Sonification of population behavior in Particle Swarm Optimization

Tiago F. Tavares  
University of Campinas  
Laboratory of Signal Processing for Communications  
School of Electrical and Computer Engineering  
Av. Albert Einstein, 400  
Campinas, SP, Brazil  
tavares@dca.fee.unicamp.br

Alan Godoy  
University of Campinas  
Laboratory of Bioinformatics and Bioinspired Computing  
School of Electrical and Computer Engineering  
Av. Albert Einstein, 400  
Campinas, SP, Brazil  
godoy@dca.fee.unicamp.br

ABSTRACT
High-dimensional optimization problems may be addressed using populational meta-heuristics, whose statistical properties may indicate important characteristics of the optimization process. There is a great number of these properties, which means their joint visualization may become impractical. We developed a method for sonically displaying characteristics of population dynamics in particle swarm optimization processes as soundscape parameters. This process allows jointly analyzing several dimensions of the population's dynamics. Moreover, design decisions aimed at generating aesthetically appealing soniscapes, which allows the proposed system to be used as an automated music composition environment.

1. INTRODUCTION
Sonification consists of artificially changing the sonic characteristics of a system so that its behavior can be more accurately informed to a human user [8]. There are many practical uses for sonification, one of which is sonically displaying data for exploration. Sonification has also been used to provide insight regarding the behavior of scientific data streams [6].

We propose a system for the sonification of the population dynamics in Particle Swarm Optimization (PSO), a system with multiple interacting agents aimed at locating the best solution for a real-parameter problem. As PSO’s optimization ability of the PSO is the product of the swarm’s collective behavior, it is interesting to develop ways to efficiently monitor the population. Our system allows jointly analyzing relevant characteristics of agent populations by mapping them into soundscape parameters. The final sonic results aim at giving insight regarding the general behavior of the optimization algorithm.

In addition, the soundscapes were designed to be aesthetically appealing, which means that they can be listened to as artistic musical pieces generated by an algorithmic composition process. Although populational meta-heuristics have been largely used to compose music, for example in work by Blackwell and Young [4] or by Biles [3], to our knowledge no composition process used a mapping of the characteristics of the population dynamics – instead of their results – for the generation of soundscapes. The proposed system is closer to the well-known fractal image generators, which may be viewed as an equation display, but also simply as artistic products.

This paper is organized as follows. Section 2 briefly describes related work in sonification and musical composition using PSO and other population dynamics. In Section 3, the PSO algorithm is discussed. The proposed sonification process is described in Section 4. Performed tests and obtained results are described in Section 5. Last, conclusive remarks are given in Section 6.

2. RELATED WORK
Sonification is the process of displaying useful information through a non-speech, sonic interface. Early work has shown that important information about diverse systems can be conveyed using audio [16]. Currently, research focuses on different ways of displaying information [8], aiming at optimizing the use of auditory media.

A process that benefits from sonification is the exploration of scientific data [9, 6, 8]. Listening to a sonification can trigger insight about the behavior of a dataset, which is important for data analysis. There are many techniques for sonification [16, 8], which have to be carefully adapted to each scenario, so that the resulting sound is a useful representation of the analyzed dataset.

Vogt [20] has stated that multiple dimensions of the same dataset can be simultaneously displayed using sonification. However, sonic displays do not show absolute values, cannot be printed and are subject to cultural biases. Walker and Kramer [21] note that the designer’s sonification decisions do not always match the expectations of the user, leading to a worse performance on the analysis.

The translation of population dynamics into sound has been used for the production of artistic pieces, remarkably by Blackwell and Young [4] and by Jones [11]. In both cases, the behavior of each individual of the multi-agent system was translated into sound, so that the final outcome would
be the result of the system’s self-organization. Both proposals are able to produce interesting artistic results, but these results are not very useful for analyzing aspects of the system’s dynamics related to its effectiveness to locate good results.

3. PARTICLE SWARM OPTIMIZATION

The Particle Swarm Optimization (PSO) [12] is a population meta-heuristic for optimization of real-valued search spaces problems. This means that, as all meta-heuristics, PSO is a search procedure for generic problems that, despite its effectiveness in many scenarios, does not guarantee the optimal (or near-optimal) solution will be found in finite time. Also, being a population technique means that, instead of improving one solution at a time (as happens in well-known algorithms, as Simulated Annealing [15]), PSO works with a population of candidate solutions that are iteratively combined in order to locate better solutions. PSO is considered part of a broader family of methods named Evolutionary Algorithms, which include the classical Genetic Algorithm [10], aimed at solving combinatorial problems, and other techniques that, like PSO, are intended for real-valued problems, as Differential Evolution [18] and Evolution Strategies [2].

Unlike most Evolutionary Algorithms, which draw inspiration from genetics and Darwinian evolution, Particle Swarm Optimization, as its name suggests, is based on collective behavior of social animals and insects, specially on social sharing of information in search and retrieval of food. In PSO, each individual (known as particle in this context) moves through the search space with its own velocity, evaluating the quality of each potential solution visited. The particle, then, combine its velocity with information about its own history (cognitive influence) and the history of its social contacts (social influence) to define the next direction to explore. As depicted in Algorithm 1, in the canonical algorithm these influences are implemented as two attractors, one positioned on the best location visited by the particle itself and the other on the best position found by any of the particle’s neighbors.

Algorithm 1: Particle Swarm Optimization for minimization

```
input: ω, c1, c2, N and neighborhood
Initialize position xi, velocity vi, and personal best position pi for each of the N particles;
while stop criterion is not satisfied do
    for i = 1 to N do
        gi ← arg min f(xj(i)) f(pi);
        vi ← ωvi + c1r1(p(i)−x(i)) + c2r2(g−x(i));
        xi ← xi + vi;
        if xi is within search range and f(xi) < f(pi) then
            pi ← xi;
        end
    end
end
```

In Algorithm 1, f is the fitness function to be minimized, ω is the inertia parameter, c1 is the cognitive factor, c2 is the social factor and r1 and r2 are random vectors with each dimension in the interval [0, 1]. The inertia parameter controls the influence of the previous velocity on the search direction, while the cognitive and the social parameters define the weight each individual will give, respectively, to its own history and to influence from its peers. The network of social contacts (known as “neighborhood”) is also an important parameter that defines much of the collective behavior of the swarm: highly connected neighborhoods stimulate a faster convergence, which may trap the swarm in a low quality local optima, whereas if the connections are too sparse convergence may be delayed, allowing the swarm to spend much time in non-promising regions when it could be better to focus on improving the best solution found so far [13]. Besides the global neighborhood, which is used in the canonical PSO and works by connecting each particle to all other in the swarm so that all particles have the same social influence, many other topologies for social connection have been proposed in the literature. Two other classical static topologies are local and von Neumann, shown in Figure 1. Also, the neighborhood may be modified during the search, as in the state-of-the-art DMS-PSO [17], in which the particles are organized into groups with m individuals that are randomly rearranged each R iterations.

3.1 Measures

Due to its ease of encoding and understanding, together with the fact that it does not need analytical formula or gradient information of the function to be optimized, PSO is applied to a variety of problems, such as optimization of power systems [1], modeling of biomechanical systems [14] and design of VLSI circuits [7]. However, PSO has some shortcomings, the major one being premature convergence, making adequate parameters and auxiliary mechanisms critical to achieve high quality solutions. Optimal configuration of parameters and neighborhoods is problem-dependent, so that a good selection of parameters providing an appropriate balance between exploration and exploitation often requires both an experienced practitioner and some trial and error.

![Figure 1: Examples of neighborhoods in PSO.](image)
Many measures were developed to analyze the behavior of a swarm, being used to assist in the process of fine-tuning parameters for a problem and also to analyze the reasons for the success or failure of a swarm in locating good solutions. Examples of these measures are:

- **Swarm velocity:** the average velocity of particles in the swarm, calculated by
  
  \[ a = \frac{\sum_{i=1}^{N} v_i}{N} \]
  
The current speed is correlated to the balance between exploration and local search. For instance, a slow swarm is probably in a phase of improvement of the current solution, with low capacity to find new, promising regions, while a faster one is probably changing the region it is exploring.

- **Swarm alignment:** similar to the velocity, but considers only the direction and not the module of individual velocities. Higher values indicate that most of the particles are moving in the same direction. Its value is given by
  
  \[ a = \frac{\sum_{i=1}^{N} v_i}{\sum_{i=1}^{N} \| v_i \|} \]
  
  A high alignment indicates that the swarm is massively moving towards a new region. This movement may have as consequence the abandonment of areas that have not yielded good results so far, concentrating efforts on places that seem more interesting, but also can lead to severe loss of diversity.

- **Diversity:** it indicates the scattering of particles through the search space, as follows
  
  \[ d = \frac{\sum_{i=1}^{N} \| x_i - c \|}{N d_{\text{max}}} \]
  
  where \(c\) is the centroid of the swarm and \(d_{\text{max}}\) is the diameter of the search space. A low value indicates the concentration of all particles in a small region, being likely that the system is working with a small set of similar solutions. When good solutions have been located, this concentration may provide better results, but if more exploration is needed a low diversity may cause the optimization process to get trapped in a local optima.

- **Fitness evolution:** the analysis of fitness evolution may indicate specific moments in which the swarm located new optima or entered a phase of stagnation. In this work we used the fitness gain to indicate the evolution of the quality of solutions along the search:
  
  \[ g = \frac{f^{t-1} - f^t}{f^{t-1}} \]
  
  where \(f^t = f(p^*_t) - f_{\min}\), \(p^*_t\) is the best solution found by any individual so far and \(f_{\min}\) is a lower bound for the fitness function, given by the user.

- **Distance to the best solution so far:** Euclidean distance from the centroid of the swarm to the best solution found so far. This measure helps to assess the influence of the current best solution over the swarm. For example, if the centroid is very near the current optimum then the attraction exerted by previous experiences may be too strong, preventing thereby the swarm to find new solutions.

- **Distance to the global optimum:** Euclidean distance from the centroid of the swarm to the global optimum. With this information, it is possible to know how efficient the system was in locating the global optimum. This measure is only available for test functions, in which the location of the global optimum is known.

Each measure will provide only a glimpse into the overall behavior of the swarm, but by monitoring a set of metrics throughout the search one can get valuable insights about the system’s inner workings and the necessary adjustments. For instance, if the distance from the swarm to the best solution found, diversity and speed are low, but the distance to the global optimum and the fitness are high, the swarm is stuck in a low quality local optimum, so that the restart of the particles or the use of a method for increasing diversity may be required.

Real-time monitoring of multiple metrics, however, can be very difficult, hence it is necessary to look for a new way to display informations about the behavior of the swarm. To solve this problem, we propose synthesizing a soundscape whose parameters are directly controlled by the relevant measures that describe the population behavior. In the next section, this process, called sonification, will be described in detail.

4. **THE SONIFICATION PROCESS**

We employed a parameter-mapping-based sonification. As described previously, there are characteristics of the population dynamics that are known to contain useful information about its behavior. This mapping was constructed by selecting a key concept that best describes it in the context of a multi-agent optimization process, and then selecting a sonic parameter that reflects that same concept. The result of this process is shown in Table 1.

The sound generation system is based on a monophonic note synthesis module that, when requested by a controller, yields musical notes that are characterized by its magnitude, duration, pitch and timbre. In the current version, this module works by mixing a sawtooth and a sinusoidal signals of desired frequency (controlling pitch) using a desired mixing ratio (controlling timbre), and then applying an attack-decay-sustain-release (ADSR) envelope to the results (controlling magnitude and duration). The user can change this module, as long as the new synthesizer allows controlling the same parameters (magnitude, duration, pitch and timbre) independently.

The pitch of the generated notes is randomly chosen between the first \(p\) degrees of a pre-defined musical scale, where \(p\) is an integer number that is higher when the tension (as defined in Table 1) of the population is higher. Additionally, the scale is transposed to a user-defined tone. The experiments described in this paper used a pentatonic scale, but this can be configured by the user.
<table>
<thead>
<tr>
<th>Population characteristic</th>
<th>Key concept</th>
<th>Sonic parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greater speed in search space</td>
<td>Speed</td>
<td>Notes are yielded faster</td>
</tr>
<tr>
<td>Greater population alignment</td>
<td>Simplicity</td>
<td>Less harmonics are produced</td>
</tr>
<tr>
<td>Greater diversity</td>
<td>Ambience</td>
<td>More echo is heard</td>
</tr>
<tr>
<td>Greater relative fitness gain</td>
<td>Energy</td>
<td>Increased loudness</td>
</tr>
<tr>
<td>Population is further away from the best solution so far</td>
<td>Tension</td>
<td>Notes have greater chance of reaching higher pitches</td>
</tr>
<tr>
<td>Population is closer to the global optimum</td>
<td>Serenity</td>
<td>Notes have greater duration</td>
</tr>
</tbody>
</table>

Table 1: Metaphors used in sonification.

The signal yielded by the note emission module is directed both to the audio output (right channel) and to a memory which will be used to build the echo effect. The memory is read with a delay (whose duration is set by the controller module), low-pass filtered at 1 kHz to add the illusion of distance, multiplied by a gain (proportional to the population diversity) and directed both to the audio output (left channel) and to the memory itself. This configuration allows separating ambience sound from direct sound, while creating more echo – hence, the illusion of more voices – when the particles are more spread in the search space.

The controller module is responsible for reading the measures after each optimizing iteration and implementing the mappings described in Table 1. Each dimension of the dynamic’s characteristics must be scaled to fit the range within the soundscape parameters are acceptable. We have chosen to leave this scaling for the user, as diverse scalings can lead to a better interpretation of different parts of the population dynamics, as well as different aesthetic experiences.

In addition to the sonification, a short hissing sound in the left ear was played each 50 iterations, so they can be better located in the real-time audio. This hissing sound does not interfere with the soundscape analysis and can be disabled by the user.

Since all characteristics of the dynamics are being displayed at the same time, a great magnitude in one (or a few) of them can mask important differences in the others. Preliminary tests have shown that this is specially true when search spaces of different sizes and dimensions are being compared – in this case, the consequent difference in the swarm speed (related to the sound tempo) masks the other relevant differences. Hence, it may be helpful to normalize the population speed and all distance-based features of each dynamic before sonifying it.

The sonification process was implemented as a patch using the PureData 0.43.4 environment, and the multi-agent dynamics was implemented using Python 2.7. To ensure reproducibility of the results, the source code was made available for download at the URL [https://dl.dropbox.com/u/64294901/Swarm/index.html](https://dl.dropbox.com/u/64294901/Swarm/index.html). In the next section, sonic examples are discussed.

5. TESTS AND RESULTS

The evaluation process consisted of a careful analysis of the generated soundscape, aiming at detecting if meaningful representations of the multi-agent dynamics could be produced. We used a Particle Swarm Optimization process to produce the sonification coefficients, and we evaluated changes on the swarm size, the function to be optimized, the swarm parameters (social and individual) and the swarm neighborhood. All sound examples used in this section are available for immediate listening at the URL [https://dl.dropbox.com/u/64294901/Swarm/index.html](https://dl.dropbox.com/u/64294901/Swarm/index.html).

The tests used the following default configuration for the PSO: $\omega = 0.7298$, $c_1 = c_2 = 1.49618$, 50 individuals, connected through a global neighborhood (all particles are neighbors of each other) and a maximum of 1000 iterations as stop criterion. The functions to be optimized were selected from the test suite for the IEEE Congress on Evolutionary Computation (CEC) 2005 Special Session on real-parameter optimization [19]. The function F8 (Shifted Rotated Ackley’s Function with Global Optimum on Bounds – this function for 2 dimensions is displayed in Figure 2) with 50 dimensions was used in the default configuration. For each test, a different configuration parameter was changed, allowing the analysis of its influence in the optimization dynamics and the sonification process.

Swarms of four different sizes were evaluated, with 1, 5, 50 and 500 particles. It is possible to hear that the swarms with less particles move faster at first, but converge faster. This behavior cannot be clearly seen using visual aids, as shown in Figure 3.

Also, it is possible to hear sudden speed increases when a new attractor is found, generating interesting auditory artifacts. In the case of the swarm with one particle, it is possible to hear that there is, obviously, no diversity and the particle alignment is always maximum. This matches the results shown by visual aids, as it can be seen in Figure 4.

Also, different benchmark functions were used, namely F2 (Shifted Schwefel’s Problem 1.2, shown in Figure 5) and F19 (Rotated Hybrid Composition Function with narrow basin global optimum, shown in Figure 6), both with 50 dimensions. While F8 is a multimodal function (function with
multiple local optima) with most of its search space consisting of solutions with similar fitness values. F2 is a very simple function formed by only one local optimum with a large basin of attraction that reaches all the search space. F19, on the other hand, is a multimodal function with high variation on fitness, local optima with large basins and the global optimum located on a narrow basin of attraction, which makes hard for any technique to locate the best solution.

It was noticed that, due to differences on search spaces sizes, the swarm speed when using F2 was much faster, and when using F19 was much slower than when using F8. To avoid masking other relevant characteristics, the dynamics were normalized and sonified again. In all cases, it is possible to hear the behavior of the swarm when finding new local optima, which is noticeably different in each case.

After some iterations, the tempo in the optimization of F2 and F19 becomes slow, the pitch in both cases becomes monotonous and the spatialization becomes inaudible. This indicates that the PSO algorithm is will not be able to find new attraction basins in these functions. On the other hand, when optimizing F8, these characteristics do not appear, which indicates that the optimization process may still find better results. As it can be seen in Figure 7, this matches what can be observed using visual aids.

It was tested, then, the change in the balance between cognitive and social influences. In order to make the modification in the swarm behavior to be primarily due to the changes in the influence of individual and social experiences, and not to other factors such as the variation in the relative weight of previous velocity in the calculation of the direction to be followed, the sum of cognitive and social parameters was kept constant. Therefore, $c_1 + c_2 = 2.99236$, following Type I restriction from Clerc and Kennedy [5].

By changing the individual and social coefficients, it is possible to hear two interesting characteristics of the swarm. First, when the social and individual coefficients are respectively changed to 0.19618 and 2.79618, the swarm quickly converges to a local optimum and the particle movement converges to the best solution found at first. When the coefficients are respectively changed to 2.79618 and 0.19618,
the swarm explores areas that are further away from the currently known local optimum, which is noticeable by the higher pitches present in the generated soundscape. This matches what can be visualized in Figure 8.

Last, different neighborhoods for the swarm were analyzed, namely the complete, local, von Neumann and DMS-PSO. The most evident difference found regards the exploration capabilities, that is, the distance to best local optimum found in the optimization process related to each neighborhood, which is sonically displayed by the presence of higher pitches. It can be heard that the complete neighborhood presents lower notes, showing lesser exploration capabilities. Also, it can be heard that the local and the von Neumann neighborhoods have greater exploration capabilities, which slightly degrade along the iterations. Last, it can be heard that the exploration capabilities of the DMS-PSO is kept high during all the optimization process. The observation derived from the soundscape analysis matches the one obtained using visual aids, as shown in Figure 9.

The generated soundscapes seemed aesthetically appealing, presenting new, interesting sonic material until the swarm converged to the best local optimum it finds in the optimization process. After the convergence, the soundscape quickly becomes boring. To avoid that, it is possible to either restart the optimization process after some iterations or to dynamically change the optimization function. The duration of the interesting part of each soundscape, however, was around one and a half minute, which is enough to use them as short musical pieces or as an automatic process for generating musical material.

The next section presents conclusive remarks and notes for future work.

6. CONCLUSION

This paper described a method for the sonification of the behavior of a population in PSO dynamics, sonically displaying relevant features of these dynamics. It was noted that long-term measure variations are harder to observe, but short-term variations are more noticeable. Also, the gener-
ated soundscapes are useful for generating insight on how the population behaves, which may lead to education and research applications.

The yielded soundscapes allow joint analysis of all chosen characteristics – the speed, alignment, diversity, fitness gain in each iteration, distance from the best local optimum found by the population and distance from a known global optimum. The joint, auditory analysis highlights aspects and correlations that may be harder to perceive if only visual aids are used. On the other hand, the sonic environment has limited resolution and does not allow displaying absolute values, hence visual aids should not be disregarded.

The aesthetics of the results differ from previous approaches as it is based on the statistical properties of the population, instead of using the movement of the agents to produce sound. Specific parameters for the optimization algorithm may be chosen inducing known behaviors in the population dynamics, leading to the control of the generated soundscapes. Therefore, the genuine musical pieces that are generated by the proposed system are an aesthetically appealing representation of an optimization process that can produce sound. Specific parameters for the optimization algorithm may be chosen inducing known behaviors in the population dynamics, leading to the control of the generated soundscapes. The genuine musical pieces that are generated by the proposed system are an aesthetically appealing representation of an optimization process that can be freely controlled by the user by changing parameters to create different, interesting soundscapes.

The proposed method was tested using Particle Swarm Optimization, but any populational heuristics (Genetic Algorithms or Ant Colony Optimization, for example), or even other variations of PSO could be immediately used. This would require using a set of measures that are more meaningful for the specific type of dynamics that would be sonified. Hence, this points a possible direction for future work. Another improvement that must be explored is the user configurability, changing the typical timbres that compose the soundscapes. This way, a higher aesthetic appealing could be reached in the results, and different characteristics of the optimization process could be made more evident.

7. REFERENCES


