Particle Swarm Optimization and Quantum Particle Swarm Optimization to Multidimensional Function Approximation

Diogo Silva, Fadul Rodor, Carlos Moraes

Abstract—This work compares the results of multidimensional function approximation using two algorithms: the classical Particle Swarm Optimization (PSO) and the Quantum Particle Swarm Optimization (QPSO). These algorithms were both tested on three functions - The Rosenbrock, the Rastrigin, and the sphere functions - with different characteristics by increasing their number of dimensions. As a result, this study shows that the higher the function space, i.e. the larger the function dimension, the more evident the advantages of using the QPSO method compared to the PSO method in terms of performance and number of necessary iterations to reach the stop criterion.

Keywords—PSO, QPSO, function approximation, AI, optimization, multidimensional functions.

I. INTRODUCTION

FUNCTION approximation is an artifice that can be used to solve two basic types of problems: to obtain a simpler function that can represent the original one and/or to find and fit the best function to empirically obtained data.

Studies about the optimization of function approximation have been developed since the 1950s, as the work initiated by [1] in which they proposed a method for defining an unknown function through data. Works related to Artificial Evolution also began appearing in the 1950s with [2]. The first algorithm conceived through evolutionary strategies was proposed by [3]. Based on [3], Fogel et al. [4] proposed a method of evolutionary programming and discussed it and its approximation with simulated evolution.

In the 1980s several techniques emerged, such as Simulated Annealing (SA) [5] and the fundamentals of Integer Programming and the Tabu Search [6], when the expression “metaheuristic search techniques” was introduced and defined as “general methodologies at a higher level of abstraction capable of guiding the modeling of solving optimization problems”.

Metaheuristics are typically inspired by behaviors observed in nature [7]. In the 1990s, works inspired by the observation of ant colonies, swarms of bees and some other kinds of nature behavior appeared. Several techniques have been developed, such as Genetic Algorithms [8], Ant Colony Optimization [9] and Particle Swarm Optimization (PSO) [10].

In this work, the techniques of PSO and Quantum Particle Swarm Optimization (QPSO) were used for function approximation. The performance and number of iterations were analyzed for each method when applied to different types of functions and dimensions in order to compare the two techniques.

Besides this first introductory section, this paper is organized as follows: section II shows a brief about PSO and QPSO algorithms, section III presents the results and comparison between the two algorithms when applied to three different kinds of functions, and Section IV approaches the conclusions of this study.

II. OPTIMIZATION ALGORITHMS

Adaptive and evolutionary optimization techniques have several advantages over some of the exact approaches. One advantage is that the techniques can deal with a large number of problem parameters and no rigid assumptions about the problem is necessary [11].

There are many techniques based on natural behavior, such as Genetic Algorithms, PSO, and the Shuffled Frog Leaping algorithms. These techniques can be used in several areas as climatology [12], control [13], [14], finance [15], acoustic [16], and power electronics [17].

In this work two different optimization methodologies were compared: The classical PSO and the QPSO, these two techniques are quite widespread for solving optimization and functions approximation problems.

A. PSO

The PSO algorithm was originally proposed by [10]. The swarm of particles is inspired by social behavior, observed in flocks of birds or shoals of fish. Each individual of a population has their own experience and is able to estimate the quality of that experience. Because individuals are social, they also have knowledge about how their neighbors behave. These two types of information correspond to individual (cognitive) learning and cultural (social) transmission, respectively. Therefore, the probability that a particular individual makes a certain decision will be a function of their performance in the past and the performance of some of their neighbors [18].

In the PSO each individual of the population is represented by a point, called a particle, these individuals move in a search space \( \mathbb{R}^n \), where \( n \) is the dimension of that space. Each point has a number of attributes, any change of these attributes...
In quantum mechanics, by the uncertainty principle it is not possible to determine the position $x_i$ and the velocity $v_i$ of the particle. In the QPSO the particles have the wave behavior, being governed by the function $\psi(x,t)$, defined in terms of iterations by

$$\psi(Y_{i,n+1}^j) = \frac{1}{\sqrt{L_{i,n}^j}} \exp\left(-\frac{Y_{i,n+1}^j}{L_{i,n}^j}\right), \quad (3)$$

where $Y_{i,n+1}^j = |X_{i,n+1}^j - p_{i,n}^j|$ with $p_{i,n}^j$ the stochastic attractor of particle from the classical PSO and $L_{i,n}^j$ the characteristic legth of wave function. By the definition of wave function, the probability distribution function is

$$F(Y_{i,n+1}^j) = 1 - \exp\left(-\frac{2Y_{i,n+1}^j}{L_{i,n}^j}\right), \quad (4)$$

The QPSO presents some advantages over classical PSO, in a quantum system the number of states is greater than in a linear system. Moreover, by the uncertainty principle, the particle can appear anywhere in the solution space, according to the probability distribution.

As the PSO algorithm, QPSO still widely used for example in the areas of image processing [28], the energy market [29] and power systems [30].

### III. PSO AND QPSO RESULTS

In this work, simulations were performed comparing the performance and number of iterations necessary for the convergence of the PSO and QPSO algorithms. The functions of Rosenbrock, Rastrigin and the Sphere function were used for performance evaluation. In all cases, the algorithms have been tested in $\mathbb{R}^2$, $\mathbb{R}^3$ and $\mathbb{R}^{10}$ space.

#### A. Rosenbrock Function

The Rosenbrock function is a non-convex function used as a performance test for optimization problems. This function has only a global minimum, at the point (1,1) for two dimensions. The function is defined by (5)

$$f(x,y) = (a - x)^2 + b(y - x^2)^2 \quad (5)$$

Fig. 2 represents the two-dimensional Rosenbrock function in 3D.

The PSO and QPSO algorithms were parameterized to find the minimum of the Rosenbrock function. The same parameters were maintained for the two algorithms. The parameters are shown in Table I.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>PARAMETERS OF THE PSO AND QPSO ALGORITHMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of particles to be optimized</td>
<td>100</td>
</tr>
<tr>
<td>Maximum number of steps in the algorithm</td>
<td>5000</td>
</tr>
<tr>
<td>Social Parameter</td>
<td>2</td>
</tr>
<tr>
<td>Cognitive Parameter</td>
<td>2</td>
</tr>
<tr>
<td>Stop Criterion (Error)</td>
<td>$10^{-5}$</td>
</tr>
</tbody>
</table>

Initially, a simulation was performed maintaining all the parameters listed in Table I for the PSO and QPSO algorithms.
The first case analyzes the responses of algorithms for Rosenbrock function in $\mathbb{R}^2$ space, the results are shown in Figs. 3 and 4.

In Fig. 3 it is possible to observe a greater spread of the particles in the case of PSO. The average particle is fast approaching the optimal solution, but the algorithm takes more iterations to reach the stopping criterion. Fig. 4 is noted that in QPSO particles are more concentrated and the stop criterion is reached sooner.

The PSO required 1229 iterations for convergence, while the QPSO required only 129. Although there was a significant difference in computational effort between the two algorithms, the two reached the stopping criterion. The PSO found the value $9.35 \cdot 10^{-6}$ and the QPSO is $4.11 \cdot 10^{-6}$. The closer to zero, the better the performance of the algorithm, it is observed that the QPSO obtained better performance.

The Rosenbrock function can also be set to more than two dimensions. Its representation for the space $\mathbb{R}^N$ is shown in (6).

$$
\sum_{i=1}^{N-1} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2
$$

with $x = [x_1, \ldots, x_N] \in \mathbb{R}^N$.

A second simulation was performed by increasing the dimension of the Rosenbrock function to five, the results are shown in Figs. 5 and 6.

In the case of the function defined in $\mathbb{R}^5$, Fig. 5 shows there is still a greater spread of the particles in the use of the PSO, but for a higher dimension the QPSO approaches the optimal solution faster, as can be seen in Fig. 6.

Both algorithms did not reach the stopping criterion before 5000 iterations, but it can be observed that the convergence tendency of the QPSO is much faster than the PSO and the QPSO algorithm obtained a better performance, reaching 0.0047, compared to 0.0111 of PSO.

The complexity of the model was increased, raising the function to the tenth dimension. The results are shown in Figs. 7 and 8.

For the ten-dimensional function, it is possible to notice a divergence in the response of the PSO in Fig. 7, which does not occur for the QPSO whose number of iterations necessary to approximate the optimal response is smaller, as shown in Fig. 8.

For the ten-dimensional Rosenbrock function the performance of the QPSO becomes evident. With five thousand iterations the QPSO found the value 0.1696 while the PSO 3.4607. Making it clear the best performance and...
Fig. 6 QPSO response to the five-dimensional Rosenbrock function

Fig. 7 PSO response to the ten-dimensional Rosenbrock function

Fig. 8 QPSO response to the ten-dimensional Rosenbrock function

Fig. 9 Two-dimensional Rastrigin function in 3D

B. Rastrigin Function

The Rastrigin function is a non-convex function used as a performance test problem for optimization algorithms. It was proposed by Rastrigin [31] as a 2-dimensional function and generalized by Muhlenbein et al. [32]. Finding the minimum of this function is a difficult problem due to its large search space and its large number of local minima.

Fig. 9 represents the two-dimensional Rastrigin function in 3D.

The Rastrigin function is defined in two dimensions but can also be set to more than two. Its representation for the space $\mathbb{R}^N$ is shown in (7).

$$\sum_{i=1}^{N-1} (10 + x_i^2 - 10 \cdot \cos (2 \cdot \pi \cdot x_i))$$

with $x = [x_1, ..., x_N] \in \mathbb{R}^N$.

The PSO and QPSO algorithms were parameterized to find the minimum of the Rastrigin function. The same parameters were maintained for the two algorithms. The parameters are shown in Table II.

<table>
<thead>
<tr>
<th>Parameters of the PSO and QPSO Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of particles to be optimized</td>
</tr>
<tr>
<td>Maximum number of steps in the algorithm</td>
</tr>
<tr>
<td>Social Parameter</td>
</tr>
<tr>
<td>Cognitive Parameter</td>
</tr>
<tr>
<td>Stop Criterion (Error)</td>
</tr>
</tbody>
</table>

The first simulation performed analyzes the responses of algorithms for the Rastrigin function in $\mathbb{R}^2$ space, the results are shown in Figs. 10 and 11.

In the case of the Rastrigin function defined in $\mathbb{R}^2$, it is possible to notice in Fig. 10 the quick approximation of the optimal solution of the PSO algorithm, but it needs a high number of iterations to reach the stop criterion.

The PSO required 3174 iterations for convergence, while the QPSO required 75. The PSO found the value $1.43 \cdot 10^{-11}$ and
the QPSO is \(1.28 \cdot 10^{-12}\). A second simulation was performed by increasing the dimension of the Rastrigin function to five, the results are shown in Figs. 12 and 13.

As in the example of the Rosenbrock function defined in \(\mathbb{R}^5\), the QPSO algorithm requires a smaller number of iterations to approximate the optimal solution for the Rastrigin function, this result it’s observed in Figs. 12 and 13.

The PSO algorithm did not reach the stopping criterion before 5000 iterations and as result reached 0.995. The QPSO algorithm just needed 755 iterations to reach \(7.10 \cdot 10^{-11}\). The results for \(\mathbb{R}^{10}\) are shown in Figs. 14 and 15.

For the function defined in \(\mathbb{R}^{10}\), the number of iterations to approximate the solution is smaller, as observed in Fig. 15.

For the ten-dimensional Rastrigin function both algorithms did not reached the stopping criterion. With 5 thousand iterations the QPSO found the value 0.0025 while the PSO 4.97. Making it clear the best performance and speed of convergence of the QPSO algorithm.
C. Sphere Function

The Sphere function is a function frequently used as a performance test problem for optimization algorithms. The Sphere function has no local minima except for the global one. It is continuous, convex and unimodal.

Fig. 16 represents the two-dimensional Sphere function in 3D.

The Sphere function can also be set to more than two dimensions. Its representation for the space $\mathbb{R}^N$ is shown in (8).

$$f(x, y) = \sum_{i=1}^{n} (x_i^2)$$

(8)

with $x = [x_1, ..., x_N] \in \mathbb{R}^N$.

The PSO and QPSO algorithms were parameterized to find the minimum of the Sphere function. The same parameters were maintained for the two algorithms. The parameters are shown in Table III.

Initially, a simulation was performed maintaining all the parameters listed in Table III for the PSO and QPSO algorithms. The first case analyzes the responses of algorithms for the Sphere function in $\mathbb{R}^2$ space, the results are shown in Figs. 17 and 18.

Despite the PSO algorithm reach the stopping criterion, it needs a high number of iterations when compared to the QPSO algorithm, as observed in Figs. 17 and 18.

The PSO required 2925 iterations for convergence, while the QPSO required only 20. The PSO found the value $6.58 \cdot 10^{-13}$ and the QPSO is $3.53 \cdot 10^{-15}$.

A second simulation was performed by increasing the dimension of the Sphere function to five, the results are shown
The PSO algorithm reached the stopping criterion with 4062 iterations and the QPSO just needed 205. PSO reached $8.07 \times 10^{-13}$ and QPSO $6.32 \times 10^{-13}$. The complexity of the model was increased, raising the function to the tenth dimension. The results are shown in Figs. 21 and 22.

As in the case of the other functions, for a higher dimension the QPSO algorithm obtained an approximation of the optimal solution with a smaller number of iterations for the case of the sphere function as observed in Figs. 21 and 22.

For the ten-dimensional Sphere function the better performance of the QPSO becomes evident. With 842 iterations the QPSO found the value $2.83 \times 10^{-13}$ while the PSO needed 4575 iterations to reach $7.78 \times 10^{-13}$, making it the best scenario in terms of performance and speed of convergence for the QPSO algorithm.

**IV. CONCLUSION**

This article presented a comparison between two optimization algorithms, the PSO and the QPSO. Three types of functions were chosen for comparing the optimization algorithms because of their different characteristics. The Rosenbrock function, which has only a global minimum and some local minimums, the Rastrigin function, which is widely used in optimization problems because it contains a high number of local minimum, and the sphere function, which has a single global minimum.

For all three tested functions the optimization by QPSO presented advantages against the classical PSO, especially when the function was optimized for a larger space dimensions. In all cases, the QPSO algorithm required fewer iterations to achieve the stopping criterion or got closer to it when the maximum number of iterations was reached, and also the QPSO presented a smaller approximation error when compared to the classic PSO algorithm.

**ACKNOWLEDGMENT**

The authors would like to thank CAPES for the financial support provided to those involved in this work. And also the Federal University of Itajubá and the Computer Science Post Graduation Program.
REFERENCES


Diogo Silva
Graduated in Control and Automation Engineering (2010) from the Federal University of Itajubá and a Masters in Electrical Engineering (2013) from the same institution. He concluded his PhD in Electrical Engineering by the Federal University of Itajubá, having completed a sandwich period at the University of Tennessee in the USA. He is currently in the postdoctoral program in Computer Science at the Federal University of Itajubá. His main areas of interest are: Control systems, automation and process control, artificial intelligence, digital systems and electronic circuits.

Fadul Rodor
Graduated in Control and Automation Engineering from the Federal University of Itajubá (2009) and master’s degree in Electrical Engineering from the Federal University of Itajubá (2012). He is currently an Assistant Professor at the Federal University of Itajubá. Has experience in the area of Electrical Engineering, with emphasis in Industrial Electronics, Systems and Electronic Controls.
Carlos Moraes Graduated in Electrical Engineering from the Federal University of Itajubá (2001), master’s degree in Electrical Engineering from the Federal University of Itajubá (2002) and PhD in Electrical Engineering from the Federal University of Itajubá (2006). He is currently an associate professor at the Federal University of Itajubá. He teaches in the course of control and automation engineering in the areas of programming, embedded systems, digital electronics, power electronics, logic circuits, linear circuits, microcontrollers and artificial intelligence. He works in the themes: intelligent control, computer vision, intelligent embedded systems, signal processing, autonomous navigation and robotics.