximate the target sound s^T. Thus orchestrations are represented as $S(i) =$

$\{s(1), s(2), \ldots, s(i), \ldots, s(I)\}, \quad \forall s(i) \in S^T$. In practice, $S(i)$ has I sounds, and each sound $s(i)$ corresponds to a note of a given instrument played with a dynamic level. Zero indicates no instrument.

2.6 Combination Functions

The sounds $s(i)$ in an orchestration $S(i)$ should approximate the target s^T when played together. Therefore, the combination functions estimate the values of the spectral features of $S(i)$ from the features of the isolated sounds $s(i)$ normalized by the RMS energy $e(i)$ [6]. The combination functions for the spectral centroid $\mu(i)$, spectral spread $\sigma(i)$, and loudness $\lambda(i)$ are given respectively by

$$\mu(S(i)) = \frac{\sum_i^I e(i)\,\mu(i)}{\sum_i^I e(i)} \tag{5}$$

$$\sigma(S(i)) = \sqrt{\frac{\sum_i^I e(i)\,(\sigma^2(i) + \mu^2(i))}{\sum_i^I e(i)} - \mu^2(S(i))} \tag{6}$$

$$\lambda(S(i)) = 20\log_{10}\left(\sum_i^I \frac{1}{N}\sum_n a(i,n)\right) \tag{7}$$

The estimation of the contribution spectral peaks of the combination $P(S(i), n)$ uses the contribution vectors $P(i,n)$ of the sounds $s(i)$ in $S(i)$ as

$$P(S(i), n) = \left\{\max_{i\in I}[P(i,1)], \max_{i\in I}[P(i,2)], \cdots, \max_{i\in I}[P(i,N)]\right\} \tag{8}$$

2.7 Fitness Function

The fitness value $F(S(i))$ of an orchestration $S(i)$ is calculated as

$$F(S(i)) = -\sum_y \alpha(y)\, D_y \tag{9}$$

where $\alpha(y)$ are the weights that establish the relative importance of the absolute distances D_y. Each D_y, in turn, measures the difference between the features from the target sound s^T and the candidate orchestration $S(i)$ as follows

$$D_\mu = \frac{|\mu(S(i)) - \mu(s^T)|}{\mu(s^T)} \tag{10}$$

$$D_\sigma = \frac{|\sigma(S(i)) - \sigma(s^T)|}{\sigma(s^T)} \tag{11}$$

$$D_\lambda = \frac{|\lambda(S(i)) - \lambda(s^T)|}{\lambda(s^T)} \tag{12}$$

The use of Eqs. (10) and (11) is specially suited for CAMO problems as these measures are more sensitive for lower frequencies, a fact that is desired considering human perception of sound. The distance between the contribution vector of the target sound $P(s^T, n)$ and the contribution vector of the orchestration $P(S(i), n)$ is calculated as

$$D_P = 1 - \cos(P(S(i), n), P(s^T, n)). \tag{13}$$

The weights used in this work are

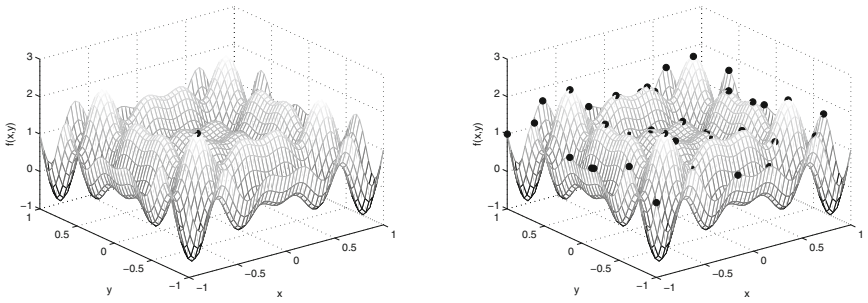
$$\alpha = \{0.1, 0.1, 0.2, 0.6\}. \tag{14}$$

The aim of CAMO is to find $S(i)$ that is close to s^T. Thus we want to minimize the distances D_y that comprise F. The negative sign in Eq. (9) makes the fitness value of all combinations negative so maximizing F approaches zero and minimizes the distance from s^T. However, the fitness landscape of $F(S(i))$ depends on the combinations $S(i)$ in S^T, giving rise to a complex space which requires an optimization method to find solutions.

3 Immune Inspired Musical Orchestration

The work of Carpentier *et al.* [3–6,19] represents a paradigm shift in CAMO. Prior to their work, most approaches [11,17,18] used greedy search procedures based on subtractive spectral matching. Their contribution is twofold, the use of perceptually related features to measure timbral similarity and the use of constrained combinatorial optimization to search for combinations of musical instruments whose features approach those of the target sound. Timbral similarity is encoded in a fitness function such that better combinations present higher fitness values. The aim is to find combinations that correspond to maxima of the fitness function as illustrated in Fig. 3. The surfaces represent the fitness function, which might have multiple peaks, and the black dots represent the fitness values of specific instrument combinations. The combinatorial explosion resulting from the exhaustive search of all possible combinations requires heuristics to find a solution in less time.

Carpentier *et al.* [3–6,19] use GAs to perform the search due to their ability to perform exploration and exploitation of the search space. Exploration is responsible for looking for new promising regions of the search space (peaks of the fitness function) and exploitation climbs the peaks looking to improve the current candidate solutions. However, the standard GA suffers from loss of diversity upon convergence, which results in only one solution corresponding to one peak is returned by the GA as shown in Fig. 3a. The stochastic nature of the search procedure does not guarantee that the global optimum is found, often getting stuck in local optima. Additionally, running the GA multiple times with



(a) Genetic Algorithm (GA) (b) Artificial Immune System (AIS)

Fig. 3. Multimodal function optimization. The figure illustrates a fitness function with multiple optima. Part (a) shows that the GA finds only one optimum. Part (b) shows the ability of the AIS to find all the optima of the fitness function.

the same parameters commonly results in different solutions corresponding to different peaks of the fitness function.

We propose an immune inspired approach to CAMO instead. We use an artificial immune system (AIS) called opt-aiNet [7] to perform the search. Figure 3b illustrates the ability of opt-aiNet to find all optima of the fitness function preserving diversity. The ability to maintain diversity translates as solutions that are different from one another.

3.1 Opt-aiNet: An Artificial Immune System for Optimization

Inspired by the natural immune system, De Castro and Timmis [7] developed an artificial immune system (AIS) called opt-aiNet for multimodal optimization problems, which typically present several possible solutions as optima of the fitness function. Opt-aiNet uses the immunological principles of clonal expansion, mutation and suppression to evolve a population of antibodies in an immune network. Opt-aiNet combines local and global search to locate and maintain multiple optima of the fitness function in parallel while preserving diversity of the solutions. This means that opt-aiNet can find a set of good candidates for the solution of the optimization problem that are different from one another.

Each network cell (antibody) is represented as a vector whose fitness is measured with a fitness function. Additionally, the similarity among antibodies is called affinity, and a high affinity means that the antibodies are similar. Affinity is measured with a distance metric such as the Euclidean distance. The antibodies are initialized at random to explore the search space. Some high fitness antibodies are selected and cloned based on their fitness value, the higher the fitness, the higher the number of clones and vice-versa. The clones generated suffer a mutation inversely proportional to their fitness and a number of high fitness clones is maintained in the network as memory. Then, the affinity among the remaining antibodies is determined. Maintenance of diversity is achieved by

eliminating the antibodies whose affinity is lower than a given threshold from the network while keeping the ones with the highest fitness. Finally, a number of newly generated antibodies are incorporated into the network.

Discrete Search Space. Originally, opt-aiNet [7] was designed to optimize functions of continuous variables, performing the search in continuous vector spaces. In our work, the search space is discrete because the representation of orchestrations $S(i)$ is a vector of discrete indices i of sounds in the database. Most of the operations of the continuous version of opt-aiNet work as originally intended for discrete vectors as well. The exception is the original mutation operator which used a continuous random variable to add a small perturbation to the vectors being mutated. Thus we adapted the mutation operator for discrete vectors using a probability of mutation to determine if the vector will undergo mutation. The probability of mutation p_m is calculated as

$$p_m = \exp(-\gamma \hat{F}) \tag{15}$$

where γ is a constant and $\hat{F}$ is the normalized fitness value of the combination vector $S(i)$ being mutated. For each index i, a uniform random variable $u(0,1)$ will determine if the corresponding sound $s(i)$ is replaced by another sound from S^T. If $u(0,1) < p_m$ then a new $i \in S^T$ is chosen from another uniform distribution. Here we set $\gamma = 1.2$. The suppression operation discards cells that have affinity values below a given threshold. In this work, the affinity between two antibodies is the distance between vectors whose components are calculated using Eqs. (10)–(13).

3.2 Phase Vocoder

The orchestrations found by the AIS are created as combinations of the sounds from the database, which have different temporal duration. The focus on timbral similarity requires to equalize the other dimensions of sound perception, including the duration of the sounds. The Phase Vocoder (PV)[1] can manipulate the pitch and duration independently, allowing to create combinations of sounds from the database with the same duration while preserving the pitch and other perceptual features.

3.3 Graphical Interface

Figure 4 shows the graphical interface (GUI) that displays the orchestrations proposed by the AIS. The GUI allows to play the orchestrations and shows the instruments comprised in them. The spectrum of the target sound and the orchestrations can also be viewed.

[1] http://labrosa.ee.columbia.edu/matlab/pvoc/.

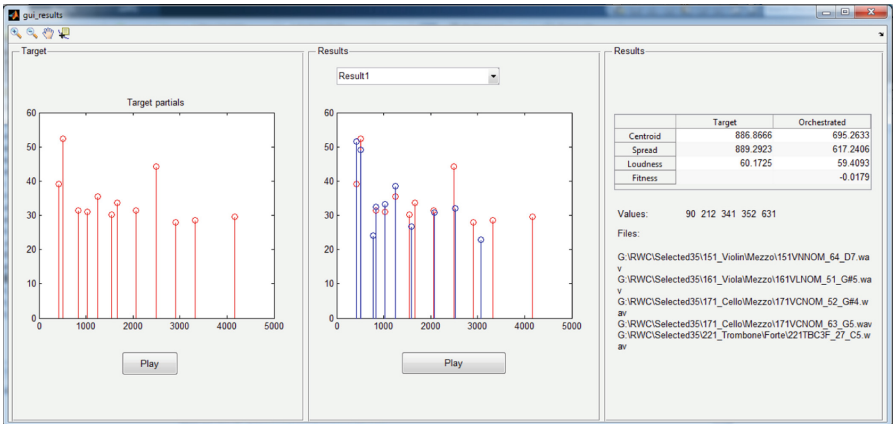


Fig. 4. Graphical interface (GUI).

4 Evaluation

The aim of the evaluation was to investigate the similarity and diversity of the orchestrations proposed by our system. The quality of a solution depends on how similar it is to the target sound. We want all the solutions proposed by the system to be as close to the target sound as possible. However, diversity is also important. Multiple solutions should be different from one another to represent alternatives, giving the user options to choose from. Therefore, we evaluate the similarity and the diversity of the orchestrations proposed by the system and compare them with an implementation of a genetic algorithm.

We performed subjective and objective evaluations for similarity and diversity. The subjective evaluation consists of a listening test, and the objective evaluation uses distance measures. For the listening test, we selected 10 target sounds, listed in Table 1. These sounds were chosen according to two criteria, *temporal variation* and *harmonicity*. Target sounds that have high temporal variation will tend to be more challenging to orchestrate because we use the average value of the features. Similarly, target sounds that are less harmonic will be more challenging to orchestrate with musical instrument sounds because all the musical instruments in the database used to find orchestrations are very close to

Table 1. Target sounds used in the listening test. The table contains an informal estimation of the degree of temporal variation (TempVar) and the degree of harmonicity (Harm) of each target sound as low (L) or high (H).

Target	Horn	Synth	Tbreed	Ahh	Harp	Didger	Eleph	Frog	Scream	Gong
TempVar	L	L	L	L	L	L	H	H	H	H
Harm	L	H	L	H	H	H	L	L	L	L

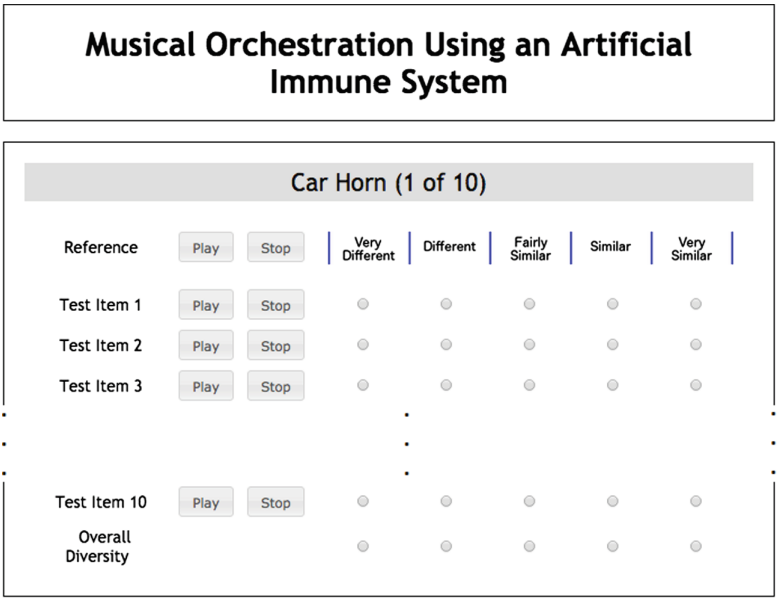


Fig. 5. Illustration of the listening test

harmonic. The listening test shown in Fig. 5 was run online[2]. The target sound is marked as *reference*, and the orchestrations are *test item*. After playing the sounds multiple times if necessary, the listeners were asked to assess the *subjective similarity* between the reference and each test item using the following scale: *very different*, *different*, *fairly similar*, *similar*, and *very similar*. Finally, the listeners were asked to judge the *overall diversity* of the test items using the same scale. In total, 23 people took the test.

The objective evaluation of similarity uses the fitness values of the solutions, while the objective evaluation of diversity uses Eq. (16), which quantifies the number of sounds in common between the orchestrations present in the set as:

$$div = 1 - \frac{k}{I}, \tag{16}$$

where k is the number of common sounds and I the total number of sounds in an orchestration. This equation quantifies the objective diversity as a value between 0 and 1, where 0 corresponds to minimum diversity and 1 corresponds to maximum diversity.

5 Results and Discussion

5.1 Subjective Evaluation

The results obtained for the subjective evaluation are shown in Table 2. Subjective similarity varies from 1 to 5 with higher values corresponding to

[2] Access http://goo.gl/weHaHI to see the test and listen to the sounds.

Table 2. Results of the subjective evaluation.

Target	Similarity	Diversity	Target	Similarity	Diversity
Horn	2.6±0.6	3.4±0.9	Didger	2.1±0.3	3.7±1.1
Synth	1.8±0.3	3.9±1.1	Eleph	2.3±0.2	3.3±1.1
Tbreed	2.0±0.3	3.3±1.3	Frog	1.2±0.1	3.4±1.7
Ahh	2.2±0.5	3.6±1.1	Scream	2.7±0.2	2.7±1.2
Harp	2.3±0.2	3.3±1.0	Gong	2.3±0.3	3.4±1.0

orchestrations that are more similar to the target. The value 1 corresponds to the option *very different* in the listening test and the value 5 corresponds to the option *very similar*. Subjective diversity also varies from 1 to 5 with higher values indicating a more diverse set of orchestrations. The value 1 corresponds to the option *very similar* in the listening test and the value 5 corresponds to the option *very different*.

Subjective Similarity. The target sound *Scream* received the highest score for subjective similarity, followed by *Horn*. Table 1 indicates that *Scream* presents high temporal variation and low harmonicity, while *Horn* presents both low temporal variation and harmonicity. We expected sounds with high harmonicity and low temporal variation such as *Synth*, *Ahh*, and *Didger* to render orchestrations that would be considered more similar than the others because the sounds in the database are stable notes from harmonic musical instruments.

The target sound *Frog* received the lowest score for subjective similarity. The croaking of a frog is characterized mostly by the temporal modulations than spectral features, thus we expected *Frog* to be a particularly challenging target sound to orchestrate with sustained notes from musical instruments.

Subjective Diversity. Most subjective diversity scores were above 3 on average (with the exception of *Scream*), corresponding to assessments between *Fairly Similar* and *Different*. This result is an indication that the AIS is capable of returning multiple orchestrations in parallel that correspond to alternative combinations of sounds.

5.2 Objective Evaluation

The evaluation of objective similarity uses the fitness values of the orchestrations while the evaluation of objective diversity uses Eq. (16). Table 3 shows the results for the objective evaluation. Fitness is always negative and the closer to zero the smaller the distance from the target. The objective diversity values vary from 0 to 1 and the higher they are the more diverse the set of orchestrations because they have fewer sounds in common.

Table 3. Objective evaluation. The table shows the fitness (Fit) and Diversity (Div) for the artificial immune system (AIS) and genetic algorithm (GA).

Target	Fit AIS (x10^{-3})	Fit GA (x10^{-3})	Div AIS	Div GA
Horn	−24.3±3.5	−31.7±3.6	0.78±0.16	0.82±0.20
Synth	−14.8±3.4	−22.5±9.3	0.84±0.13	0.83±0.15
Tbreed	−53.0±5.0	−66.8±5.8	0.82±0.15	0.79±0.16
Ahh	−45.6±3.6	−54.0±3.9	0.84±0.14	0.78±0.17
Harp	−23.9±4.9	−35.7±7.6	0.80±0.15	0.78±0.18
Didger	−16.5±3.1	−18.5±1.8	0.64±0.16	0.62±0.18
Eleph	−64.5±4.1	−74.3±7.4	0.86±0.13	0.80±0.17
Frog	−154.3±7.8	−166.8±13.5	0.78±0.15	0.76±0.19
Scream	−58.6±7.4	−55.0±4.2	0.85±0.14	0.64±0.16
Gong	−50.3±5.8	−65.9±12.0	0.82±0.15	0.72±0.14

Objective Similarity. We compare the results obtained by the AIS opt-aiNet with the results obtained using a standard GA. We ran the GA 10 times and stored the fitness value of the best classified solution after each run. The results shown in Table 3 are the average fitness values obtained in the 10 executions of the GA and the average fitness values obtained in a single run of the AIS. The AIS returns multiple solutions, so we averaged the values of the top 10 orchestrations ranked by fitness value. In general, Table 3 shows that the fitness for the AIS is closer to zero than the fitness for the GA, indicating better matches. However, the results of the objective similarity evaluation in Table 3 do not reflect the subjective similarity from Table 2. For example, the orchestrations that have the best fitness values are *Synth* and *Didger* but these target sounds were not considered to render the closest orchestrations. Therefore, the perceptual significance of the fitness function remains to be investigated.

Objective Diversity. The objective diversity was computed using Eq. (16). The diversity for the AIS was calculated from the top 10 ranked solutions from a single run, while the diversity for the GA was calculated after 10 runs. Table 3 indicates that a single run of the AIS returns a set of orchestrations with objective diversity comparable with multiple runs of the GA.

6 Conclusions and Perspectives

We proposed to use an artificial immune system (AIS) to find combinations of sounds from a database that approach a target sound. The AIS is capable of finding multiple combinations that are good candidate solutions while preserving diversity, contrary to the standard GA. We evaluated the objective and subjective similarity of the orchestrations returned by the AIS to 10 target sounds and

compared the results with a standard GA. Similarly, we evaluated the objective and subjective diversity of 10 solutions found by the AIS and compared with 10 independent runs of the GA. The orchestrations found by the AIS presented similarity and diversity comparable to running the GA multiple times to obtain different orchestrations.

We focused on spectral features of sounds, neglecting the inherent temporal aspect of sound perception. The attack time is the most salient feature in dissimilarity studies and should be considered when searching for orchestrations. Orchestrating percussive sounds such as the *Gong* with sustained musical instruments seems less intuitive than using percussive sounds such as piano notes or plucked violin strings. The temporal evolution of the features was not considered in this work. Sounds that vary in time are expected to pose a greater challenge to orchestrate using notes of musical instrument sounds. Most musical instruments from an orchestra can be played with temporal variations, such as *glissando* or *vibrato*. Thus it seems natural to use target sounds that vary in time.

Future work should investigate how to incorporate time in the search for orchestrations to find better combinations for target sounds that present temporal evolution. The attack time is an important feature to distinguish percussive from sustained sounds. Target sounds with a high degree of temporal variation such as the *Elephant* require a fitness function that encodes temporal variations of the features. Finally, the assumption that orchestrations that are more similar to the target sound are aesthetically better could be investigated.

The results obtained in this work were made available online in a dedicated webpage[3].

Acknowledgments. This work is financed by the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project "UID/EEA/50014/2013." The authors would like to thank the integrated masters program in Electrical and Computer Engineering (MIEEC) from the University of Porto (FEUP) for the financial support.

References

1. Caclin, A., McAdams, S., Smith, B., Winsberg, S.: Acoustic correlates of timbre space dimensions: a confirmatory study using synthetic tones. J. Acoust. Soc. Am. **118**(1), 471–482 (2005)
2. Camacho, A., Harris, J.: A sawtooth waveform inspired pitch estimator for speech and music. J. Acoust. Soc. Am. **124**(3), 1638–1652 (2008)
3. 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)
4. Carpentier, G., Tardieu, D., Assayag, G., Rodet, X., Saint-James, E.: An evolutionary approach to computer-aided orchestration. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 488–497. Springer, Heidelberg (2007)

[3] Access http://goo.gl/4l9NqX to listen to the target sounds and results.

5. Carpentier, G., Assayag, G., Saint-James, E.: Solving the musical orchestration problem using multiobjective constrained optimization with a genetic local search approach. J. Heuristics **16**(5), 681–714 (2010)
6. Carpentier, G., Tardieu, D., Harvey, J., Assayag, G., Saint-James, E.: Predicting timbre features of instrument sound combinations: application to automatic orchestration. J. New Music Res. **39**(1), 47–61 (2010)
7. de Castro, L., Timmis, J.: An artificial immune network for multimodal function optimization. In: CEC 2002, Proceedings of the 2002 Congress on Evolutionary Computation, vol. 1, pp. 699–704, May 2002
8. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC music database: popular, classical and Jazz music databases. In: Proceedings of the International Society for Music Information Retrieval Conference, vol. 2, pp. 287–288 (2002)
9. Grey, J.: Multidimensional perceptual scaling of musical timbres. J. Acoust. Soc. Am. **61**(5), 1270–1277 (1977)
10. 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)
11. 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)
12. Kendall, R.A., Carterette, E.C.: Identification and blend of timbres as a basis for orchestration. Contemp. Music Rev. **9**(1–2), 51–67 (1993)
13. Krumhansl, C.L.: Why is musical timbre so hard to understand? Struct. Percept. Electroacoust. Sound Music **9**, 43–53 (1989)
14. Lartillot, O., Toiviainen, P.: A matlab toolbox for musical feature extraction from audio. In: International Conference on Digital Audio Effects, pp. 237–244 (2007)
15. McAdams, S., Giordano, B.L.: The perception of musical timbre. In: Hallam, S., Cross, I., Thaut, M. (eds.) The Oxford Handbook of Music Psychology, pp. 72–80. Oxford University Press, New York (2009)
16. McAdams, S., Winsberg, S., Donnadieu, S., De Soete, G., Krimphoff, J.: Perceptual scaling of synthesized musical timbres: common dimensions, specificities, and latent subject classes. Psychol. Res. **58**(3), 177–192 (1995)
17. 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)
18. Rose, F., Hetrik, J.E.: Enhancing orchestration technique via spectrally based linear algebra methods. Comput. Music J. **33**(1), 32–41 (2009)
19. Tardieu, D., Rodet, X.: An instrument timbre model for computer aided orchestration. In: 2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 347–350. IEEE (2007)