EUSIPCO 2013 1569744633

HYPERBOLIC PARTICLE SWARM OPTIMIZATION WITH APPLICATION IN RATIONAL IDENTIFICATION

Péter Kovács [a,*], *Serkan Kiranyaz* [b], *Moncef Gabbouj* [b]

[a] Eötvös L. University, Budapest, Hungary,
[b] Tampere University of Technology, Tampere, Finland

ABSTRACT

The rational function systems proved to be useful in several areas including system and control theories and signal processing. In this paper, we present an extension of the well-known particle swarm optimization (PSO) method based on the hyperbolic geometry. We applied this method on digital signals to determine the optimal parameters of the rational function systems. Our goal is to minimize the error between the approximation and the original signal while the poles of the system remain stable. Namely, we show that the presented algorithm is suitable to localize the same poles by using different initial conditions.

Index Terms— Rational functions, Malmquist–Takenaka system, Hyperbolic geometry, Particle swarm optimization.

1. INTRODUCTION

Rational function systems have a wide range of applications. For instance in system, control theories they play an important role in the representation of the transfer function, see e.g. [1], [2]. Furthermore, rational functions like the Blaschke functions and the orthonormal Malmquist–Takenaka (MT) or the biorthogonal systems [3] are effectively used for approximating signals especially electrocardiograms [4], [5]. In our former work, a general MATLAB library, called RAIT [6], has been implemented for manipulating different types of rational functions systems. We also built in methods for finding the best parameters of a good approximation automatically [7]. Generally speaking, the following parameters should be determined: the number, the positions and the multiplicities of the poles. The deterministic optimization search technique, Nelder–Mead simplex algorithm [8] was applied for this purpose. Although it produced good pole configurations, it turned out to be unstable. In other words it was sensitive to the initial conditions, i.e. the starting points of the simplex. As a result, running the algorithm more times on the same problem may result in different poles. This issue is well-known in the optimization theory, which is caused by being trapped in a local maxima or minima. In this work we will show that the particle swarm optimization (PSO) method is more appropriate for this problem.

The outline of this article is as follows. In Section 2 we summarize our former works considering the rational function approximations by using different types of systems. Section 3 contains the proposed hyperbolic extension of the PSO algorithm. Section 4 presents an application of the method on ECG signals. Section 5 is a summary of conclusions and future plans.

2. RELATED WORKS

In this section we outline the mathematical concepts behind our algorithms.

Let $\mathbb{C}$ stand for the set of complex numbers, $\mathbb{D} := \{ z \in \mathbb{C} : |z| < 1 \}$ for the open unit disc, $\mathbb{N} := \{ 1, 2, 3, \dots \}$ for the set of natural numbers, and $\mathbb{T} := \{ z \in \mathbb{C} : |z| = 1 \}$ for the unit circle (or torus).

The basic rational functions are defined as follows

$$r_{a,k}(z) = \frac{1}{(1-\overline{a}z)^k}\,, \qquad (a \in \mathbb{D}, k \in \mathbb{N})\,. \tag{1}$$

The parameter a is referred to as *inverse pole* (because $1/\overline{a}$ is a pole in the standard sense), k is said to be the *order* of the basic function. Using a terminology similar to the trigonometric case, the value $k = 1$ corresponds to the *fundamental tone* and $k > 1$ the *overtones*.

The basic rational functions form a linearly independent but not orthogonal set of functions. The corresponding orthonormal systems called Malmquist–Takenaka (MT) systems are generated by Gram–Schmidt orthogonalization to basic rational functions. The scalar product used on $\mathbb{T}$ is:

$$\langle F, G \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(e^{it}) \overline{G}(e^{it})\, \mathrm{d}t \quad (F, G \in H^2(\mathbb{D}))\,. \tag{2}$$

Naturally we use the uniformly sampled discrete approximation of this integral in our calculations.

Manuscript received March 17, 2013. The research was supported by the National Development Agency (NFÜ), Hungary (grant agreement Research and Technology Innovation Fund. EITKIC_12.)

* Corresponding address: Department of Numerical Analysis, Eötvös L. University, Budapest, 1117 Hungary (phone: 36-1-372-2500/8471; e-mail: kovika@inf.elte.hu).

A handy property of the MT systems is that their elements can be expressed as Blaschke products. Namely, taking the basic functions for a given $n \in \mathbb{N}$ and $a_1, \dots, a_n \in \mathbb{D}$ the orthogonalized MT system can be written as:

$$\Phi_k(z) = \frac{\sqrt{1-|a_k|^2}}{1-\overline{a_k}z} \prod_{j=1}^{k-1} B_{a_j}(z)\,, \tag{3}$$

with $1 \leq k \leq n$, where

$$B_{a_k}(z) := \frac{z-a_k}{1-\overline{a_k}z} \qquad (z \in \mathbb{C} \setminus \{\, 1/\overline{a_k} \,\})\,. \tag{4}$$

Usually we take the restriction of the functions on the unit circle, which we identify with the $[-\pi, \pi)$ real interval, in our algorithm. To this order we apply the map $t \mapsto e^{it} \in \mathbb{T}$. Then the real part and the imaginary part can both be used for signal processing purposes. More precisely, any signal f can be approximated by taking the real part of the linear combinations of the Φ_k functions.

This method was successfully applied for representing ECG signals [5]. Furthermore, we used the basic rational functions to model the QRS complex of the electrocardiograms [4]. We note that this process is highly adaptive, in contrast with other types of orthogonal systems, for instance the trigonometric functions [9]. Additionally, other types of rational functions were also developed such as the biorthogonal systems [3] to improve the localization property in time.

If the poles and multiplicities are given then the MT system is determined. Let us consider the different poles $a_1, \dots, a_n \in \mathbb{D}$ and their related multiplicities $m_1, \dots, m_n \in \mathbb{N}$. In this case the approximation has $m := m_1 + m_2 + \cdots + m_n$ number of c_k coefficients. In our former work [6], a priory information was used to predict the number and the multiplicities of the poles. For instance, in the case of an ECG curve, the number of diagnostic (P,QRS,T) waves were used as an estimation of the number of the poles. Furthermore, the lobes P and T are less significant than the QRS complex which makes it evident to use the $[1, 1, 2]$ vector respectively for their multiplicities.

Now, our aim is to construct an MT system by using the best parameters to minimize the approximation error. To achieve this goal we perform the following steps:

- define appropriate operators for vector scalar multiplication, and vector addition in the Poincaré model;
- extend the basic PSO algorithm for localizing the best position of a single pole by using hyperbolic geometry;
- extend the hyperbolic single pole PSO to solve multipole problems as well.

3. HYPERBOLIC PSO ALGORITHM

The PSO algorithm was introduced by Eberhart and Kennedy [10] as a population based stochastic optimization technique. The method was inspired by the social behavior of bird flocking or fish schooling. The algorithm works similar to a swarm, which is flying through the (problem) space while they are looking for an optimal point (e.g. food). During the search process the swarm is navigated by the closest particle to the optima. The method is initialized with a random population where each individual has a solution for the given problem. In this sense, it shows similarities to the Evolutionary Algorithms (EAs) such as Genetic Algorithm (GA). Generally, it can also be considered as a special case of the Monte Carlo simulation where the random population is led by certain points at each iteration.

Namely, each individual keeps track of its position in an n dimensional search space related to the personal best solution so far achieved. Using this terminology the global best coordinates of the swarm can be determined in a similar way. These properties along with the velocities v_k and the positions x_k of the kth individual will be updated in the ith iteration by using the following equations:

$$\begin{aligned} v_k(i+1) =& w(i) \cdot v_k(i) + c_1 r_1(i) \cdot (y_k(i) - x_k(i)) \\ &+ c_2 r_2(i) \cdot (\widehat{y}_k(i) - x_k(i))\,, \\ x_k(i) =& x_k(i) + v_k(i)\,, \end{aligned} \tag{5}$$

where

- $y_k(i) \in \mathbb{R}^n$ is the personal best position of the kth particle at the ith iteration,
- $\widehat{y}_k(i) \in \mathbb{R}^n$ is the global best position of the kth particle at the ith iteration,
- c_1, c_2 are the learning factors, usually set as, $c_1 = c_2 = 2$,
- $w(i)$ is the inertia weight which is linearly decreased from 0.8 to 0.2,
- $r_1(i), r_2(i) \in (0, 1)$ are uniformly distributed random numbers.

If we do not permit arbitrary large jumps in the search space then it is possible to restrict the velocities and the positions to a certain interval defined by the variables, $V_{max}, X_{min}, X_{max}$. More details can be found in [11].

For the sake of simplicity we are considering the two dimensional case at first, where we need to find only one complex pole while its multiplicity is fixed. So, the remaining task is to find the position of this single pole for an MT system which minimizes the error of the approximation. As we know from Section 2 the poles of a rational function system should lie within the unit circle. This implies the idea to use the Poincaré model on $\mathbb{D}$ of the hyperbolic geometry. In this case, the search space should be the open unit disc, and the particles are containing two coordinates related to the real and imaginary part of the pole. Namely, the algorithm will be returned with $a_1 := x_{k,1} + i x_{k,2}$.

If we want to apply the basic PSO algorithm with the Poincaré model then the $+$ and $\cdot$ operators in Eq. (5) should be replaced by hyperbolic multiplication and addition.

3.1. Hyperbolic multiplication

Using the terminology of the Euclidian geometry the vector scalar multiplication of the hyperbolic space can be defined in a similar way. Namely, it means scaling of a hyperbolic vector by keeping its direction. In this case, the geodesics of this space are represented by arcs of circles that are orthogonal to the torus. Furthermore, it can be shown that the hyperbolic segments can be defined by using the Blaschke functions.

Let us denote the Blaschke transformation $\epsilon B_a(t)$ by $\mathcal{B}_\mathfrak{a}(t)$ where $\mathfrak{a} := (a, \epsilon) \in \mathbb{D} \times \mathbb{T}$. Then the function $\mathcal{B}_\mathfrak{a}(t)$ maps the interval $[0, p]$ onto the segment connecting w_1, and w_2, where

$$p = |B_{w_1}(w_2)| \,, \quad \epsilon = \frac{B_{w_1}(w_2)}{|B_{w_1}(w_2)|} \,, \quad a = -\bar{\epsilon} w_1 \,. \tag{6}$$

Now the hyperbolic vector $\overrightarrow{w_1 w_2}$ can be defined as a directed segment with $\mathcal{B}_\mathfrak{a}(0) = w_1$ and $\mathcal{B}_\mathfrak{a}(p) = w_2$. For the scaling of these vectors we will use the so-called pseudo-hyperbolic metric on $\mathbb{D}$:

$$\rho_0(z_1, z_2) := \frac{|z_1 - z_2|}{|1 - \bar{z}_1 z_2|} = |B_{z_1}(z_2)| \quad (z_1, z_2 \in \mathbb{D}) \,. \tag{7}$$

Then the hyperbolic metric is as follows:

$$\rho(z_1, z_2) := \operatorname{arth}(\rho_0(z_1, z_2)) \qquad (z_1, z_2 \in \mathbb{D}) \,. \tag{8}$$

$(\mathbb{D}, \rho)$ is a complete metric space invariant with respect to the Blaschke transforms, i.e. for any $\mathfrak{a} \in \mathbb{D} \times \mathbb{T}$ and $z_1, z_2 \in \mathbb{D}$,

$$\rho(\mathcal{B}_\mathfrak{a}(z_1), \mathcal{B}_\mathfrak{a}(z_2)) = \rho(z_1, z_2) \,. \tag{9}$$

Let us denote the hyperbolic vector scalar multiplication by $\odot$. Then the expression $\lambda \odot \overrightarrow{w_1 w_2}$ means the scaling of $\overrightarrow{w_1 w_2}$ by the factor $\lambda \in \mathbb{R}$. We should calculate the new endpoint w_λ on the geodesic connecting w_1 and w_2 that satisfies the following equation

$$\rho(w_1, w_\lambda) = \lambda \rho(w_1, w_2) \,. \tag{10}$$

By using the parameters of Eq. (6) for the segment $\overrightarrow{w_1 w_2}$ and the invariant property of the Blaschke transform from Eq. (9) this can be written as,

$$\operatorname{arth}(s_\lambda) = \rho(0, s_\lambda) = \lambda \rho(0, p) = \lambda \operatorname{arth}(p) \,, \tag{11}$$

where $w_\lambda = \mathcal{B}_\mathfrak{a}(s_\lambda)$. Hence

$$s_\lambda = \operatorname{th}(\lambda \operatorname{arth}(p)) \,. \tag{12}$$

In summary, the expression $\overrightarrow{w_1 w_\lambda} = \lambda \odot \overrightarrow{w_1 w_2}$ can be evaluated by using the equations (6), (12), and $w_\lambda = \mathcal{B}_\mathfrak{a}(s_\lambda)$. Fig. 1(a) shows this procedure. The dashed blue lines represent the original vectors while the scaled ones are marked by red color. Furthermore, the pseudocode of this algorithm is as follows.

Algorithm 1 hyperbolic multiplication $\odot$

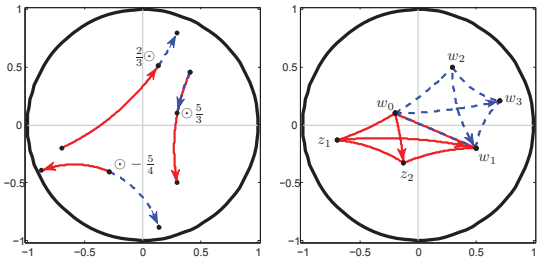
```
function HYP_MUL(λ,w1,w2)
    if w1 = w2 then
        sλ := 0
        wλ := w1
    else
        p := |Bw1(w2)|
        ϵ := Bw1(w2)/ |Bw1(w2)|
        a := −ϵ̄w1
        sλ := th(λ arth(p))
        wλ := ϵBa(sλ)
    end if
    return wλ
end function
```

3.2. Hyperbolic addition

According to the previous section the natural approach for defining hyperbolic addition would be a proper interpretation of the parallelogram rule. In this sense the difference of two vectors $\overrightarrow{w_0 w_1}$ and $\overrightarrow{w_0 w_2}$ will be the diagonal that connects the endpoints. The other diagonal represents the sum of these vectors. The hyperbolic parallelogram can be easily constructed by reflecting the initial point w_0 with respect to the bisection of the segment $\overline{w_1 w_2}$. Additionally, the required construction of hyperbolic bisectors and reflections can be found in [4].

Unfortunately, the parallelogram method cannot be used since the difference of two vectors $\overrightarrow{w_0 w_1} - \overrightarrow{w_0 w_2}$ is usually not equal with the addition in opposite direction $\overrightarrow{w_0 w_1} + (-\overrightarrow{w_0 w_2})$. One can see an example on Fig. 1(b). Here, z_1 denotes the reflection of w_2 onto w_0. Then the difference of the vectors were constructed as $\overrightarrow{w_0 z_2} = \overrightarrow{w_0 w_1} + \overrightarrow{w_0 z_1}$. As we mentioned that, it is not equal to the vector $\overrightarrow{w_2 w_1}$. For instance, their magnitudes on Fig. 1(b) are 0.92 and 0.46 with respect to the hyperbolic metric in Eq. (8). The reason behind this phenomena is that the similarity and the congruency are equivalent definitions in the hyperbolic space. For instance, two triangles are similar or congruent only if their angles are equal. It means that scaling an object without distortion is not possible in the hyperbolic space. In other words the hyperbolic geometry has an absolute unit of length.

The proper definition of hyperbolic addition realizes on compositions of Blaschke functions. It can be shown that the set of Blaschke transforms $\mathfrak{B} = \{B_\mathfrak{a} : \mathfrak{a} \in \mathbb{D} \times \mathbb{T}\}$ is closed for the composition operator $\circ$: $\mathcal{B}_{\mathfrak{a}_1} \circ \mathcal{B}_{\mathfrak{a}_2} = \mathcal{B}_\mathfrak{a}$. If $\mathfrak{a}_1 = (w_1, 1)$ and $\mathfrak{a}_2 = (w_2, 1)$ then $\mathfrak{a} = (w, 1)$ where $w = \dfrac{w_1 + w_2}{1 + w_1 \overline{w_2}}$. This is the well-known Möbius transformation which maps the unit disc onto itself. This can be interpreted as a vector addition in the hyperbolic space for vectors with initial point at zero. We denote this operator by $\oplus : (\overrightarrow{0w_1}, \overrightarrow{0w_2}) \longmapsto \overrightarrow{0w}$. For details on these transformations we refer to [9].



(a) Hyperbolic scaling. (b) Hyperbolic addition.

Fig. 1. Hyperbolic operators.

3.3. Hyperbolic PSO with multiple poles

In the last section, the hyperbolic PSO (HPSO) algorithm was defined by replacing the operators $+, \cdot$ in Eq. (5) with $\oplus, \odot$. However, we considered only the problem of single pole optimization in the Poincaré disc model, it is easy to extend it into higher dimensions. Let's denote the sequence of different poles by $\mathbf{a} = a_1, \ldots, a_n$. Then the optimization problem for a given f function or signal can be written as

$$\arg\min_{\mathbf{a}} \|f - S^m_{\mathbf{a}} f\|_2 \,, \tag{13}$$

where $S_{\mathbf{a}} f$ is the MT series expansion of f

$$S^m_{\mathbf{a}} f = \sum_{k=0}^{m-1} \langle f, \Phi_k \rangle \, \Phi_k \,. \tag{14}$$

Describing rigid movements of the hyperbolic space is a hard task in high (> 3) dimensions. So, our geometrical approach cannot be applied directly to extend the HPSO algorithm on multi-pole problems. By this reason we perform the optimization separately on each pole. It means that we should replace Eq. (13) by

$$\arg\min_{a_i} \|f - S^m_{\mathbf{a}} f\|_2 \quad (i = 1, \ldots, n) \,. \tag{15}$$

The original multi-pole problem was separated into single pole optimizations. So, the two dimensional HPSO method can be applied successively. However, the equation should be evaluated n times, it represents only one step of the algorithm. In addition, one can proceed with the optimization by iterating Eq. (15) over every poles sequentially until a condition is not satisfied. For instance, the algorithm can be terminated when a certain level of approximation error is achieved or the locations of the poles are not changed enough between the last iterations.

We note that, the particles cannot leave the unit circle during the algorithm. It is coming naturally from our model which makes the search space boundaries, X_{min}, X_{max} unnecessary. However, a maximal velocity can be also defined by measuring the distance of the movements using Eq. (8).

4. EXPERIMENTAL RESULTS

In this section, we perform some experiments on the hyperbolic PSO algorithm to test its efficiency with various aspects. Our goal is to compare this method with an other optimization technique the well-known Nelder–Mead algorithm. It was used in our former work [6] to find the best poles of the system. However, recall that these poles were strongly dependent on the initial conditions. We tested the stability of the poles and the approximation error in the case of using Nelder–Mead and HPSO algorithms.

In the first experiment, we generated 400 signals with random poles and coefficients. Then both algorithms were run 100 times on each synthesized signal. One can see an example on Fig. 2(a) and Fig. 2(b) which show the resulted poles on the same record. The original ones are marked by black squares. Moreover, Fig. 2(c) displays the average error of every 100 run on each signal. One can see that the stability of the poles is about two times better in the case of the HPSO algorithm.

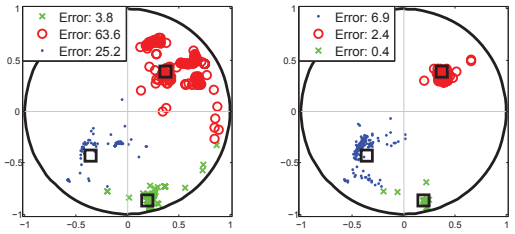
In the second experiment we tested the approximation error of both optimization method. For this purpose, we used the MIT-BIH Arrhythmia Database from PhysioNet [12]. Namely, the first 3 minute of each ECG record was segmented into heart beats. Then these beats were approximated by using HPSO and Nelder-Mead algorithms (overall run> 9000). Based on our former research on electrocardiographs [5] we chose 3 poles with multiplicities $m_1 = m_2 = 2, m_3 = 4$. Fig. 2(d) shows the average error of each ECG record where the approximation error was measured by using the percentage root mean square difference (PRD):

$$\mathrm{PRD} = \sqrt{\frac{\sum_{n=1}^{N}(x(n) - \widetilde{x}(n))^2}{\sum_{n=1}^{N}(x(n) - \overline{x})^2}} \times 100 \,, \tag{16}$$

where $\overline{x}$ is the signal mean, $x(n), \widetilde{x}(n)$ denote the synthesized and the reconstructed signals and N is the number of samples. However, the HPSO is slightly better than the Nelder-Mead algorithm, but the difference is not significant. On the other hand, it means that the poles that was found by our method are more stable while the approximation keeps its accuracy. By taking advantage of this property it is possible to use the poles as a feature, for instance in ECG beat classification.

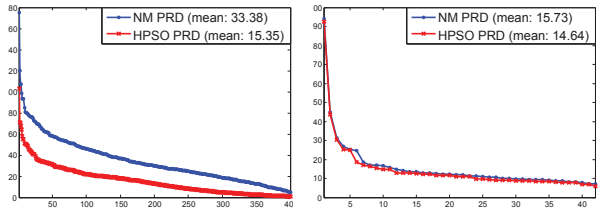
5. CONCLUSIONS

We have extended the basic PSO algorithm to hyperbolic spaces by using the Poincaré disc model. The particles of this method move inside the unit circle, which does not require additional conditions like X_{min} and X_{max}. The algorithm is adequate to construct good approximations with high pole stability. Furthermore, we showed that it outperforms the Nelder–Mead algorithm in every sense, except for the execution time which was about 10 times slower. It is a consequence of evaluating the approximation on a large population



(a) Poles coming from 100 runs of the Nelder-Mead method.

(b) Poles coming from 100 runs of the HPSO method.



(c) Average PRDs of the resulted poles for synthetic signals.

(d) Average PRDs of the real ECG approximations.

Fig. 2. Experimental results on synthetic and real datas.

(> 20), but this can be improved by using parallel implementations. A number of applications and simple extensions of the model are possible.

- The algorithm can be extended to any type of simply connected region by using the conformal mappings.
- Further research is required to apply methods to avoid premature convergence such as the fractional global best formations [13].
- Additional improvements can be achieved by combining our method with the multi-dimensional (MD) PSO [14].
- Now, the stabilized poles can be used as a feature in heart beat classification.

In the future we plan to examine the properties of the MD PSO algorithm in rational function approximation. We assume that it can be easily adapted to the pole optimization problem by applying the same analogy of [14]. In this sense, a good approximation can be acquired without using any type of priory information. Furthermore, it is possible to determine not just the positions of the best poles, but multiplicities and the number of different poles as well.

6. REFERENCES

[1] J. Bokor and F. Schipp, "Approximate linear $\mathbb{H}^\infty$ identication in laguerre and kautz basis," in *AUTOMATICA*. IFAC, 1998, vol. 34, pp. 463–468.

[2] P. S. C. Heuberger, P. M. J. Van den Hof, and B. Wahlbeg, *Modelling and Identification with Rational Orthogonal Basis Functions*, pp. 61–102, Springer-Verlag, London, 2005.

[3] S. Fridli and F. Schipp, "Biorthogonal systems to rational functions," *Annales Univ. Sci. Budapest., Sect. Comp*, vol. 35, pp. 95–105, 2011.

[4] S. Fridli, P. Kovács, L. Lócsi, and F. Schipp, "Rational modeling of multi-lead QRS complexes in ECG signals," *Annales Univ. Sci. Budapest., Sect. Comp*, vol. 37, pp. 145–155, 2012.

[5] S. Fridli, L. Lócsi, and F. Schipp, "Rational function system in ECG processing," in *Proc. 13th Int. Conf. Computer Aided Systems Theory–EUROCAST 2011, Part I (eds. R. Moreno-Díaz et al.)*. LNCS 6927, 2011, pp. 88–95.

[6] P. Kovács and L. Lócsi, "RAIT: The rational approximation and interpolation toolbox for MATLAB," in *35th International Conference on Telecommunications and Signal Processing (TSP)*, 2012, pp. 671–677.

[7] L. Lócsi, "Approximating poles of complex rational functions," *Acta Univ. Sapientiae, Mathematica*, vol. 1, no. 2, pp. 169–182, 2009.

[8] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[9] P. Henrici, *Applied and Computational Complex Analysis*, vol. 1-3, John Wiley & Sons, Inc., Canada, 1974.

[10] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948.

[11] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *The IEEE International Conference on Evolutionary Computation Proceedings*, 1998, pp. 69–73.

[12] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. 215–220, 2000.

[13] S. Kiranyaz, J. Pulkkinen, A. Yildirim, and M. Gabbouj, "Multi-dimensional particle swarm optimization in dynamic environments," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2212–2223, 2011.

[14] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, "Evolutionary artificial neural networks by multi-dimensional particle swarm optimization," *Neural Networks*, vol. 22, no. 10, pp. 1448–1462, 2009.