



SonOpt: understanding the behaviour of bi-objective population-based optimisation algorithms through sound

Tasos Asonitis¹ · Richard Allmendinger² · Matt Benatan³ · Ricardo Climent¹

Received: 15 June 2022 / Revised: 8 February 2023 / Accepted: 10 February 2023 /
Published online: 13 March 2023
© The Author(s) 2023

Abstract

We present an extension of SonOpt, the first ever openly available tool for the sonification of bi-objective population-based optimisation algorithms. SonOpt has already introduced benefits on the understanding of algorithmic behaviour by proposing the use of sound as a medium for the process monitoring of bi-objective optimisation algorithms. The first edition of SonOpt utilised two different sonification paths to provide information on convergence, population diversity, recurrence of objective values across consecutive generations and the shape of the approximation set. The present extension provides further insight through the introduction of a third sonification path, which involves hypervolume contributions to facilitate the understanding of the relative importance of non-dominated solutions. Using a different sound generation approach than the existing ones, this newly proposed sonification path utilizes pitch deviations to highlight the distribution of hypervolume contributions across the approximation set. To demonstrate the benefits of SonOpt we compare the sonic results obtained from two popular population-based multi-objective optimisation algorithms, Non-Dominated Sorting Genetic Algorithm (NSGA-II) and Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), and use a Multi-objective Random Search (MRS) approach as a baseline. The three algorithms are applied to numerous test problems and showcase how sonification can reveal various aspects of the optimisation process that may not be obvious from visualisation alone. SonOpt is available for download at <https://github.com/tasos-a/SonOpt-2.0>.

Keywords SonOpt · Sonification · Multi-objective optimisation · Population-based optimisation algorithms · Algorithm behaviour · Hypervolume · Sound

✉ Tasos Asonitis
anastasios.asonitis@manchester.ac.uk

Extended author information available on the last page of the article

1 Introduction

The use of sound as a data representation and communication medium, commonly referred to as *data sonification* [6], has been applied in various process monitoring fields. From medical applications like heart rate monitors, to everyday tools like parking sensors, sonification has proven its benefits both as an exclusive method for data communication and as an addition to visual or other sensory inputs [5, 26]. Sonification has also served as a meeting point between the scientific and the artistic community. Numerous artistic projects have focused on the exploration of data through sound, resulting in interesting mergers of informative and aesthetic outcomes [9]. Although the advantages of incorporating sonification in process monitoring tasks have been documented [55], sonification has yet to receive the necessary attention in the multi-objective optimisation research field [16, 39]; a field that is concerned with optimisation problems that possess multiple conflicting objectives and thus potentially complex underlying trade-offs between objectives and advanced algorithms to discover and understand these. This motivated us to conceive and design SonOpt [4], an application that aims to facilitate the understanding of optimisation algorithmic behaviour by converting various aspects of the multi-objective optimisation progress to sound.

The multi-objective optimisation community primarily relies on visualisation techniques for evaluating and understanding the algorithmic performance and behaviour [22, 54]. Although insightful, visualisation methods can come short in specific scenarios. For example, scatter plots are a popular visualisation choice but for problems with four or more objectives they would need to be used in conjunction with a dimensionality-reduction method in order to achieve a mapping to a visualisable 2D or 3D space, leading to a loss of Pareto dominance relation information between solutions [54]. An alternative is to use pairwise plots but this would increase the number of required plots rapidly as the number of objectives increases. More advanced visualisation techniques, such as heatmaps and parallel coordinate plots, can provide useful insights into algorithm behaviour but also these techniques become less intuitive as the number of objectives increases. Visualisable many-objective test problems [21, 22, 30, 31] try to circumvent the visualisation issue by sticking to a 2D design space but this class of problems are limited to distance-based problems. Furthermore, visualisation techniques can pose difficulties to the inclusivity of the optimisation community as they do not take into account individuals with impaired vision [1, 38]. The evolution of the non-dominated set of solutions and their corresponding hypervolume contributions through a number of generations can be treated as a time-series. Sawe [48] mentions how visualisation can be less effective in such processes. In cases like these, sonification can suggest an effective alternative [15], or even an important addition to current visualisation methodology. An advantage offered by sonification is the ability of sound to bring forward and highlight events that take place over the course of a single run of an optimisation algorithm [23]. This allows for a low level monitoring of the algorithmic performance that can be combined with the higher level visual representations which tend to focus on performance assessments over multiple algorithmic runs.

SonOpt was created to address the above mentioned issues. The concept of using sound as a way to monitor the optimisation progress is not new, however, the documented applications deal exclusively with single objective tasks [23, 35, 53]. SonOpt is the first attempt to apply sonification in the multi-objective optimisation domain. Although currently the application is limited to bi-objective problems, the core motivation behind SonOpt is to introduce a sonification methodology that can be generalized to scenarios with more objectives. Ultimately, the goal is to create a tool that employs sound to extend and enhance the monitoring capabilities of existing visualisation approaches. For this reason, SonOpt does not aim to substitute visualisation methods for sonification equivalents, but to encourage the multi-modal monitoring of the optimisation process and suggest an alternative perspective on algorithmic behaviour, a perspective in terms of sound. A better understanding of the algorithmic behaviour leads to the improvement of current algorithms and to the design of new, more effective ones. This way SonOpt can contribute to the benefitting of industries that heavily rely on optimisation algorithms for providing solutions to important problems. We believe that SonOpt can also be useful as a teaching asset for the introduction of students to multi-objective optimisation and to the way various algorithms behave during the run. On an artistic note, SonOpt can also function as a creative tool which will encourage composers working with sound to experiment with the sonic potential of optimisation processes.

SonOpt is a real-time application that receives the objective function values of non-dominated solutions, also called approximation set, on each generation of the algorithmic run. The approximation set is sent over to SonOpt in the form of a 2D matrix. By following a workflow that will be covered extensively on following section, SonOpt uses these values to generate two distinct sounds, referred to as *sonification path 1* and *sonification path 2* respectively. The first path aims to provide information on the shape of the approximation set at each generation. It does so by transferring the shape of the approximation set at each generation to the shape of a waveform that is used as a look-up table in a wavetable synthesis engine [28]. The second path offers an insight to the recurrence of points in the approximation set across consecutive generations, by raising the amplitude of harmonic partials on a fundamental frequency depending on the amount and location of the recurrence. In this paper we introduce an extension on the existing capabilities of SonOpt. Specifically, along with the approximation set values, SonOpt's current edition also receives a list with the hypervolume contributions [10, 20] of each of the non-dominated solutions. Subsequently, *sonification path 3* is created, this time informing the listener on how the hypervolume contributions are distributed across the approximation set at each generation, by employing consecutive pitch deviations on a fixed frequency. With this extension we aim to increase the utility of SonOpt by offering the user the opportunity not only to monitor in real-time the position and recurrence of the non-dominated solutions but to further understand their relative importance and how this evolves through the optimisation process. We chose to focus on hypervolume contributions because of its reliance on the hypervolume indicator, which is a popular quality indicator to measure the performance of multi-objective optimisers as it provides information about diversity and convergence of non-dominated solutions. This indicator is also often used to drive indicator-based optimisers and

is Pareto compliant [60]. Note, while the hypervolume indicator tells us something about the quality of an entire set of non-dominated solutions, the hypervolume contribution of a non-dominated solution tells us about the contribution of that solution to the hypervolume of the set (a formal definition is provided in the next section).

It is important to mention that SonOpt does not operate on single metrics that apply to the entirety of the approximation set. Instead, it targets information that relates to each of the non-dominated solutions separately - its position in the objective space and its hypervolume contribution. In order to visually condense this information for every generation, complicated, or animated graphs would be required. In this case, sonification proposes a meaningful alternative since sound can convey this amount of information effectively [48].

In the following section (Sect. 2) we provide formal definition for the terms that appear extensively throughout the text both in relation to multi-objective optimisation and sound. Subsequently, Sect. 3 puts this work in context with existing research, and in Sect. 4 we introduce the workflow and sound generation mechanisms of SonOpt. Following on, Sect. 5 provides details of the experimental setup, and Sect. 6 showcases the effectiveness of SonOpt for bi-objective optimisation problems of varying complexity. Section 7 discusses the potential of SonOpt as a creative tool that can be used in musical composition. Finally, Sect. 8 concludes the paper and details areas of future research.

2 Preliminaries

The following conventions will be used throughout the paper.

2.1 Multi-objective optimisation

Definition 1 (Multi-Objective Optimisation (MOO) Problem). A MOO problem can be defined generally as “minimize” $f(x)$ subject to $x \in X$, where x is an n -dimensional candidate solution vector (or solution), $X \subset \mathbb{R}^n$ is the search domain, and $f = (f_1, \dots, f_m)$ is a vector objective function $f : X \rightarrow Y$ mapping solutions to a m -dimensional objective space $Y \subset \mathbb{R}^m$. The term “minimize” is written in quotes to indicate that in general there is no single solution that minimizes all objectives simultaneously, and a further definition is needed to define an ordering on candidate solutions (see below).

Definition 2 (Pareto Dominance). Consider two solutions x_1 and x_2 . We say that x_1 dominates x_2 , also written as $x_1 < x_2$, if and only if there is at least one i such that $f_i(x_1) < f_i(x_2)$ and for all j , $f_j(x_1) \leq f_j(x_2)$. This relation is also sometimes called strict Pareto dominance in contrast to the weak Pareto dominance defined below.

Definition 3 (Weak Pareto Dominance). Consider two solutions x_1 and x_2 , we say that x_1 is weakly dominated by x_2 , also written as $x_1 \leq x_2$, if and only if $\forall j, f_j(x_1) \leq f_j(x_2)$.

Definition 4 (Pareto Optimal). A solution x_1 is called Pareto optimal if there does not exist a solution x_2 that dominates it.

Definition 5 (Pareto Set). The set of all Pareto optimal solutions is said to form the Pareto set.

Definition 6 (Pareto Front). The image of the Pareto set in the objective space Y is known as the Pareto front.

Definition 7 (Approximation Sets and Performance). Any set of points in the objective space with elements that are all non-dominated within the set is called an approximation set [60], denoted here by A . Such sets can be partially ordered according to the better relation, analogously to the dominance order on points. We refer to the shape created by points in A in the objective space as the approximation set shape. Later in the paper (relating to *sonification path 2*), we will be distinguishing between approximation sets at different generations; where this is the case, we will be using the notation A_t , where t refers to the generation counter.

The aim of multi-objective optimisation can be defined as finding the best possible approximation set A , where best is determined by this order. As a proxy method for assessing approximation sets, the hypervolume indicator [61] (see below) has been recommended. We use the concept of hypervolume contributions [20] to determine the contribution of a individual point of the approximation set to the hypervolume of the entire set.

Definition 8 (Hypervolume Indicator). Given a set $P \in Y$, the hypervolume indicator S of P is defined as the Lebesgue measure of the subspace in Y dominated by P and a user-defined reference point r is defined as [58] $S(P) = \text{Vol}(\cup_{p \in P} [p, r])$, where Vol is the Lebesgue measure on \mathbb{R}^m , and the reference point r selected such that it is dominated by all points in S .

Definition 9 (Hypervolume Contribution). Given a point $p \in P$, its hypervolume contribution with respect to P is $\Delta S(P, p) = S(P) - S(P \setminus p)$.

2.2 Sound

Definition 10 (Sonification). The use of non-verbal audio to represent data is called sonification [6].

Definition 11 (Audification). The process of converting the array of data samples directly into audio samples is known as audification [32].

Definition 12 (Audio Frequency). The frequency of a periodic vibration that falls within the audible range by humans is referred to as audio frequency [41].

Definition 13 (Wavetable Synthesis). The periodic reproduction of a lookup table that includes samples from a dynamically changing single-cycle waveform, is commonly known as wavetable synthesis [28]. In this work, we refer to the frequency of this periodic reproduction as *wavetable oscillator frequency*.

Definition 14 (Additive Synthesis). Harmonic additive synthesis, or generally, additive synthesis, in audio refers to the concept of synthesising spectra by adding sine waves with frequencies equal to integer multiples of a common fundamental frequency. These sine waves are called harmonic partials [50].

Definition 15 (Frequency Modulation Synthesis). The sound synthesis method that involves the production of spectra through the modulation of the frequency of a carrier wave according to a second, modulating, wave, in such a way that the frequency of the modulator determines the rate of change on the frequency of the carrier, is called Frequency Modulation (FM) synthesis. In this work, we refer to the parameter that controls the carrier wave frequency as *base frequency* [12].

3 Background

Research around sonifying optimisation processes is limited, perhaps due to the need to combine expertise from several domains, such as music, operations research, software engineering, and audio signal processing. The purpose of this literature review is to provide an overview of existing research at the intersection of these areas. The interested reader is referred to domain-specific literature to gain a deeper understanding about a specific domain.

This paper is an extension of our previous, initial work on SonOpt [4]. This involved the motivation behind sonification of Multi-Objective Evolutionary Algorithms (MOEAs) and the implementation of SonOpt as a first attempt to achieve a useful sonification. We carried out an experimental study featuring NSGA-II and MOEA/D applied to three popular bi-objective optimisation problems, and the documentation of the responsiveness of the system in these scenarios. To our knowledge, this was the first application of sonification in optimisation tasks with more than one objective. Our previous work included only two sonification paths and, as already mentioned, does not involve Multi-objective Random Search (MRS) as a baseline.

Grond et al. [23] were the first who pointed at the benefits of sonification in single-objective optimisation by placing it within the wider context of process monitoring applications. By applying various levels of modification to the sounds of a mechanical keyboard, they aimed to convey information on the optimisation run of Evolutionary Strategies (ES), a popular type of population-based heuristic (originally) designed for continuous search spaces [46]. The authors motivated the intervention of the user to the optimisation run, based on the information received through the auditory display.

In [53], Tavares and Godoy designed a system that sonifies various characteristics of the population behaviour in an optimisation task. Their focus is on

Particle Swarm Optimisation (PSO) [7, 29]. The authors looked at specific performance metrics including swarm velocity, alignment, diversity and fitness evolution, and designed a system that uses musical parameters, such as notes, tempo, dynamics and timbral qualities, to create sonic representations of these metrics. The authors mention that their approach can be generalised to other population-based heuristics. An important aspect of this work is that it attempts to combine the informative display with a musically appealing outcome. This is reflected on the prioritising of musical qualities in the sonic mapping. The work is applied to single-objective tasks exclusively.

Another documented approach in sonification of optimisation processes is by Lutton et al. [35]. The work is focused on island-based systems, a method of splitting the computational burden of optimisation to more than one system in order to enhance the speed and efficiency of the process. Sections of a musical score or a popular song in audio format are distributed among the computational nodes. Depending on the optimisation progress of each node, the corresponding section might be reproduced faithfully or not. This way the listener can monitor the individual contribution of each node to the overall optimisation progress. Some musical training might be necessary in order to make the most out of the proposed system.

Arguably, the guide of Hermann et al. [24] remains the most comprehensive guide and introduction to auditory display and its various applications. Albeit not specifically related to optimisation, the guide provides a detailed account of sonification including theory, technology, applications and benefits. Vickers [55] discusses explicitly how sonification can assist with process monitoring and specifically highlights the application of sound as a facilitator for the understanding of algorithmic processes. The author lists applications which used audio to help the developers debug run-time processes in simple programs. The auditory display in most of these cases included sound notifications, which were triggered by important events during the algorithmic operation. The majority of the monitored processes mentioned in the text relate to older programming methods and do not include optimisation applications. Vickers highlights another important aspect of sonification relating to peripheral process monitoring. This refers to how the auditory display allows the user to maintain focus on a primary, perhaps visual, task, while aurally monitoring a secondary task. As stated in the previous section, SonOpt is designed to encourage multi-modal process monitoring by providing an insightful addition to existing visualisations. This way the user can combine the comprehensive overview of the algorithmic process provided by visual graphs, with the real-time low level detail, offered by SonOpt.

Research has shown that in process monitoring tasks, multi-modality achieves better results. Hildebrandt et al. [25] showed through experimentation that increased performance was achieved when individuals were provided simultaneously with continuous audio streams and visual cues of the monitored process. In [13, 44] it was shown that experts from various fields tend to choose the coexistence of continuous aural display and visualisation in process monitoring tasks, over single-modal monitoring involving hearing or vision only. Axon et al. [5] were able to show the benefits of combining visual and sonic displays in monitoring tasks for

security purposes. Providing further evidence in a different field, [26] points towards improved performance when using sonification to monitor physical production systems.

Search Trajectory Networks (STNs) by Ochoa et al. [40] focus on visualisations that aim to map the collective trajectories of metaheuristics on various problems. Albeit the work is not related to sonification, the motivation behind STNs and SonOpt is similar. In [34] STNs were expanded to problems with two and three objectives, to facilitate the understanding of search dynamics into the multi-objective domain. Unlike SonOpt, which focuses on a single algorithmic run, STNs aim to provide information by looking at multiple runs of the algorithm.

Schuller et al. [49] present data sonification as an accessible approach that can assist with the explainability of Artificial Intelligence. Lyu et al. [36] apply an interactive virtual environment that allows real-time tweaking of a neural network's hyperparameters by providing users with auditory feedback on the impact of their choices. In [1, 38], sonification is proposed as a more inclusive approach to data representation for individuals who might face difficulties with visualisations. Such examples are individuals with lack of technical expertise or vision related problems.

Experimental studies [42, 43] have discussed the extent to which human ear is sensitive to the harmonic content of a periodic sound. These findings support the sonification methodologies implemented in sonification paths 1 and 2 which rely on the presence of harmonics on a periodic waveform in order to provide a concise representation of the algorithmic behaviour during the optimisation run (for a more detailed explanation see Sects. 4.1 and 4.2). However these methodologies do not require the user to be able to discern each harmonic partial individually but rather recognize the overall change caused in the sonic outcome by the varying presence or absence of harmonics. As such, the use of SonOpt does not require specialized training.

Concluding this section, we should mention that *sonification* is a general term that refers to auditory display and is therefore the choice of preference throughout this text. However, the methods and techniques described here relate heavily to a specific branch of auditory display, which is often described as *audification*. Kramer [32] refers to audification as the most immediate way of sonically representing data by “directly playing back the data samples”. There are cases where audification can be the most fruitful approach for auditory display. In [19], Dombois and Eckel describe these requirements as a sufficiently large dataset consisting of samples that might result in a wave-like signal, the complexity of this signal and whether it presents subtle changes throughout. SonOpt receives two inputs, the approximation set points and their corresponding hypervolume contributions, and both of these incoming datasets meet the above criteria.

4 SonOpt workflow

Figure 1 illustrates SonOpt's workflow, which we explain in more detail in this section. SonOpt has been created within Max/MSP [45], a popular programming language tailored to coding tasks that involve sound. In order to make SonOpt

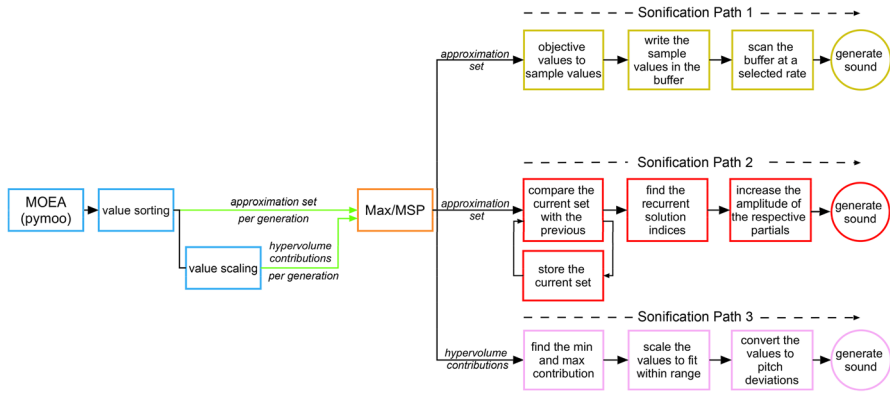


Fig. 1 SonOpt’s workflow. The arrows in green designate the messages sent via OSC from the Python script to Max/MSP (Color figure online)

accessible to individuals that might not be familiar with Max, we introduced an intuitive presentation panel that offers a simplified overview of the parameter controls. The ability to integrate to SonOpt a simple and comprehensive panel that the user can interact with simply by using the mouse, made Max an ideal choice for the development of SonOpt. The panel includes some graphic animation, however, the aim of the visual component is merely to provide a simple insight into the underlying operation of the different sonification paths. Currently SonOpt applies only to bi-objective problems (but its extension to more objectives is part of future research).

SonOpt can work alongside any Python scripting platform, which we use here to carry out the optimisation task (as opposed to converting it into sound, which is done in Max/MSP). For the results presented in this work, we used the pymoo framework [8] to run multi-objective optimisers on the various test problems, and it is that optimisation process that we are sonifying. Pymoo was chosen because of its flexible modular design and its wide application within the multi-objective optimisation community; pymoo is also open-source. However, the users do not have to use a specific Python library and can implement their own algorithms and problems. SonOpt communicates with pymoo in real-time via the Open Sound Control (OSC) [56] protocol, using the python-osc library. In order to use the full functionality of SonOpt (i.e. all sonification paths), the user needs to ensure that two inputs are passed on to Max/MSP at each generation of the algorithm¹: The first input is an approximation set A , which is in our case the objective function values (i.e. a 2D matrix) of non-dominated solutions in the population at that generation. The second input is a 1D array (vector) with the hypervolume contributions of each of these non-dominated solutions. After A has been obtained at a given generation within pymoo, the 2D values contained in it need to be sorted. The purpose of this sorting

¹ Note, in a steady state MOEA, a generation may equate to the creation and processing of a single solution.

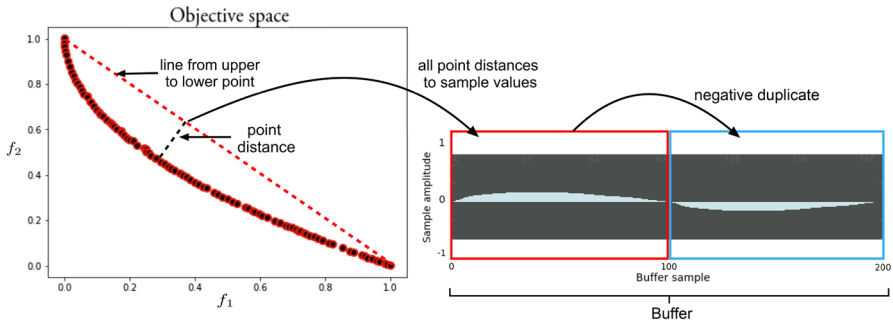


Fig. 2 NSGA-II on ZDT1 via *sonification path 1* at generation 250

is to create a sequence of points that has the exact order as the sequence of points in the objective space graph, reading from left to right (Fig. 2). Once this process has been completed, the first input from pymoo to Max/MSP is ready. For the work presented in this paper, we extended pymoo by adding a short custom loop that calculates the hypervolume contributions of points in A at each generation. To calculate the hypervolume contributions, A needs to be scaled to be within the same range, which is here $[0, 1]$. This scaling is necessary to ensure consistency in the calculation of the hypervolume in different optimisation scenarios. Scaling A to be in $[0, 1]$ allows us to use the same reference point (which will be $[1.2, 1.2]$ here) throughout an optimisation run and for every algorithm and problem combination. In turn, this renders the sonic results generated by SonOpt comparable. After the scaling, the hypervolume contribution of each point in A is calculated. The array of hypervolume contributions follows the order of the points in A as obtained after the sorting. Once the hypervolume contributions are calculated, the second input from pymoo to Max is also ready to be sent. Once the two inputs reach Max/MSP, SonOpt uses the received values to generate three different audio streams. We refer to these streams as *sonification paths* and each of them aims to provide different information about the optimisation process under examination. We will now proceed into examining each of these paths separately.

4.1 Sonification path 1

Sonification path 1 aims to provide information on the evolving shape of the approximation set A during the optimisation process. For this path, SonOpt uses the received 2D matrix. At each generation, SonOpt calculates a straight line that connects the uppermost with the lowermost point of the approximation set A (Fig. 2), which corresponds to the minimum of objective 1 and the minimum of objective 2 respectively. After this, SonOpt calculates the distance of each of the points to this line. These distances are scaled in order to fit within the range $[0, 1]$, and then they are mapped to sample values inside an audio buffer. The resulting waveform has the shape of the approximation set at each generation. Using this technique, negative sample values cannot be obtained. Therefore, in order to generate a bipolar

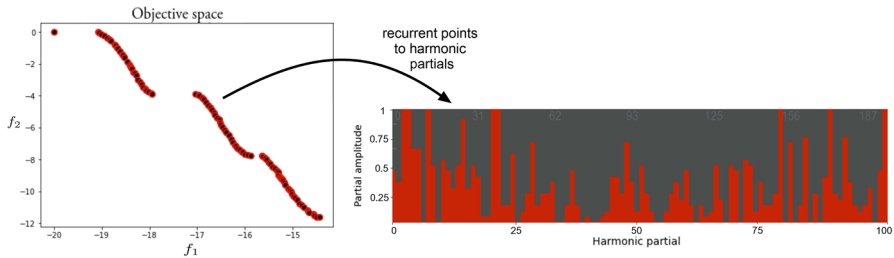


Fig. 3 NSGA-II on Kursawe via *sonification path 2* at generation 250

waveform,² SonOpt duplicates the first half of the buffer and multiplies it with -1 . The buffer size needs to be allocated before the algorithmic run, since Max/MSP does not allow the seamless dynamic resizing of a buffer and should be at least $2N$ samples big, where N is the population size of our MOEA. For sound to be generated, the audio buffer content is treated as a lookup-table and scanned at a user-defined frequency, following a traditional technique known as wavetable synthesis [28]. It should be mentioned that since the number of non-dominated solutions might not be the same at different generations of an algorithmic run, SonOpt only scans the part of the buffer that was generated by the non-dominated solutions of the current generation. This method results in a sound that progressively morphs as a direct result of the gradual change of the approximation set shape and, hence, the optimisation progress.

Through *sonification path 1*, specific characteristics of the approximation set shape are highlighted. A discontinuous shape of A (and thus usually the Pareto front) leads to more jagged waveforms and therefore sounds that are harsh and rich in harmonics. Convex and concave approximation set shapes lead to softer sounds, reminiscent of the sound of a sinewave. Bigger curvature of the approximation set shape results in higher sample values and therefore louder sound.

4.2 Sonification path 2

Similar to path 1, *sonification path 2* receives A , however, the sound generation follows a different approach. Path 2 provides information on the recurrence of non-dominated solutions across consecutive generations, and the location of these solutions across the approximation set. The sound generation technique implemented in this path follows the concept of additive synthesis [50]. The number of harmonic partials is equal to the population size defined in the MOEA. When SonOpt receives A_t , it compares it with the previous approximation set, A_{t-1} . If there are recurrent values between the two, SonOpt finds the positions of these values within the set of the current generation and raises the amplitude of the corresponding harmonic

² Unipolar waveforms tend to put strain on speaker cones, hence why SonOpt works with bipolar waveforms.

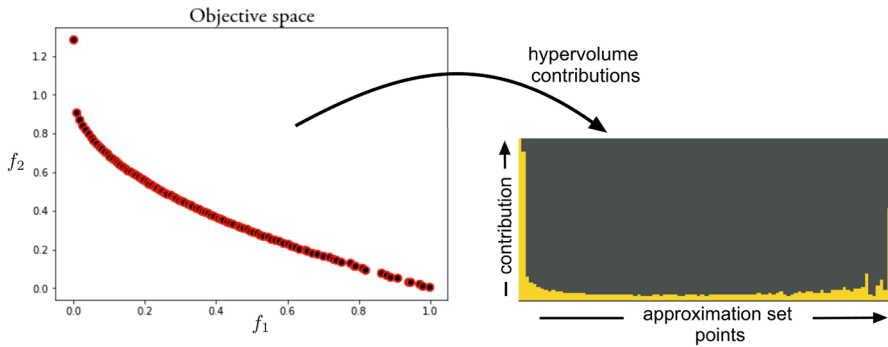


Fig. 4 MOEA/D on ZDT1 via *sonification path 3* at generation 250

partially by a small increment (Fig. 3). The fundamental frequency can be defined by the user and can be changed in real-time as the algorithm runs. It should be noted that *sonification path 2* calculates recurrence taking place across consecutive generations; if a solution occurs for a given number of successive generations, then disappears for a generation, and reappears on the very next one, *sonification path 2* resets its recurrence and starts counting again from the first successive reappearance of the solution.

The harmonic richness of the sound produced by *Sonification path 2* depends on the amount of non-dominated solution recurrence across consecutive generations. More recurrence generates bright, harsh, harmonically rich sounds. On the other hand, less recurring values lead to spectrally “focused”, “soft”, or even individual tones as a result of possibly single sounding harmonic partials. Depending on the location of the recurrent points in A , the produced spectrum can be based on the low, mid or high frequency range. *Sonification path 2* can therefore provide a quick sonic snapshot of where value recurrence takes place in A . For populations of large size, presenting such information visually would require very dense or animated visualisations.

4.3 Sonification path 3

In this paper we introduce for the first time *sonification path 3*. Diverting from the previous two paths, *sonification path 3* receives the hypervolume contributions of each point in A on a per-generation basis. Once received, SonOpt scales the values so that they fit within a range defined by the user. It then proceeds with scanning through the list of (ordered) hypervolume contributions and subsequently, depending on the amount of each contribution, raises the frequency (pitch) of a sinewave, according to the range defined by the user (Fig. 4). The original frequency of this sinewave is also set by the user and can change in real-time as the algorithm runs. Following the method described, SonOpt creates a short sound at each generation of the algorithm. The points in A are treated as time steps: the first points in the set correspond to pitch deviations occurring in the beginning of the sound, while the latter points correspond to the pitch deviations taking place at the middle or end of

it. This is the reason why the hypervolume contributions need to be already sorted in the Python script in order to match the sequence of points in A sent to the other two paths. The sonification approach followed in this path conveys information about the optimisation process through the temporal evolution of sound, while the two previous paths rely on information obtained from the spectral characteristics of the generated sound. It should be mentioned that SonOpt scales the received values according to the minimum and maximum values in the list of hypervolume contributions received at each generation. Thus the produced sound provides information on the relevant levels of the hypervolume contributions of the non-dominated solutions; i.e. how does a solution compares against the minimum and maximum contribution at a given generation. Another parameter that greatly affects the sound of *sonification path 3* is the actual duration it takes for an algorithm to run. If each generation is completed too quickly, the sonic result might be imperceptible. For this reason we suggest that the user slows down the algorithmic run by at least 300 to 500 milliseconds per generation.

The sound produced by *sonification path 3* is affected by the deviations of the per-solution hypervolume contributions within a given generation, as well as how these are distributed across the approximation set. At the beginning of an optimisation run, the algorithms usually explore the objective space resulting in a sound with large pitch deviations. As the algorithm starts to converge, the deviations are gradually reduced and eventually distinctive repetitive rhythmic patterns emerge. Since the scaling is applied on the contributions of each generation separately, the pitch contour, usually forming local pitch “spikes”, depending on how much bigger the biggest contributions are compared to the rest. The location of the spikes across the duration of the sound depends on the location of the most contributing points within A .

4.4 General remarks

The first edition of SonOpt included only *sonification paths 1* and *2*. For this reason, the two paths were designed to evolve in opposite directions during the algorithmic run. *Sonification path 1* is characterized by quick changes and sonic complexity during the initial stage of the optimisation. This is expected since the approximation set shape changes drastically during this phase of the optimisation. Once the approximation set has converged sufficiently, the sound changes in this sonification path are generally subtle. *Sonification path 2* tends to work in the opposite way. At the beginning of the optimisation, it produces simple sounds that tend to evolve into complex and harsh after sufficient convergence has been achieved, since this is when the approximation set points tend to be repeated more through consecutive generations. This way, the information conveyed by the two paths can be distinguished even when both sonification paths are being auditioned at the same time. With the newly introduced addition of *sonification path 3*, we provide the user the opportunity to quickly mute any path as the algorithm runs, as well as control the volume of each path separately, depending on the desired focus of the investigation.

A conscious choice during the design of SonOpt was to keep the sonification methodology away from musical qualities like harmony and melody. SonOpt's approach is based on sound synthesis instead, by directly converting data points into sound. The rationale behind this design choice was to allow SonOpt to be used by individuals that do not have the necessary training to understand the differences amongst musical chords or scales. The followed approach allows the user to focus on the way data samples change through time by attending to the primitive qualities of sound, such as frequency range, loudness vs quietness, and "softness" versus "harshness". Furthermore, using musical intervals arranged in semitones would require discretisation of the input dataset. The reason for this is that the minimum increment of change in the sonic output would necessarily be that of a semitone. We find this a limiting factor when trying to convey aurally subtle changes that may take place in the dataset. Characteristics like the harmonic richness of a sound, or the shape of its waveform can noticeably convey nuanced changes in the dataset.

5 Experimental setup

This section provides details of the experimental setup as used in the experimental study carried out in the subsequent section.

5.1 Algorithm settings

In this paper we employ three popular algorithms: Non-dominated Sorting Genetic Algorithm (NSGA-II) [18], Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [57] and a multi-objective version of random search (MRS). NSGA-II and MOEA/D were chosen because of their popularity in the community and their reliance on different optimisation paradigms (Pareto-dominance vs decomposition). Since SonOpt relies on a population of solutions being processed at each generation, we had to introduce the notion of a population into MRS. To allow for a straightforward comparison of MRS and the other two algorithms, our RS algorithm is in essence a randomized version of NSGA-II: MRS generates every offspring solution at random (as we do traditionally when initializing the population of MOEAs) instead of using parental selection followed by crossover and mutation; environmental selection (non-dominated sorting and crowding distance applied the combined pool of the current and offspring population) is done identically in MRS and NSGA-II. Both algorithms are by default set to eliminate the duplicate solutions of the current or offspring population after the merging, repeating mating until the offspring population includes only unique solutions [8]. This does not affect *sonification path 2*, which calculates the recurrence of unique solutions across consecutive generations (and not the number of identical solutions in a population). The reason to use MRS is it does not perform guided search and thus we would expect it to behave differently from algorithms, such as NSGA-II and MOEA/D. This provides us with the opportunity to test the responsiveness of SonOpt as we expect the sonic outcome to be different from the output of the other two algorithms. The pymoo

Table 1 Algorithm parameter settings as used in the experimental study

Algorithm	Parameter	Setting
NSGA-II	Population size	100
	Population initialization	Random sampling
	Parental selection	Binary tournament with replacement
	Crossover	Simulated binary crossover ($\eta_c = 20$, $p_c = 0.9$)
	Mutation	Polynomial mutation ($\eta_m = 20$, $p_m = 1/n$)
	Eliminate duplicates	Yes
MOEA/D	Population initialization	Random sampling
	Reference vectors	99 (set following [14])
	Size of neighborhood	$T = 20$
	Crossover	Simulated binary crossover ($\eta_c = 20$, $p_c = 0.9$)
	Mutation	Polynomial mutation ($\eta_m = 20$, $p_m = 1/n$)
MRS	Population size	100
	Eliminate duplicates	Yes
All	Number of generations	250

framework was used to execute all three algorithms with MRS being integrated to the framework for the purpose of the study.³

SonOpt is not limited to these algorithms and can be applied to any population-based algorithm. It is important to remind ourselves that the purpose of this work is not to compare algorithmic performance or decide which algorithm is most suitable for a given problem. For this reason, for all of the algorithms, we have used the recommended default parameters of the algorithms (see Table 1 for an overview of algorithm parameters and their settings). The only parameters that required custom setting, were population size for NSGA-II and MRS and number of reference vectors for MOEA/D. The population size was set to 100 and 99 reference vectors were used (NSGA-II typically works with a population size of an even number), for consistency and ease of result demonstration. For every result figure we show in the paper, we used a snapshot at generation 250. The reader is referred to a range of videos available at <https://tinyurl.com/sonopt2> for live demonstrations of SonOpt.

The algorithms were run on a collection of bi-objective problems (see videos <https://tinyurl.com/sonopt2>), however, in this paper we present selected results obtained for ZDT1, ZDT3, ZDT4 [59], Kursawe [33], Tanaka [52] and CTP2 [17]; these problems vary in complexity and Pareto front shapes. Please refer to Appendix B for a formal definition of these problems and visualisations of their Pareto fronts. Note, Tanaka and CTP2 are constrained problems that cannot be tackled by MOEA/D by default in pymoo. All problems used, both in the paper and videos, are implemented by default in pymoo. Some of the results shown here for *sonification path 1* and *2* have been presented in a condensed form in our previous work [4]. We recap and expand on these in this paper.

³ The authors thank Julian Blank, the leading developer of pymoo, for the integration.

Table 2 Audio-related parameters of SonOpt and their settings

Setting	Path 1	Path 2	Path 3	Value range	Impact
Population size	100	100	100	n/a	n/a
Sample value scaling	500	n/a	n/a	0–max	Loudness
Wavetable oscillator frequency (Hertz)	80	n/a	n/a	60–200 (suggested)	Frequency range
Amplitude	0.3	0.15	0.3	0–1	Volume
Stereo position (speaker)	Left	Right	Middle	Left–right	Stereo placement
Fundamental frequency (Hertz)	n/a	80	n/a	60–200 (suggested)	Frequency range
Sleep interval (milliseconds)	n/a	n/a	500	250–1000 (suggested)	Duration
Exponential scaling	n/a	n/a	1	0–1	Pitch contour
Base frequency (Hertz)	n/a	n/a	80	60–200 (suggested)	Frequency range

5.2 SonOpt settings

Table 2 presents an overview of audio-related parameters of SonOpt and their settings as used in the experimental study. The table also shows the possible value ranges of each algorithm parameter and the impact of a parameter on the sonic outcome or working of SonOpt. Having tested different configurations, we recommend these parameters as default for SonOpt because they offer sufficient sonic clarity. Of course, users can define these parameters according to their personal preference. *Population size* is a setting that needs to be transferred from the algorithm settings to SonOpt. It is not a sonification parameter as such, but needs to be defined in SonOpt since it affects a group of parameters that should otherwise be defined individually. *Sample value scaling* refers to the scaling that is applied to the incoming approximation set point coordinates, in order to convert them to sample values.⁴ It only applies to *sonification path 1* and depends on the range of the received values. The closer these values are to 0, the larger the scaling needs to be and vice versa. The highest the scaling value is, the louder the produced sound gets. The maximum limit of this value depends on the incoming values and can be only defined per case. *Wavetable oscillator frequency* defines the buffer scanning rate in *sonification path 1*. This parameter impacts the overall frequency range of the produced sound for *sonification path 1*. A higher frequency leads to higher frequency ranges and vice versa. We suggest that the range between 60 and 200 Hertz is a suitable range that produces aurally perceptible results. *Amplitude* controls the loudness and can be set separately for each path. *Stereo position* needs to be set individually for each path and determines the stereo placement of the sound produced by the respective path (left, right speaker, or anywhere in between). *Fundamental frequency* is the frequency of the fundamental sinewave in *sonification path 2*. Similar to *Wavetable oscillator frequency*, this parameter affects the range of the produced frequency spectrum. *Sleep interval* affects only *sonification path 3* and is transferred from the python script. It

⁴ Only for the Kursawe problem this value was set to 800 to avoid clipping distortion.

defines the pause interval between generations. As explained in Sect. 4.3, a sleep interval needs to be defined between each generation, so that *sonification path 3* can produce perceptible results. The bigger the sleep interval is, the longer the duration of the sound, leading to a more distinguishable sonic outcome. However, because efficiency is taken into account, we have found that 500 milliseconds is a good interval for perceptible results. Since path 3 generates short sound clips, one for each generation, SonOpt ensures there is a short pause after each clip informing the user when a clip ends and the next one begins. The duration of this pause is automatically calculated from SonOpt and is included within the sleep interval defined by the user. *Exponential scaling* applies only to path 3 as well. It is an optional parameter that affects the relative scaling of the hypervolume contributions in order to convert them to pitch deviations. The default value is set to 1, which equals to a linear scaling. However exponential scaling can be useful in cases where the user prefers to sonically accentuate the pitch deviations generated by the hypervolume contributions. This becomes handy if the contributions are very similar to each other, and the pitch deviation patterns are not clearly discernible. The final parameter of *sonification path 3* is *base frequency*. This parameter defines the frequency of the base sine wave, which is subjected to the pitch deviations and affects the frequency range produced through *sonification path 3*.

All audio-related parameters of SonOpt can be changed dynamically during an algorithmic run. However, *population size* is expected to stay fixed during the algorithm run.

5.3 Additional media

SonOpt is a sonification application and as such, generates sound from input data. It becomes clear that the best way to experiment with SonOpt and gain the most out of its utilities, is to use it yourself. In order to be able to discuss the results of using SonOpt in a written context we have provided images of SonOpt's interface as well as mel spectrograms of the sonic output. Mel spectrograms display the distribution of energy across the mel scale, a non-linear scale that corresponds to the perception of pitch by humans [51]. Mel spectrograms were preferred in this text because they can reveal effectively the impact of the distribution of spectral energy to the human hearing, thus highlighting changes in sound that are noticeable to listeners. Additionally, as mentioned previously, we provide a collection of videos with SonOpt being applied to more optimisation scenarios.

6 Experimental study

This section includes a demonstration and analysis of SonOpt applied to several MOEAs and test problems. The analysis itself is done for each of the sonification paths separately to improve clarity. Specifically for MRS, we demonstrate and discuss the produced results only in the first problem category for each sonification path. The reason behind this is that the behaviour of MRS does not vary

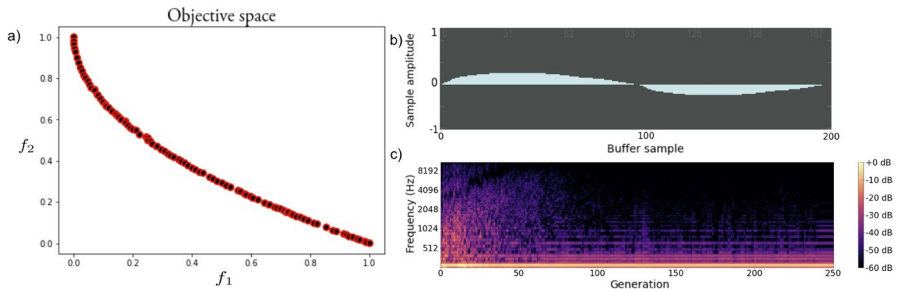


Fig. 5 NSGA-II on ZDT1 via *sonification path 1*: **a** Approximation set obtained after 250 generations; **b** buffer contains on the 250th generation; **c** mel spectrogram of sonification path 1 across 250 generations. Subsequent figures of results will be of same structure in terms of information shown

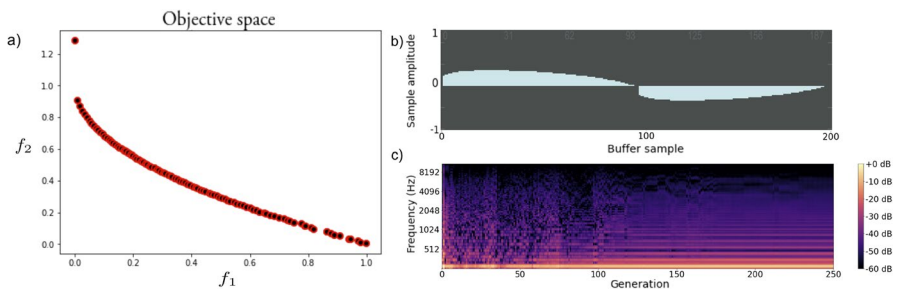


Fig. 6 MOEA/D on ZDT1 via *sonification path 1*

substantially depending on different problems and therefore does not allow room for additional analysis on each problem category. We refer the reader to the Appendix A where the results of MRS on more problem categories can be found.

6.1 Sonification path 1

In this section we demonstrate the responsiveness of *sonification path 1* in various optimisation scenarios. *Sonification path 1* provides information about the shape of the approximation set (i.e. the Pareto front eventually). Imagining a particular sound or a change in sound is difficult from written text alone. Hence, to follow our analysis and reduce ambiguity, we suggest to read through the analysis and watch the corresponding video simultaneously.

6.1.1 Sonification path 1 on a convex bi-objective problem

The first case demonstrates the use of *sonification path 1* on ZDT1. NSGA-II and MOEA/D both managed to converge successfully by the 250th generation. However, the approximation set in NSGA-II achieved convexity quicker than in the case of MOEA/D. This becomes evident by the presence of harmonic content generated by

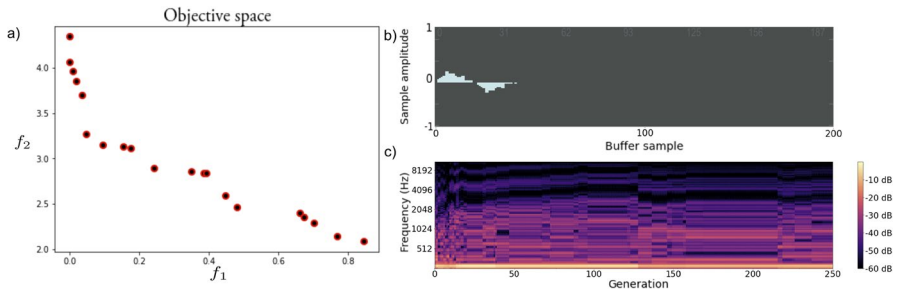


Fig. 7 MRS on ZDT1 via *sonification path 1*

sonification path 1 on each algorithm. For NSGA-II, the harmonic content reduces shortly after the 75th generation, meaning that after that, the approximation set shape has reached sufficient convexity in order to create a corresponding waveform that resembles the shape of a sinewave (Fig. 5); for clarity reasons, all result figures for *sonification path 1* follow the same structure in terms of information shown (approximation set, buffer, and mel spectrogram). This translates to a “softer” sound with less harmonic content in comparison to the sound produced during the first generations. For MOEA/D, the harmonics in the sound persist until the 100th generation (Fig. 6). Even after this point, the sound maintains a noticeable harmonic content, due to the single outlying point on the upper part of the approximation set. This point changes the overall shape of the set sufficiently enough to affect the produced waveform and, therefore, the sonic outcome. It becomes clear that *sonification path 1* is very sensitive to the approximation set shape and can thus convey information on the algorithmic behaviour with sufficient accuracy. MRS did not converge successfully after 250 generations, as can be witnessed from the obtained approximation set shape (Fig. 7). The number of non-dominated solutions produced is smaller in comparison to the other two MOEA’s. This translates to a waveform with fewer samples, hence, a “buzzy”, harmonically rich sound. The harmonics do not cease through the optimisation, making clear that the algorithm did not manage to converge. Furthermore, *sonification path 1* provides information on how smoothly the approximation set shape was obtained. In the case of MRS, the produced sound conveyed changes in a step-like manner; ergo it stays unchanged across a few generations, instead of changing between subsequent generations, as is the case with guided search (at least until convergence occurs). In this case, the output of *sonification path 1*, confirms the expected behaviour of MRS, since the random sampling of solutions in each generation results in a slow optimisation progress.

6.1.2 Sonification path 1 on a convex multi-modal bi-objective problem

When applied to a multi-modal problem like ZDT4, *sonification path 1* can provide useful insights to the optimisation process highlighting the alternation between exploitation and exploration. The sound generated for NSGA-II and MOEA/D presents significant instability through the first 100 generations (Figs. 8 and 9). The

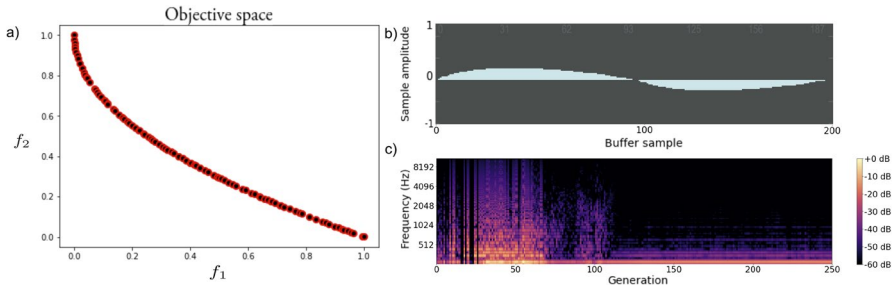


Fig. 8 NSGA-II on ZDT4 via *sonification path 1*

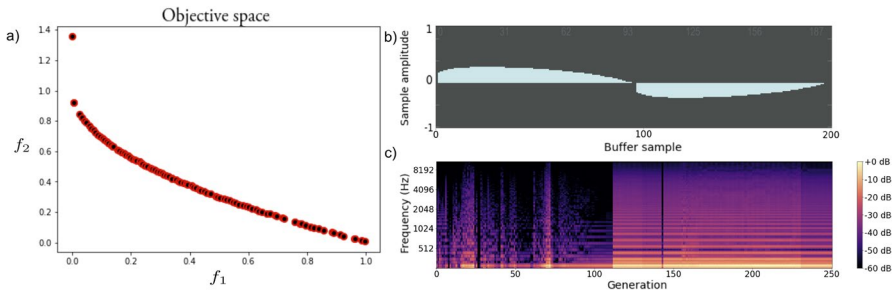


Fig. 9 MOEA/D on ZDT4 via *sonification path 1*

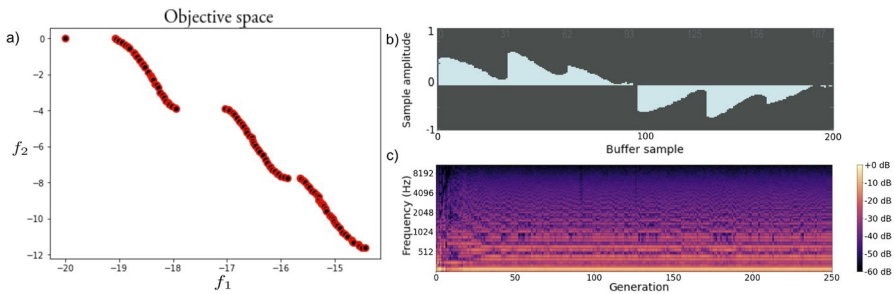


Fig. 10 NSGA-II on Kursawe via *sonification path 1*

sound in both cases includes moments of silence followed by harsh sound “bursts”, while overall characterized by strong harmonic presence. This behaviour is due to the local optima present in the objective space of ZDT4. The approximation set shape is changing drastically during the first half of the algorithm run, as a result of the algorithm getting stuck temporarily at various local optima, and this is directly affecting the outcome of *sonification path 1*. Eventually both algorithms converge, which results to a simpler, sinewave-like tone for NSGA-II and a stable sound for MOEA/D, again affected by the outlier on the upper part of the approximation set.

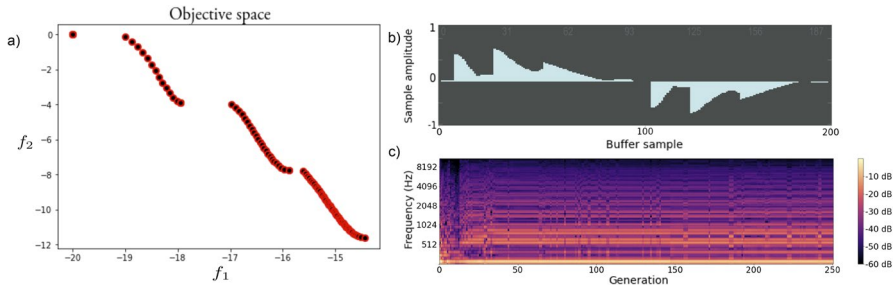


Fig. 11 MOEA/D on Kursawe via *sonification path 1*

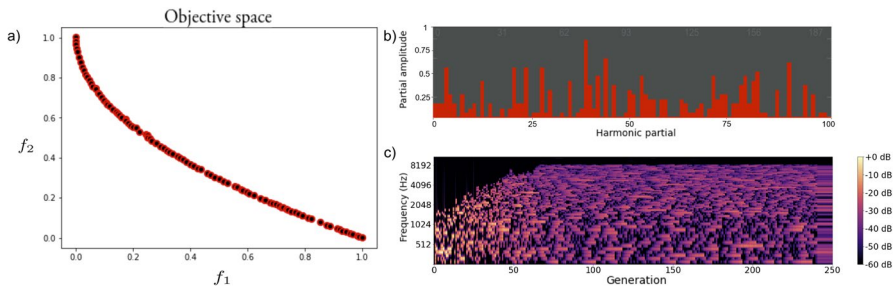


Fig. 12 NSGA-II on ZDT1 via *sonification path 2*. **a** Approximation set obtained after 250 generations; **b** amplitude of the harmonic partials on the 250th generation; **c** mel spectrogram of *sonification path 2* across 250 generations. Subsequent figures of results will be of same structure in terms of information shown

6.1.3 Sonification path 1 on a discontinuous bi-objective problem

In the case of Kursawe, *sonification path 1* reveals useful information on the different optimisation methodologies between NSGA-II and MOEA/D. For NSGA-II the sound evolves smoothly through the optimisation process, since small changes on the approximation set shape take place on each generation (Fig. 10). For MOEA/D on the other hand, the sound evolves in a “blocky” way, changing noticeably only every few generations (Fig. 11). This highlights the different search paradigms, non-dominated sorting versus decomposition, on which the two algorithms are based. Since the Kursawe function presents a discontinuous Pareto front, the sound obtained through *sonification path 1* presents harmonic content even after convergence. Generally, discontinuity on the approximation set shape is translated to edgy waveforms which are rich in harmonics.

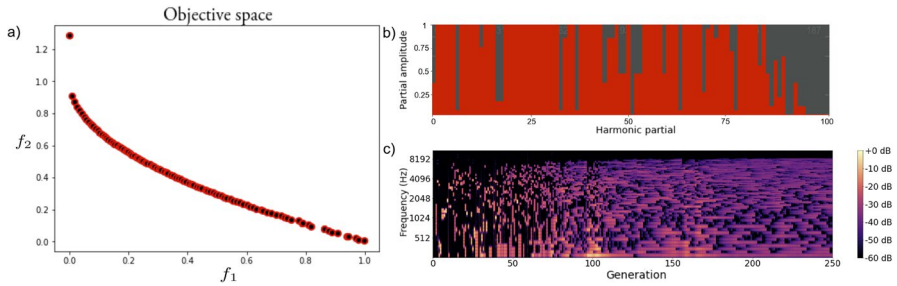


Fig. 13 MOEA/D on ZDT1 via *sonification path 2*

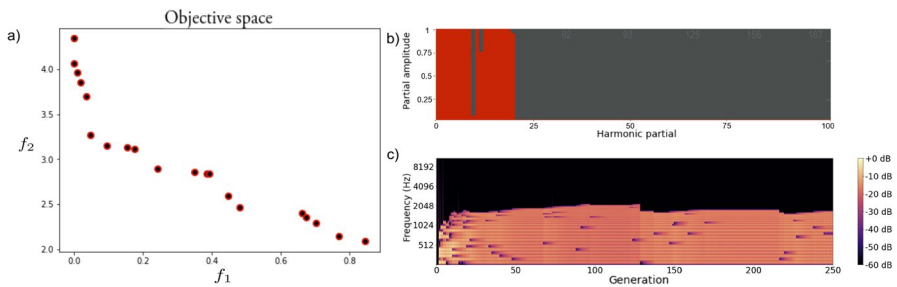


Fig. 14 MRS on ZDT1 via *sonification path 2*

6.2 Sonification path 2

This sonification path informs the user about the level of solution recurrence, and the location of recurrence along the approximation set. Naturally, recurrence is a property that depends on both the way an algorithm works (in particular its diversity-maintenance mechanism) and the properties of the problem being solved.

6.2.1 Sonification path 2 on a convex bi-objective problem

The results of applying *sonification path 2* on ZDT1 for NSGA-II and MOEA/D are presented in Figs. 12 and 13 respectively. It can be observed from the results that in both cases the algorithms generated an excessive number of recurring points in the approximation set across generations. NSGA-II generated a steadily increasing number of recurring solutions during the first 60 generations, as can be heard from the gradually increasing harmonic content of the sound produced through *sonification path 2*. This reflects the fact that it took NSGA-II approximately 60 generations to reach the maximum number of non-dominated solutions (100 in this case). For MOEA/D however, the distribution of the harmonics is more dispersed through the spectrum, meaning that the maximum number of non-dominated solutions was reached earlier than MOEA/D and consequently recurrence was present across the whole approximation set. For the first 50 generations, the recurrence seems to be more “local”, resulting in clearly distinguished individuals harmonic partials, while

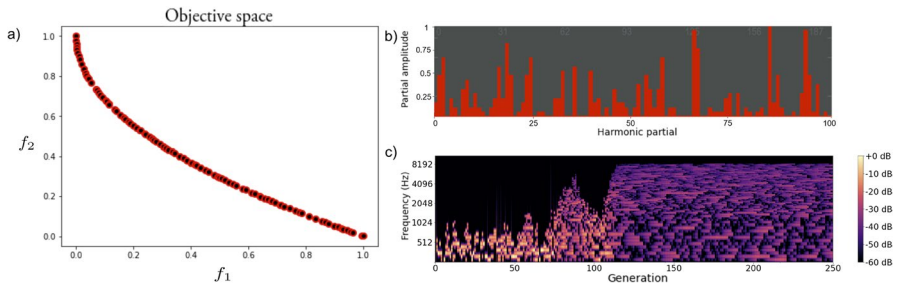


Fig. 15 NSGA-II on ZDT4 via *sonification path 2*

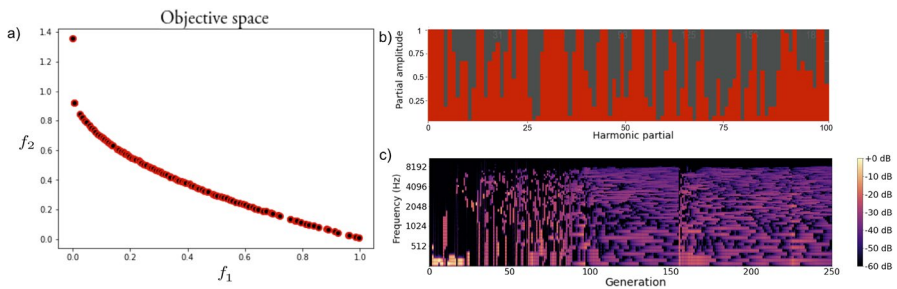


Fig. 16 MOEA/D on ZDT4 via *sonification path 2*

for the rest of the optimisation process recurrence is evenly distributed across the entire approximation set, resulting in a “harsh”, harmonically rich sound. MRS did not approximate the Pareto front sufficiently after 250 generations (Fig. 14). As expected, due to unguided search of MRS, the approximation set does not change drastically as the optimisation progresses resulting in a constant recurrence of solutions. This behaviour is evident from the sonic output of *sonification path 2*, which stays unchanged through the entire process. Another observation we can make from SonOpt’s output is that MRS does not manage to obtain an entire population of non-dominated solutions resulting in a sonic outcome that is spectrally limited to the low and mid-high frequency range, in comparison to NSGA-II and MOEA/D.

6.2.2 Sonification path 2 on a convex multi-modal bi-objective problem

Sonification path 2 sheds light on the objective space of ZDT4. Running NSGA-II on ZDT4 produces a sound that oscillates between the low and mid-high end of the spectrum (Fig. 15). This reflects ways in which the local optima affect the optimisation progress. The longer NSGA-II is being stuck at a local optimum, the greater the solution recurrence during the corresponding generations. As a result, the sound gets “brighter” as recurrence increases. When the algorithm moves away from a local optimum, the recurrence ceases abruptly which in turn results to the sound returning to the low-frequency range. Eventually, after the

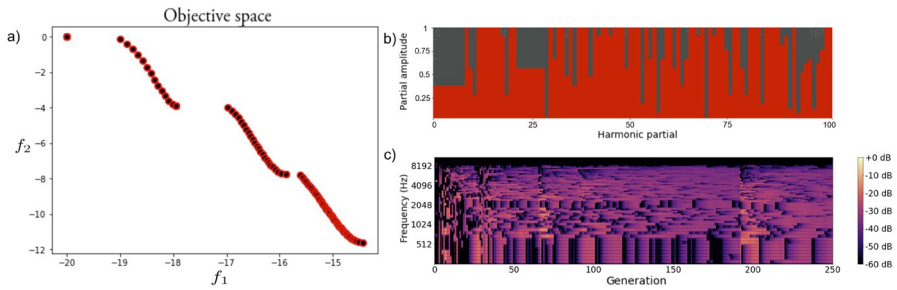


Fig. 17 MOEA/D on Kursawe via *sonification path 2*

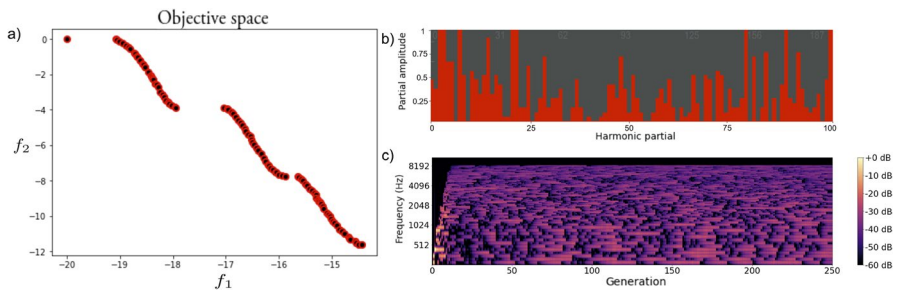


Fig. 18 NSGA-II on Kursawe via *sonification path 2*

100th generation NSGA-II converges, leading to a sound with increased harmonic presence. The transition between local optima can be heard in the application of MOEA/D on ZDT4 through the moments of silence during the first half of the run, designating the complete ceasing of consecutive solution recurrence at specific generations (Fig. 16c).

6.2.3 Sonification path 2 on a discontinuous bi-objective problem

The sound generated from *sonification path 2* in the case of MOEA/D, conveys information on the convergence of the algorithm. During the first 60 generations, the harmonic partials produced are scattered around the spectrum and short in duration (Fig. 17). This reflects the quick and drastic changes taking place across the approximation set during this part of the optimisation process. As the approximation set approaches the Pareto front, the harmonic partials obtain longer duration, resulting in a continuous sonic stream, as an outcome of the non-dominated solutions repeating over more consecutive generations. NSGA-II produces harmonic partials short on duration throughout the whole run (Fig. 18c), meaning that recurrent solutions do not last for many consecutive generations. This may imply that it exploits the objective space more actively (i.e. maintains more diversity) than MOEA/D.

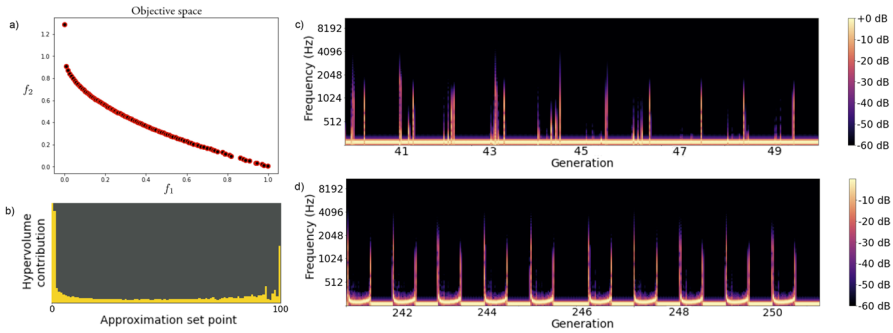


Fig. 19 MOEA/D on ZDT1 via *sonification path 3*. **a** Approximation set obtained after 250 generations; **b** hypervolume contributions on the 250th generation; **c** mel spectrogram of sonification path 3 during generations 40–49; **d** mel spectrogram of *sonification path 3* during generations 241–250

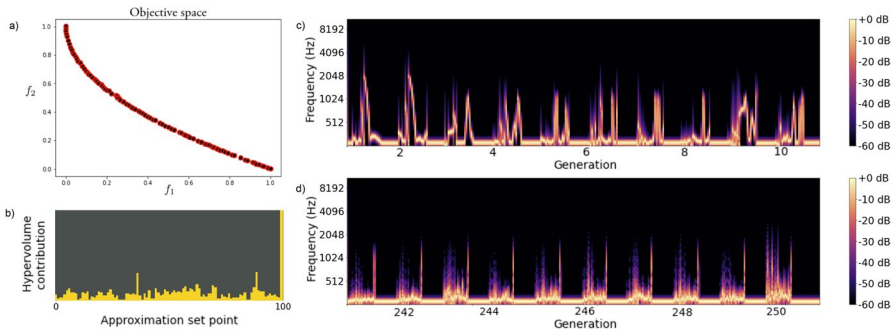


Fig. 20 NSGA-II on ZDT1 via *sonification path 3*. **(a and b)** as above. **c** Mel spectrogram of sonification path 3 during generations 1–10. **d** Mel spectrogram of sonification path 3 during generations 241–250

6.3 Sonification path 3

In this section we present and discuss the use of *sonification path 3*, which provides sonic information to the user about the level of hypervolume contribution of non-dominated solutions in a population, and where these contributions are located along the approximation front.

6.3.1 Sonification path 3 on a convex bi-objective problem

Fig. 19 demonstrates the results of *sonification path 3* when applying MOEA/D, NSGA-II and MRS on ZDT1. As expected, before convergence has been achieved, the hypervolume contribution of each non-dominated solution in the population changes significantly per generation. Once convergence has been achieved, the hypervolume contributions present very small differentiation from generation to generation. This transition is clearly audible through *sonification*

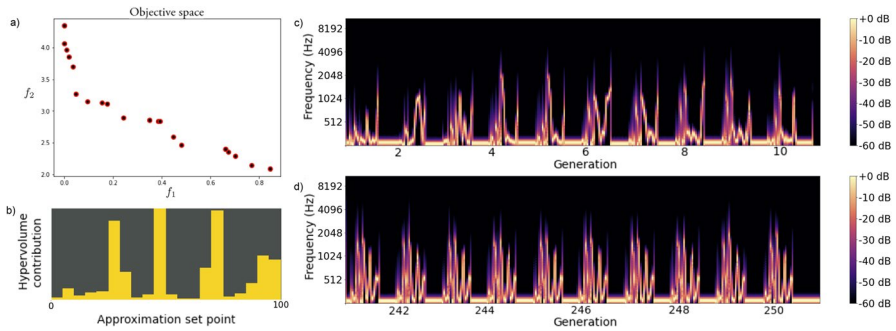


Fig. 21 MRS on ZDT1 via *sonification path 3*. (a and b) as above. c Mel spectrogram of *sonification path 3* during generations 1–10. d Mel spectrogram of *sonification path 3* during generations 241–250

path 3. The sound produced changes dramatically across consecutive generations as MOEA/D and NSGA-II explore the objective space (subfigure c in Figs. 19 and 20 respectively). By the end of the optimisation process, and once the algorithm has converged, the sound generated through *sonification path 3* becomes a repeating pattern, revealing that the hypervolume contributions change by a very small amount, or even not at all, from one generation to the next (subfigure d in Figs. 19 and 20). Since the methodology implemented in *sonification path 3* relies on the temporal transformation of sound, it becomes quickly evident when the algorithm has converged and how the hypervolume contributions are distributed across the approximation set. In the case of MOEA/D on ZDT1, the distribution of the contributions (Fig. 19b), can be clearly heard through the generated sound pattern. This becomes evident even visually by comparing the shape in Fig. 19b with the mel spectrogram of the corresponding sound displayed in Fig. 19d. The smooth convexity presented in Fig. 19b can be heard in the pitch transformation of the base frequency. The resulting sound consists of a smooth pitch “dive” followed by a similarly smooth pitch rise.

On the other hand, the distribution of the hypervolume contributions as obtained by NSGA-II (Fig. 20b) by the 250th generation does not present the convexity of MOEA/D, resulting in a “granular” sound. In this case, *sonification path 3* reveals that, although both NSGA-II and MOEA/D converged successfully, more diversity is maintained by NSGA-II. It also reveals that MOEA/D, because of decomposition, tends to distribute the solutions equidistantly across the approximation set, directly impacting how each solution contributes to the hypervolume. Since MRS converges very slowly, the hypervolume contributions of the produced solutions are not expected to present abrupt changes during the first generations optimisation run, as was the case with MOEA/D and NSGA-II. Indeed, the sonic outcome of this path for MRS changes only gradually during the run, and pattern repetition can be heard from early on, for example during generations 7 to 10 (Fig. 21c).

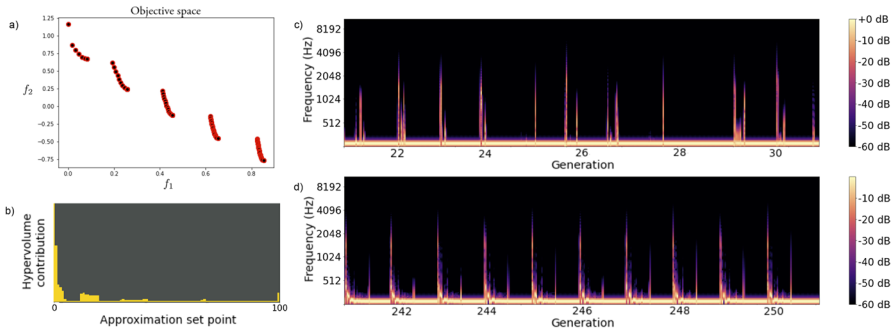


Fig. 22 MOEA/D on ZDT3 via *sonification path 3*. **a** Approximation set obtained after 250 generations; **b** Hypervolume contributions on the 250th generation; **c** Mel spectrogram of sonification path 3 during generations 21–30; **d** Mel spectrogram of *sonification path 3* during generations 241–250

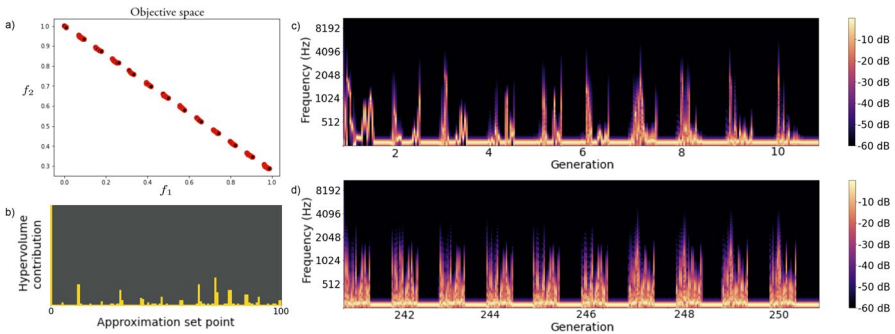


Fig. 23 NSGA-II on CTP2 via *sonification path 3*. **a** and **b** as above. **c** Mel spectrogram of sonification path 3 during generations 1–10. **d** Mel spectrogram of sonification path 3 during generations 241–250

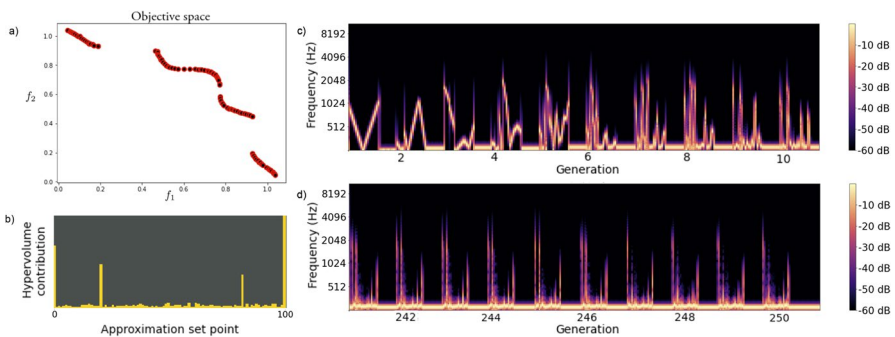


Fig. 24 NSGA-II on Tanaka via *sonification path 3*. **a** and **b** as above. **c** Mel spectrogram of sonification path 3 during generations 1–10. **d** Mel spectrogram of sonification path 3 during generations 241–250

6.3.2 Sonification path 3 on various discontinuous bi-objective problems

When the Pareto front consists of discontinuities, *sonification path 3* can be particularly helpful. The approximation set points located at the breakpoints (ie. the approximation set points after which discontinuity occurs on the objective space) of the objective space tend to contribute the most towards the hypervolume as can be seen in subfigures b of Figs. 22, 23 and 24. Through *sonification path 3* these higher contributions are translated into pitch “spikes”. These spikes are aurally discernible, and therefore not only provide information on the convergence progress, but also on the shape of the approximation set. In this aspect, *sonification path 3* works complimentary to *sonification path 1*, which primarily aims to inform the user about the shape of the approximation set. As with ZDT1, in these examples the sonic outcome transitions from irregularity, as can be seen in subfigures c of Figs. 22, 23 and 24), to pattern repetition, as can be seen in subfigures d of of Figs. 22, 23 and 24).

6.4 Concluding remarks

The experimental study investigated the ability of each of the three sonification paths to translate the search behaviour of the algorithms NSGA-II, MOEA/D and MRS into sound. The three algorithms were applied on a collection of bi-objective problems exhibiting different challenges for an optimizer (e.g. local fronts, disconnected fronts, and fronts of different shapes). *Sonification paths 1* and *2* provide information on different levels. *Sonification path 1* offers an insight into the exploratory phase of the optimisation process, during which the approximation set shape is expected to change drastically over the course of a few generations. *Sonification path 2* becomes particularly useful during the exploitative phase of the optimisation. During this phase, the approximation set presents small changes from one generation to the next, and therefore the solution recurrence can tell us more about the behaviour of the algorithm. *Sonification path 3* sits in between the other two paths, facilitating the understanding of the algorithm both on a macro (approximation set) and micro (individual solution) level. This becomes possible by informing the user not only on which solutions contribute the most towards the hypervolume, but also on the way the hypervolume contributions distribute across the approximation set. The latter notifies the user about the optimisation progress, since the contributions tend to change dramatically during the exploratory phase and eventually reach stability (or fluctuate minimally) during the exploitative phase. We were also able to show how *sonification path 3* works in tandem with *sonification path 1*, since it provides information on the shape of the approximation set. With the addition of *sonification path 3*, SonOpt facilitates the understanding of the algorithm behaviour, providing useful overviews (approximation set shape, convergence etc.) as well as specific insights (individual solution recurrence, hypervolume contributions), through the information conveyed jointly by the three sonification paths.

Apart from providing different levels of information on the algorithmic behaviour, the co-existence of the three sonification paths is important for one more

reason. As shown in cases like MOEA/D on ZDT4, and NSGA-II or MOEA/D on Kursawe, sometimes the audition of a single sonification path cannot provide sufficient information on the convergence of the algorithm. In the case of MOEA/D on ZDT4, the presence of an outlier solution in the objective space causes the shape of the resulting waveform to produce plenty of harmonics (Fig. 9), even though the algorithm is considered to have converged successfully. In a similar fashion, the discontinuity of the Pareto front in the case of Kursawe results in waveforms that present increased levels of harmonics after convergence has been achieved (Figs. 10 and 11 respectively). This shortcoming of *sonification path 1* is compensated by the functionality of the two other paths and especially *sonification path 3*. As pointed out in Sect. 6.3.2, the solutions located on the breakpoints across the objective space tend to contribute more to the hypervolume. Therefore these breakpoints become audible, through the pitch “spikes” they cause on the sound of *sonification path 3*. This can help the user identify that convergence has been achieved. Similarly, Fig. 19 shows that the achieved convergence of MOEA/D is aurally represented through the produced sound of path 3, without the outlier noticeably affecting the outcome. We thus encourage the users of SonOpt to consult all three paths in order to obtain an accurate estimation of the algorithmic behaviour. The spectral characteristics of the sounds produced by *sonification path 3* vary significantly to the ones produced by the other two paths, hence allowing discernible results during the simultaneous audition of path 3 with these paths. The option of placing the sonic output of each path on a different place in the stereo panorama is also provided, in order to further enhance sonic clarity.

7 SonOpt as a creative tool

In the previous sections we discussed and demonstrated SonOpt’s ability to facilitate the understanding of algorithmic behavior for bi-objective optimisation processes. SonOpt is a sound generating application and, as such, can also function as a creative tool for composers that look into experimental techniques for creating new sounds. The common principle behind the three sonification paths is the creation of sound with data extracted from the algorithmic run (approximation set points and hypervolume contributions). Experimental sound synthesis techniques have been used many times and there are numerous applications that provide the user with a set of options to generate new waveforms [11, 47]. The legendary tool UPIC (Unité Polyagogique Informatique CEMAMu) designed by Iannis Xenakis, considered a remarkable innovation at its time [37], allowed the user to draw a waveform on 2D display. This waveform could be used later on as the building block for a musical composition. In SonOpt, the waveform design is handled by the optimisation algorithm. A straightforward application of this can be seen in *sonification path 1*. Although not by directly acting on the shape of the waveform, in *sonification path 2* the algorithm manipulates the produced sound by adding or subtracting harmonic partials on a fundamental frequency. This path can generate a vast range of sounds with various degrees of complexity. *Sonification path 3* involves a simplified version of Frequency Modulation [12] synthesis.

Depending on the scaling range set by the user, pulsing rhythmic patterns, sounds with a vibrato effect or extreme frequency-modulated sounds are a few of the possible outcomes. The selected duration of each generation can create very short, percussive sounds, up to long sonic structures. The objective space thus becomes a source for the generation of sonic timbres (*sonification paths 1 and 2*) and rhythmic patterns (*sonification path 3*).

SonOpt features a comprehensive collection of parameters that can shape the sonic outcome of each sonification path. Default values have been provided where possible in order to minimize the necessary time to setup the system. However, for the user who wishes to delve into the sound generation capabilities of SonOpt, the further adjustment of these parameters provides plenty of room for experimentation. The familiarity of the artistic community with Max/MSP invites musicians and creative individuals to explore the sonic potential of the tool. Through SonOpt, multi-objective optimisation is examined as a dynamic process that can provide rich sonic experimentation.

8 Conclusions and future work

We presented the motivation and methodology behind SonOpt, an open-source tool designed to facilitate the understanding of the population-based algorithmic behaviour in bi-objective optimisation tasks. This paper has extended on the authors' previous work, through the addition of a third audio output that involves the sonification of the hypervolume contributions as they evolve during the optimisation. This extra output offers insight on the relative importance of the objective function values. Alongside the collection of optimisation scenarios involving NSGA-II and MOEA/D, we further included a multi-objective random search (MRS) algorithm into the experimental study to demonstrate the responsiveness to SonOpt to algorithms relying on different search paradigms. Moreover, we used a variety of bi-objective optimisation problems to demonstrate the functionality and effectiveness of SonOpt through its three sonification paths. These paths offer insights into various characteristics of the optimisation process as well as properties of the problem being solved: on a higher level we obtain information on the shape of the approximation set and thus the Pareto front eventually, convergence or stagnation, location of discontinuities, and population diversity, and on a per-solution basis SonOpt offers insights including recurrence and location of recurrent values within the set, hypervolume contributions and distribution of these across the set. We also discussed the creative aspect of SonOpt as an experimental sound synthesis tool that encourages composers and sonic artists to examine the optimisation algorithms as a generative source of musical material.

Sonification of multi-objective optimisation is still at its infancy and SonOpt is the first attempt towards this direction. As such, the application presents plenty of opportunity for further improvement. Our immediate goal is on

expanding SonOpt to sonify problems with more than two objectives to support understanding of algorithm behaviour for many-objective optimisation problems. Research suggests that the difficulty of many-objective problems depends on features of both algorithms and problems [27], and that increasing the number of objectives can have an impact on various optimisation-related routines [2, 3]. Consequently, adapting SonOpt to such problems may require updating existing sonification paths (e.g. *sonification path 1*) and/or the development of additional sonification paths. Thinking even further ahead, the concept of sonification can be used to capture various other aspects related to an algorithm's search behaviour, such as level of feasibility and robustness of solutions as a way to understand better the impact of constraint and uncertainty-handling strategies, respectively. The addition of sonification paths in turn increases the set of sound-related parameters for the user to interact with (either a priori or in real-time during an optimisation run). Enhanced interactivity can also expand the creative application of the tool, since it will encourage further experimentation with the sonic results. Furthermore, to improve the usability of SonOpt, we aim to move SonOpt entirely to Python (at the moment, only the optimisation algorithms are run in Python but the actual sound generation is done in Max/MSP). This will allow for a seamless workflow, improvement of the sonic flexibility to increase the transparency and perceptibility of audio, and the inclusion of sonification of behavioural differences between algorithms in order to facilitate performance comparison. Lastly, we plan to conduct and publish a survey in which individuals will be testing SonOpt in various algorithm and problem combinations and report the insights it can provide as a method for monitoring optimisation.

Appendix A: Additional results of MRS on ZDT4 and Kursawe

See Figs. 25, 26, 27, 28.

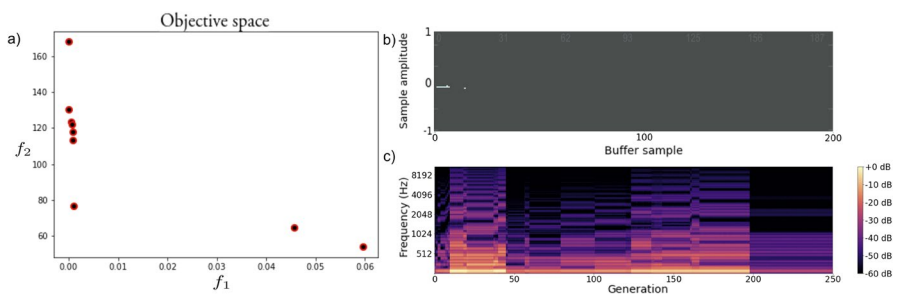


Fig. 25 MRS on ZDT4 via *sonification path 1*. **a** approximation set obtained after 250 generations; **b** buffer contains on the 250th generation; **c** mel spectrogram of sonification path 1 across 250 generations

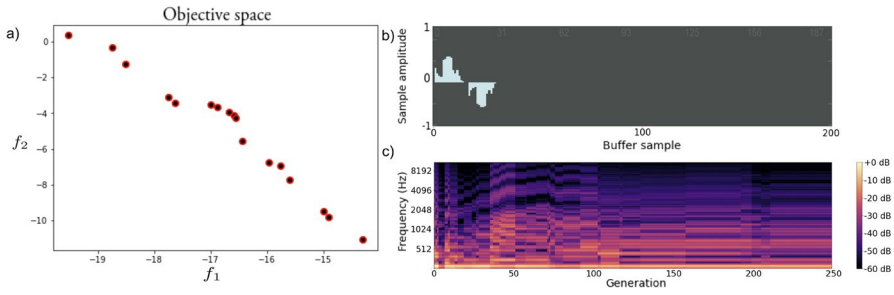


Fig. 26 MRS on Kursawe via *sonification path 1*. **a**, **b**, **c** as above

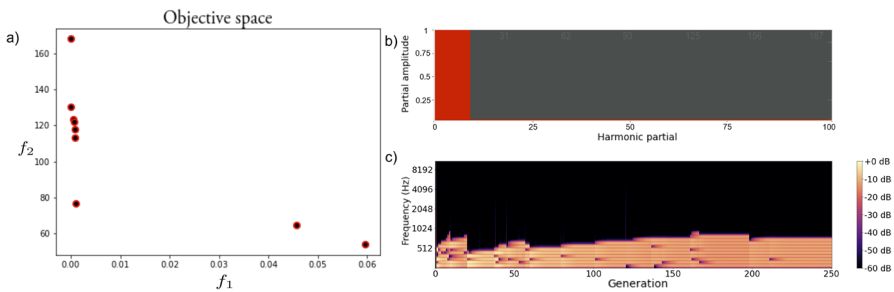


Fig. 27 MRS on ZDT4 via *sonification path 2*. **a** approximation set obtained after 250 generations; **b** amplitude of the harmonic partials on the 250th generation; **c** mel spectrogram of *sonification path 2* across 250 generations

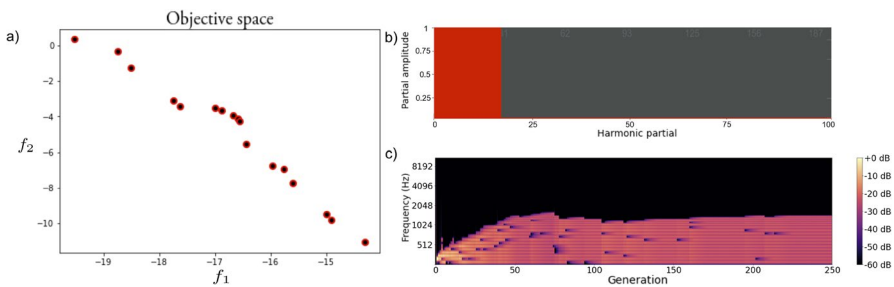


Fig. 28 MRS on Kursawe via *sonification path 2*. **a**, **b**, **c** as above

Appendix B: Optimisation problem definitions

The ZDT and Tanaka [52] definitions are taken from pymoo [8]. The Kursawe and CTP2 are taken from the original papers, [33] and [17] respectively.

B.1 ZDT

The ZDT problem suite [59] is based on the construction process:

$$\begin{aligned} \min f_1(x) \\ \min f_2(x) = g(x)h(f_1(x), g(x)), \end{aligned}$$

where two objective have to be minimized. The function $g(x)$ can be considered as the function for convergence and usually $g(x) = 1$ holds for pareto-optimal solutions (except for ZDT5).

B.1.1 ZDT1

This is a 30-variable problem ($n=30$) with a convex Pareto-optimal set (Fig. 29):

Definition

$$\begin{aligned} f_1(x) &= x_1 \\ g(x) &= 1 + \frac{9}{n-1} \sum_{i=2}^n x_i \\ h(f_1, g) &= 1 - \sqrt{f_1/g} \\ 0 \leq x_i &\leq 1, \quad i = 1, \dots, n \end{aligned}$$

Optimum

$$0 \leq x_1^* \leq 1 \text{ and } x_i^* = 0 \text{ for } i = 2, \dots, n$$

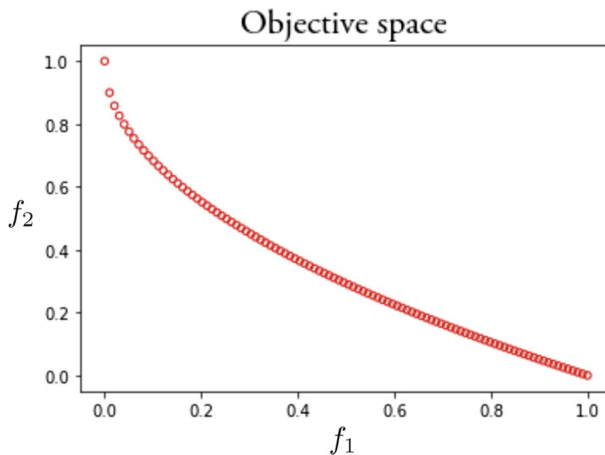


Fig. 29 ZDT1 Pareto front

B.1.2 ZDT3

This is also a 30-variable problem ($n=30$) with a number of disconnected optimal Pareto fronts (Fig. 30):

Definition

$$f_1(x) = x_1$$

$$g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

$$h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1)$$

$$0 \leq x_1 \leq 1, i = 1, \dots, n$$

Optimum

$$0 \leq x_1^* \leq 0.0830$$

$$0.1822 \leq x_1^* \leq 0.2577$$

$$0.4093 \leq x_1^* \leq 0.4538$$

$$0.6183 \leq x_1^* \leq 0.6525$$

$$0.8233 \leq x_1^* \leq 0.8518$$

$$x_i^* = 0 \text{ for } i = 2, \dots, n$$

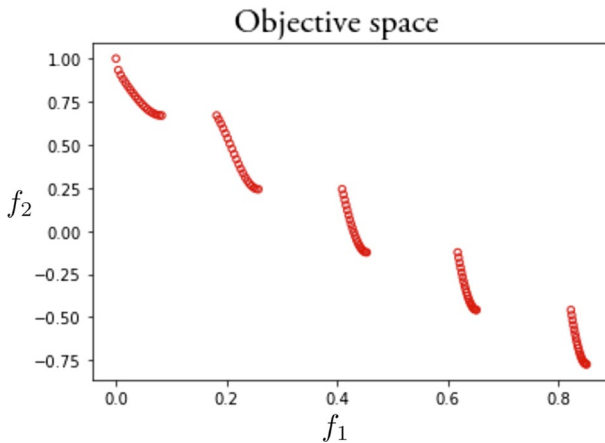


Fig. 30 ZDT3 Pareto front

B.1.3 ZDT4

This is a 10-variable ($n=10$) problem having a convex Pareto-optimal set. There exist many local Pareto-optimal solutions in this problem (Fig. 31). Therefore, algorithms can easily get stuck in a local optimum.

Definition

$$f_1(x) = x_1$$

$$g(x) = 1 + 10(n - 1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$$

$$h(f_1, g) = 1 - \sqrt{f_1/g}$$

$$0 \leq x_1 \leq 1$$

$$-10 \leq x_i \leq 10, i = 2, \dots, n$$

Optimum

$$0 \leq x_1^* \leq 1 \text{ and } x_i^* = 0 \text{ for } i = 2, \dots, n$$

B.2 Kursawe

Definition:

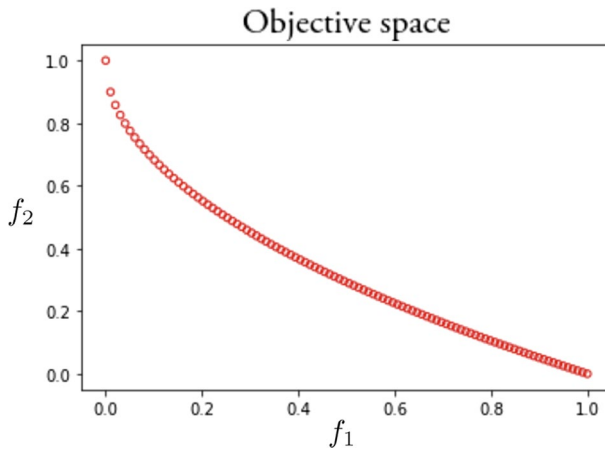


Fig. 31 ZDT4 Pareto front

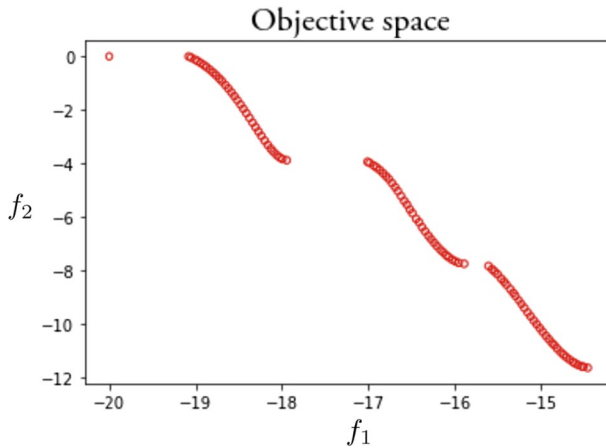


Fig. 32 Kursawe Pareto front

$$\min f_1(x) = \sum_{i=1}^{n-1} (-10e^{(-0.2\sqrt{x_i^2+x_{i+1}^2})}),$$

$$\min f_2(x) = \sum_{i=1}^n (|x_i|^{0.8} + 5 \sin(x_i)^3),$$

with $-5 \leq x_i \leq 5$, for $i = 1, 2, \dots, n$

Pareto front can be seen on Fig. 32.

B.3 Tanaka

Definition:

$$\begin{aligned} \min f_1(x) &= x_1, \\ \min f_2(x) &= x_2, \\ \text{subject to } C_1(x) &\equiv x_1^2 + x_2^2 - 1 - 0.1 \cos(16 \arctan \frac{x_1}{x_2}) \geq 0, \\ C_2(x) &\equiv (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5, \\ 0 &\leq x_1 \leq \pi, \\ 0 &\leq x_2 \leq \pi. \end{aligned}$$

Optimum

Since $f_1 = x_1$ and $f_2 = x_2$, the feasible objective space is also the same as the feasible decision variable space. The unconstrained decision variable space consists of all solutions in the square $0 \leq (x_1, x_2) \leq \pi$. Thus, the only unconstrained

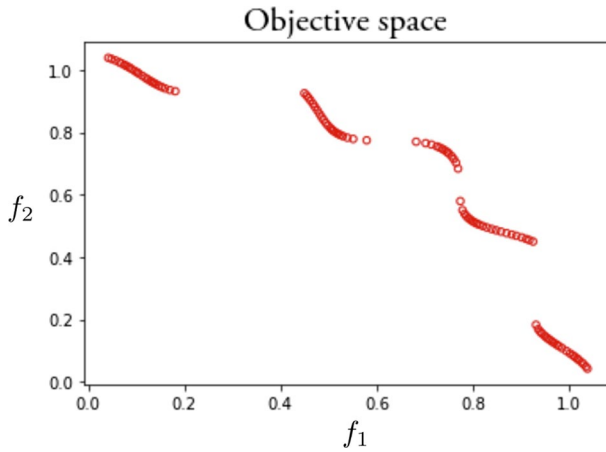


Fig. 33 Tanaka Pareto front

Pareto-optimal solution is $x_1^* = x_2^* = 0$. However, the inclusion of the first constraint makes this solution infeasible. The constrained Pareto-optimal solutions lie on the boundary of the first constraint. Since the constraint function is periodic and the second constraint function must also be satisfied, not all solutions on the boundary of the first constraint are Pareto-optimal. The Pareto-optimal set is disconnected. Since the Pareto-optimal solutions lie on a nonlinear constraint surface (Fig. 33), an optimisation algorithm may have difficulty in finding a good spread of solutions across all of the discontinuous Pareto-optimal sets.

B.4 CTP2

Definition

$$\begin{aligned} \min \quad & f_1(x) = x_1, \\ \min \quad & f_2(x) = g(x) \left(1 - \frac{f_1(x)}{g(x)} \right), \\ \text{Subject to} \quad & c(x) \equiv \cos(\theta)(f_2(x) - e) - \sin(\theta)f_1(x) \geq \\ & \alpha |\sin(b\pi(\sin(\theta)(f_2(x) - e) + \cos(\theta)f_1(x))^c)|^d. \end{aligned}$$

The decision variable x_1 is restricted in $[0, 1]$ and the bounds of other variables depend on the chosen $g(x)$ function. The constraint has six parameters ($\theta, \alpha, b, c, d,$ and e). Pareto front can be seen on Fig. 34.

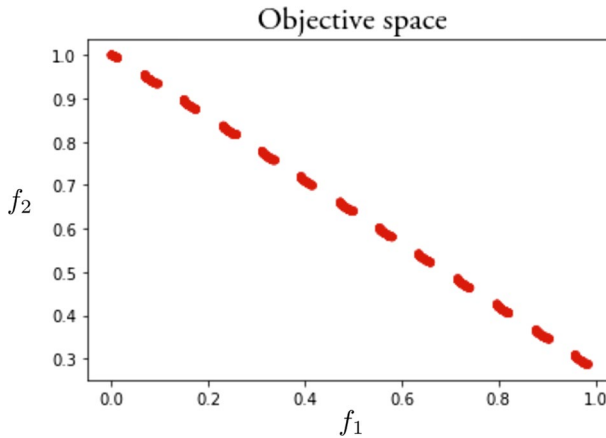


Fig. 34 CTP2 Pareto front

Acknowledgements This work was supported by the Engineering and Physical Sciences Research Council [Grant Number 2492452].

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. S. Ali, L. Muralidharan, F. Alfieri, M. Agrawal, J. Jorgensen, in *Sonify: making visual graphs accessible*. in Proceedings of the 1st International Conference on Human Interaction and Emerging Technologies **1018**, pp. 454–459 (2019)
2. R. Allmendinger, A. Jaskiewicz, A. Liefooghe, C. Tammer, What if we increase the number of objectives? Theoretical and empirical implications for many-objective combinatorial optimisation. *Comput. Op. Res.* **145**, 105857 (2022)
3. R. Allmendinger, J. Knowles, Heterogeneous objectives: state-of-the-art and future research, arXiv preprint [arXiv:2103.15546](https://arxiv.org/abs/2103.15546) (2021)
4. T. Asonitis, R. Allmendinger, M. Benatan, R. Climent, in *SonOpt: sonifying bi-objective population-based optimisation algorithms*. International Conference On Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar), pp. 3–18 (2022)

5. L. Axon, B. Alahmadi, J. Nurse, M. Goldsmith, S. Creese, Data presentation in security operations centres: exploring the potential for sonification to enhance existing practice. *J. Cybersecur.* **6**, 1–16 (2020)
6. S. Barrass, G. Kramer, Using sonification. *Multimedia Syst.* **7**, 23–31 (1999)
7. T. Blackwell, M. Young, Self-organised music. *Organ. Sound* **9**(2), 123–136 (2004)
8. J. Blank, K. Deb, Pymoo: multi-objective optimisation in python. *IEEE Access* **8**, 89497–89509 (2020)
9. N. Bonet Filella, Data Sonification in Creative Practice, PhD thesis, University of Plymouth (2019)
10. K. Bringmann, T. Friedrich, An efficient algorithm for computing hypervolume contributions. *Evol. Comput.* **18**, 383–402 (2010)
11. A. Brown, G. Jenkins, in *The interactive dynamic stochastic synthesizer*. Proceedings of the Australian Computer Music Conference 2004 Ghost in the Machine Performance Practice in Electronic Music, pp. 18–22 (2004)
12. J. Chowning, The synthesis of complex audio spectra by means of frequency modulation. *J. Audio Eng. Soc.* **21**, 526–534 (1973)
13. J. Crawford, M. Watson, O. Burmeister, P. Sanderson, in *Multimodal displays for anaesthesia sonification: timesharing, workload, and expertise*. Proceedings of the Joint ESA/CHISIG Conference on Human Factors (2002)
14. I. Das, J. Dennis, Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimisation problems. *SIAM J. Optim.* **8**, 631–657 (1998)
15. A. De Campo, in *Toward a data sonification design space map*. 13th International Conference in Auditory Display, pp. 342–347 (2007)
16. K. Deb, *Multi-objective optimisation using evolutionary algorithms* (Wiley, New Jersey, 2001)
17. K. Deb, A. Pratap, T. Meyarivan, in *Constrained test problems for multi-objective evolutionary optimisation*. International Conference on Evolutionary Multi-criterion Optimisation, pp. 284–298 (2001)
18. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
19. F. Dombois, G. Eckel, Audification, in *The Sonification Handbook*. ed. by T. Hermann, A. Hunt, J.G. Neuhoff (Logos Publishing House, Berlin, 2011), pp.301–324
20. M. Emmerich, C. Fonseca, in *Computing hypervolume contributions in low dimensions: asymptotically optimal algorithm and complexity results*. International Conference on Evolutionary Multi-criterion Optimisation, pp. 121–135 (2011)
21. J. Fieldsend, T. Chugh, R. Allmendinger, K. Miettinen, in *A feature rich distance-based many-objective visualisable test problem generator*. Proceedings of the Genetic and Evolutionary Computation Conference, pp. 541–549 (2019)
22. J. Fieldsend, T. Chugh, R. Allmendinger, K. Miettinen, A visualizable test problem generator for many-pbjective optimisation. *IEEE Trans. Evol. Comput.* **26**(1), 1–11 (2022)
23. F. Grond, O. Kramer, T. Hermann, in *Interactive sonification monitoring in evolutionary optimisation*. 17th International Conference on Auditory Display, p. 166 (2011)
24. T. Hermann, A. Hunt, J. Neuhoff, *The Sonification Handbook* (Logos, Berlin, 2011)
25. T. Hildebrandt, T. Hermann, S. Rinderle-Ma, Continuous sonification enhances adequacy of interactions in peripheral process monitoring. *Int. J. Hum Comput Stud.* **95**, 54–65 (2016)
26. M. Iber, P. Lechner, C. Jandl, M. Mader, M. Reichmann, Auditory augmented process monitoring for cyber physical production systems. *Pers. Ubiquit. Comput.* **25**(4), 691–704 (2021)
27. H. Ishibuchi, N. Akedo, H. Ohyanagi, Y. Nojima, in *Behavior of EMO algorithms on many-objective optimization problems with correlated objectives*. 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 1465–1472 (2011)
28. R. Johnson, in *Wavetable synthesis 101, a fundamental perspective*. Audio Engineering Society Convention, pp. 1–27 (1996)
29. J. Kennedy, R. Eberhart, Particle swarm optimisation, proceedings of international conference on Neural Netw. **4**, 1942–1948 (1995)
30. M. Köppen, R. Vicente-García, B. Nickolay, in *Fuzzy pareto-dominance and its application in evolutionary multi-objective optimisation*. International Conference on Evolutionary Multi-Criterion Optimisation, pp. 399–412 (2005)
31. M. Köppen, K. Yoshida, in *Substitute distance assignments in NSGA-II for handling many-objective optimisation problems*. International Conference on Evolutionary Multi-Criterion Optimisation, pp. 727–741 (2007)

32. G. Kramer, *Auditory display: Sonification* Audification and Auditory Interfaces. (Addison-Wesley, Boston, 2000)
33. F. Kursawe, in *A variant of evolution strategies for vector optimisation*. International Conference on Parallel Problem Solving from Nature, pp. 193–197 (1990)
34. Y. Lavinias, C. Aranha, G. Ochoa, in *Search trajectories networks of multi-objective evolutionary algorithms*. International Conference on the Applications of Evolutionary Computation (Part of EvoStar), pp. 223–238 (2022)
35. E. Lutton, H. Gilbert, W. Cancino, B. Bach, J. Pallamidessi, P. Parrend, P. Collet, Visual and audio monitoring of island based parallel evolutionary algorithms. *J. Grid Comput.* **13**(3), 309–327 (2015)
36. Z. Lyu, J. Li, B. Wang, in *Alive: interactive visualization and sonification of neural networks in virtual reality*. 2021 IEEE International Conference on Artificial Intelligence And Virtual Reality (AIVR), pp. 251–255 (2021)
37. G. Marino, M. Serra, J. Raczinski, in *The UPIC System: Origins and Innovations*. Perspectives of New Music, pp. 258–269 (1993)
38. K. Marriott, B. Lee, M. Butler, E. Cutrell, K. Ellis, C. Goncu, M. Hearst, K. McCoy, D. Szafir, Inclusive data visualisation for people with disabilities: a call to action. *Interactions* **28**, 47–51 (2021)
39. K. Miettinen, *Nonlinear Multi-objective Optimisation* (Springer Verlag, New York, 2012)
40. G. Ochoa, K. Malan, C. Blum, Search trajectory networks: a tool for analysing and visualising the behaviour of metaheuristics. *Appl. Soft Comput.* **109**, 107492 (2021)
41. M. Pilhofer, H. Day, *Music theory for dummies* (John Wiley and Sons, New Jersey, 2019)
42. R. Plomp, The ear as a frequency analyzer. *J. Acoust. Soc. Am.* **36**, 1628–1636 (1964)
43. R. Plomp, A. Mimpen, The ear as a frequency analyzer II. *J. Acoust. Soc. Am.* **43**, 764–767 (1968)
44. M. Poguntke, K. Ellis, in *Auditory attention control for human-computer interaction*. Conference on Human System Interactions, pp. 231–236 (2008)
45. M. Puckette, in *The patcher, Proceedings of The 1986 International Computer Music Conference*. pp. 420–429 (1988)
46. I. Rechenberg, Evolutionsstrategien, Simulationsmethoden In der Medizin und Biologie, pp. 83–114 (1978)
47. G. Roma, G.A. Xambó, A tabletop waveform editor for live performance, NIME, pp. 249–252 (2008)
48. N. Sawe, C. Chafe, J. Treviño, Using data sonification to overcome science literacy. *Numeracy Vis. Barriers Sci. Commun. Front. Commun.* **5**, 46 (2020)
49. B. Schuller, T. Virtanen, M. Riveiro, G. Rizos, J. Han, A. Mesaros, K. Drossos, in *Towards sonification in multimodal and user-friendly explainable artificial intelligence*. Proceedings of the 2021 International Conference on Multimodal Interaction, pp. 788–792 (2021)
50. J. Smith, *Spectral audio signal processing*, (W3K, Branson, 2011)
51. S. Stevens, J. Volkman, The relation of pitch to frequency: a revised scale. *Am. J. Psychol.* **53**, 329–353 (1940)
52. M. Tanaka, H. Watanabe, Y. Furukawa, T. Tanino, in *GA-based decision support system for multicriteria optimisation*. 1995 IEEE International Conference On Systems, Man And Cybernetics. Intelligent Systems for the 21st Century **2**, pp. 1556–1561 (1995)
53. T. Tavares, A. Godoy, in *Sonification of population behaviour in particle swarm optimisation*. Proceedings of Annual Conference Companion on Genetic and Evolutionary Computation, pp. 51–52 (2013)
54. T. Tušar, B. Filipič, Visualisation of pareto front approximations in evolutionary multiobjective optimisation: a critical review and the prosection method. *IEEE Trans. Evolut. Comput.* **19**(2), 225–245 (2015)
55. P. Vickers, Sonification for Process Monitoring, in *The Sonification Handbook*. ed. by T. Hermann, A. Hunt, J.G. Neuhoff (Logos Publishing House, Berlin, 2011), pp.455–492
56. M. Wright, A. Freed, in *Open sound control: a new protocol for communicating with sound synthesizers*. International Computer Music Conference, pp. 101–104 (1997)
57. Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)
58. E. Zitzler, Evolutionary algorithms for multiobjective optimisation: methods and applications. PhD Thesis, Swiss Federal Institute of Technology (1999)
59. E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results. *Evol. Comput.* **8**(2), 173–195 (2000)

60. E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, V. Da Fonseca, Performance assessment of multi-objective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**, 117–132 (2003)
61. E. Zitzler, J. Knowles, L. Thiele, Quality Assessment of Pareto set Approximations, *Multiobjective Optimisation*, 2008), pp.373–404

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Tasos Asonitis¹  · Richard Allmendinger² · Matt Benatan³ · Ricardo Climent¹

¹ NOVARS Research Department, University of Manchester, Manchester M13 9PL, United Kingdom

² Alliance Manchester Business School, University of Manchester, Manchester M15 6PB, United Kingdom

³ Sonos Experience Limited, London SE1 3ER, United Kingdom