World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:12, No:5, 2018

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].

Diogo Silva and Carlos Moraes are with the Department Systems Engineering and Information Technology Institute, Federal University of Itajubá, Itajubá, 37500-903, Brazil (e-mail: diogoleonardof@unifei.edu.br, valerio@unifei.edu.br).

Fadul Rodor is with the Federal University of Itajubá, Campus Itabira, Itabira, MG 35903-087, Brazil (e-mail: fadulrodor@unifei.edu.br).

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 eletronics [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

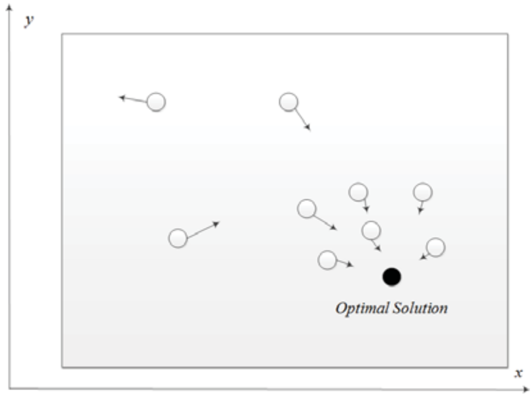

Fig. 1 PSO serching the optimal solution

makes the particle moves in space. The main idea is that the particles move towards one another and influence each other.

The PSO is based on two metrics. The first one, defined as g_B, is related to the behavior of all elements of the population, so each of the particles is influenced by the movement of the whole group. The second metric, defined as p_B, creates a neighborhood between the particle in question and its neighboring particles. The two metrics are measured by a fitness function that corresponds to the optimal solution of the problem.

The direction of motion of a particle p_i is a function of the current position x_i of this particle, the velocity of this particle $v_i(t+1)$, the position in which the particle found its best performance thus far (p_B), and the best overall performance of the system so far (g_B). The speed of the particle is defined by

$$v_i(t+1) = v_i(t) + \varphi_1 \times (p_B - x_i(t)) + \varphi_2 \times (g_B - x_i(t)) \quad (1)$$

In which φ_1 and φ_2 are constants defined by [10] as the "cognitive" and "social" components, respectively.

The position of each particle is defined as the sum of the influence of the previous position and the calculated speed, being defined by

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

The algorithm is repeated in loopd until a stop criterion is reached or the change rate of the particles is close to zero. A simplified representation of individuals seeking the optimal solution on the PSO is shown in Fig. 1.

The PSO continues to be widely used in a wide range of areas, including control [19]; power systems [20]; photovoltaic systems [21], [22]; image processing [23]; vehicle routing [24] and computer science [25], [26].

B. QPSO

The quantum PSO was introduced by [27] inspired by fundamentals of physics and quantum mechanics. In classical mechanics, used to define the PSO, a particle has its position vector x_i and the velocity vector v_i defined. These vectors determine the trajectory of this particle. The concept of trajectory exists in Newtonian mechanics, but it is not defined in quantum mechanics.

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

$$\psi(Y^j_{i,n+1}) = \frac{1}{\sqrt{L^j_{i,n}}} exp(\frac{-Y^j_{i,n+1}}{L^j_{i,n}}), \quad (3)$$

where $Y^j_{i,n+1} = |X^j_{i,n+1} - p^j_{i,n}|$ with $p^j_{i,n}$ the stochastic attractor of particle from the classical PSO and $L^j_{i,n}$ the characteristic leght of que wave function. By the definition of wave function, the probability distribution function is

$$F(Y^j_{i,n+1}) = 1 - exp(\frac{-2Y^j_{i,n+1}}{L^j_{i,n}}). \quad (4)$$

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 search 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 $\mathbb{R}^2$, $\mathbb{R}^5$ 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 I
PARAMETERS OF THE PSO AND QPSO ALGORITHMS

Number of particles to be optimized	100
Maximum number of steps in the algorithm	5000
Social Parameter	2
Cognitive Parameter	2
Stop Criterion (Error)	10^{-5}

Initially, a simulation was performed maintaining all the parameters listed in Table I for the PSO and QPSO algorithms.

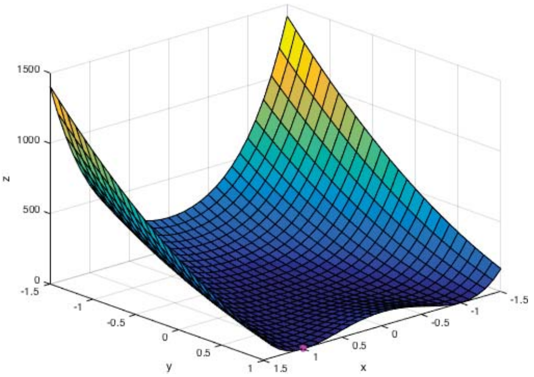

Fig. 2 Two-dimensional Rosenbrock function in 3D

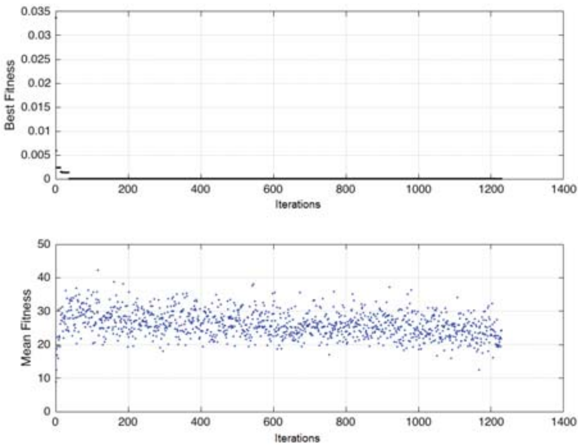

Fig. 3 PSO response to the two-dimensional Rosenbrock function

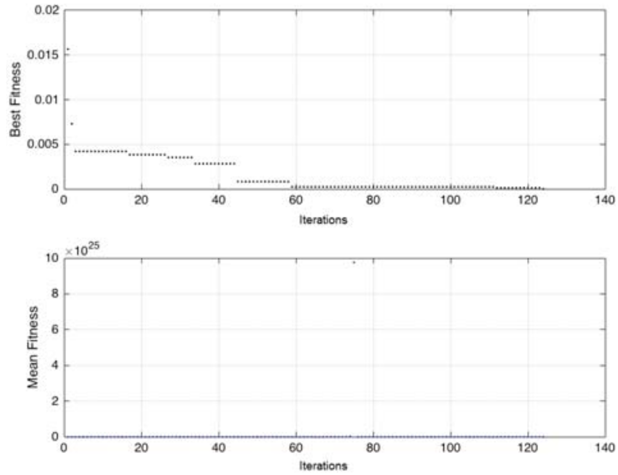

Fig. 4 QPSO response to the two-dimensional Rosenbrock function

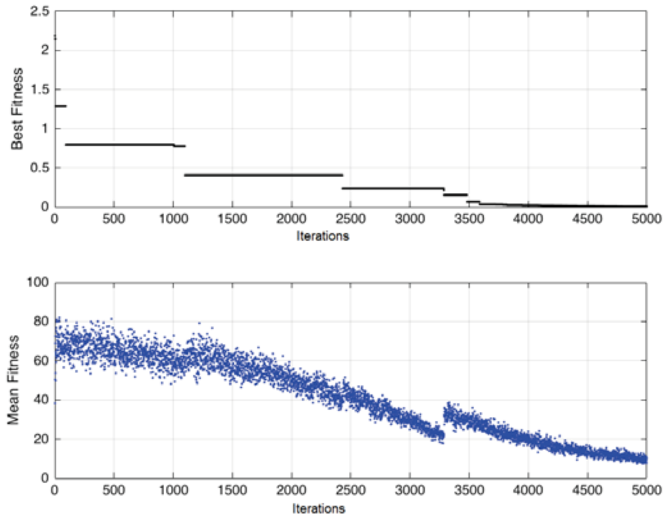

Fig. 5 PSO response to the five-dimensional Rosenbrock function

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 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).

$$\sum_{i=1}^{N-1} 100(x_{i+1} - {x_i}^2)^2 + (1 - x_i)^2 \qquad (6)$$

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

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

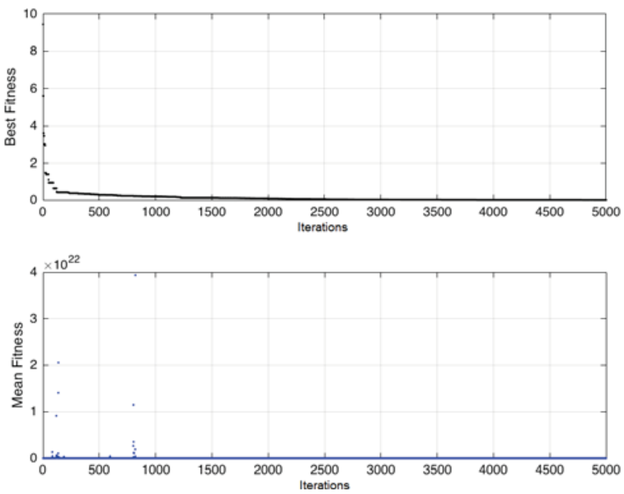

Fig. 6 QPSO response to the five-dimensional Rosenbrock function

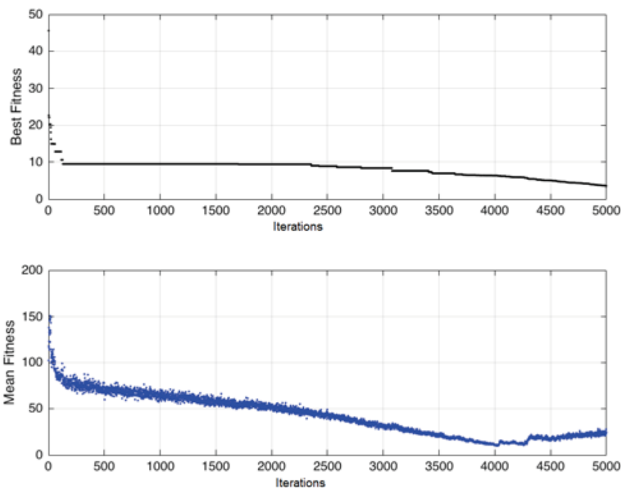

Fig. 7 PSO response to the ten-dimensional Rosenbrock function

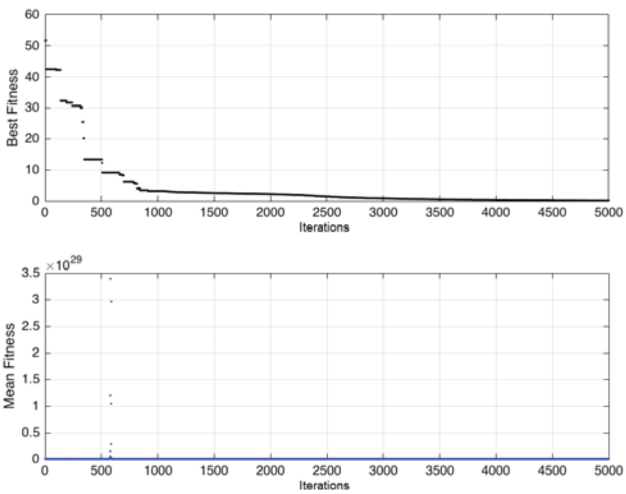

Fig. 8 QPSO response to the ten-dimensional Rosenbrock function

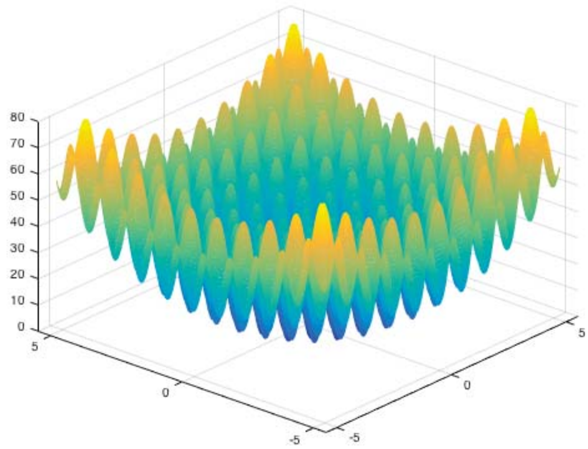

Fig. 9 Two-dimensional Rastrigin function in 3D

speed of convergence.

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 it's 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 it's 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).

$$\sum_{i=1}^{N-1} \left(10 + {x_i}^2 - 10 \cdot \cos\left(2 \cdot \pi \cdot x_i\right)\right) \qquad (7)$$

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

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 II
PARAMETERS OF THE PSO AND QPSO ALGORITHMS

Number of particles to be optimized	100
Maximum number of steps in the algorithm	5000
Social Parameter	2
Cognitive Parameter	2
Stop Criterion (Error)	10^{-10}

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

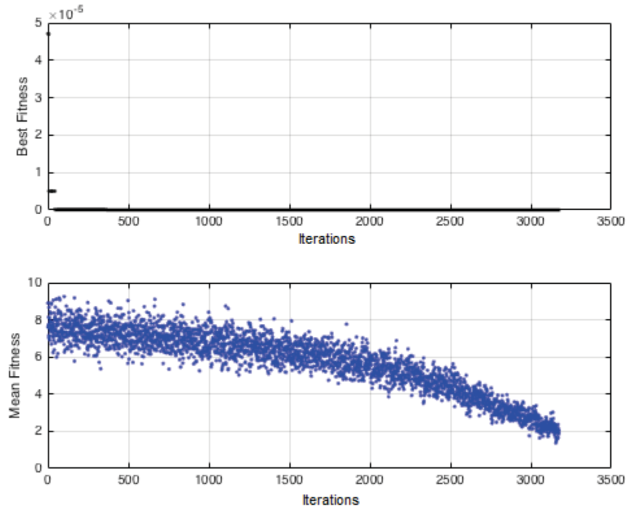

Fig. 10 PSO response to the two-dimensional Rastrigin function

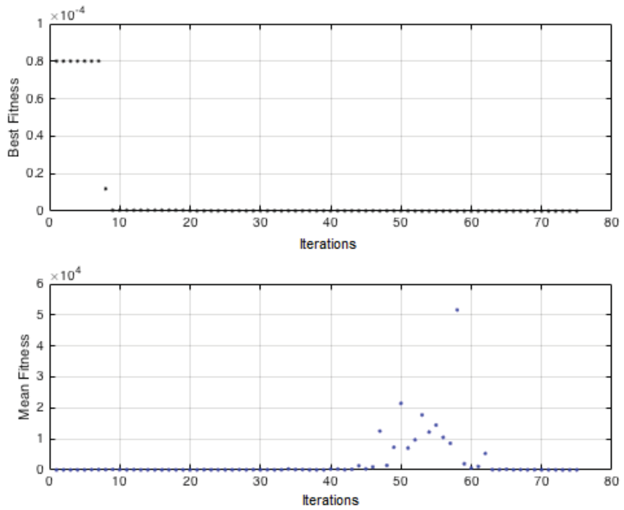

Fig. 11 QPSO response to the two-dimensional Rastrigin function

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.

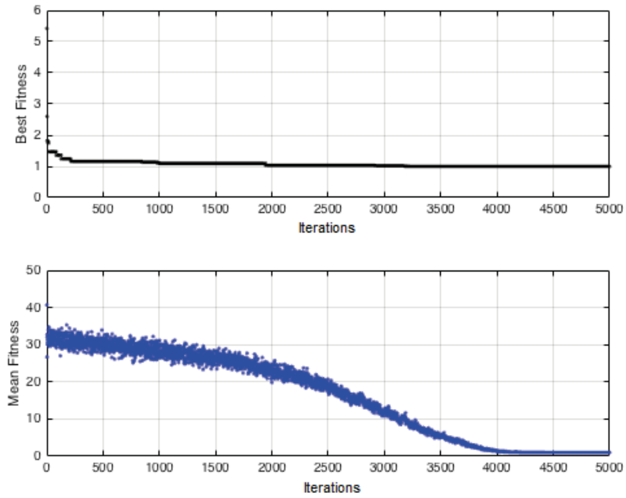

Fig. 12 PSO response to the five-dimensional Rastrigin function

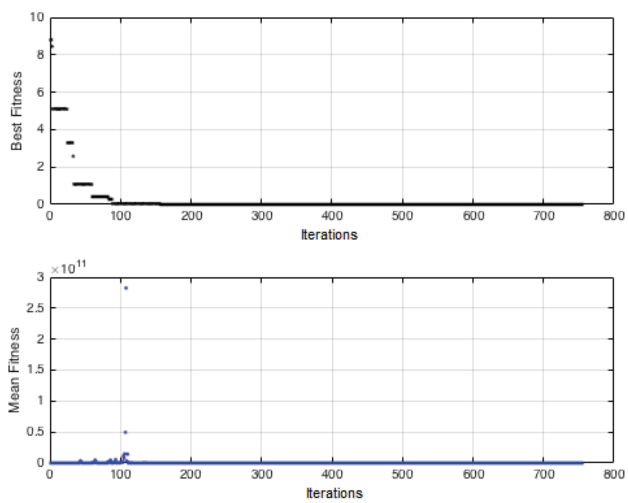

Fig. 13 QPSO response to the five-dimensional Rastrigin function

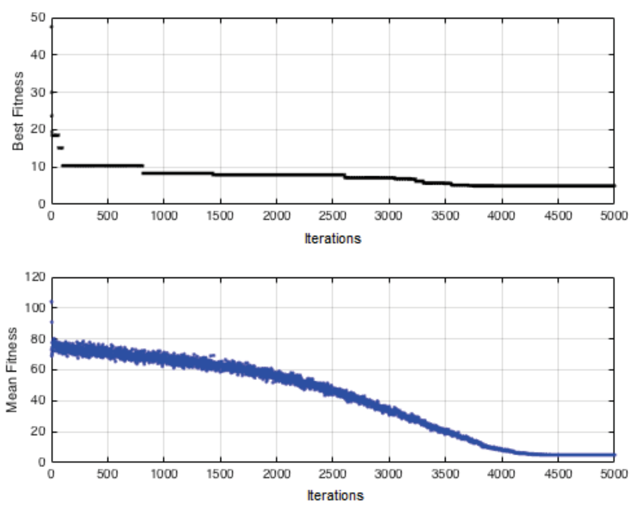

Fig. 14 PSO response to the ten-dimensional Rastrigin function

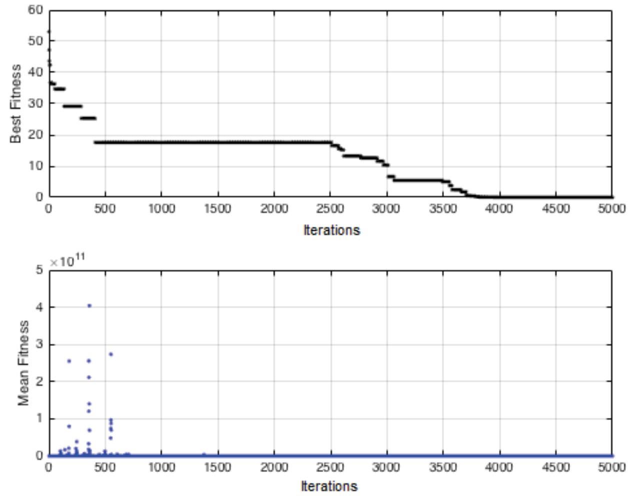

Fig. 15 QPSO response to the ten-dimensional Rastrigin function

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.

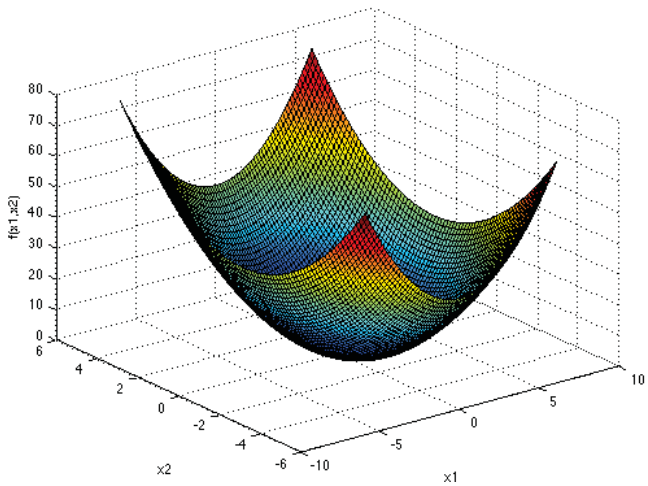

Fig. 16 Two-dimensional Sphere function in 3D

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

$$f(x, y) = \sum_{i=1}^{n} (x_i^2) \quad (8)$$

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

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.

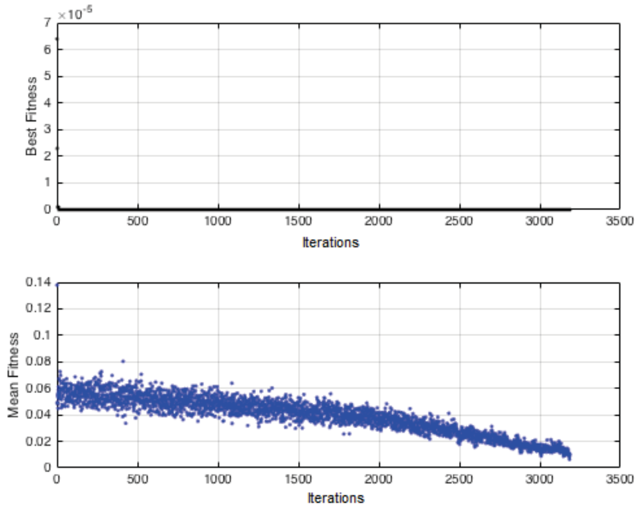

Fig. 17 PSO response to the two-dimensional Sphere function

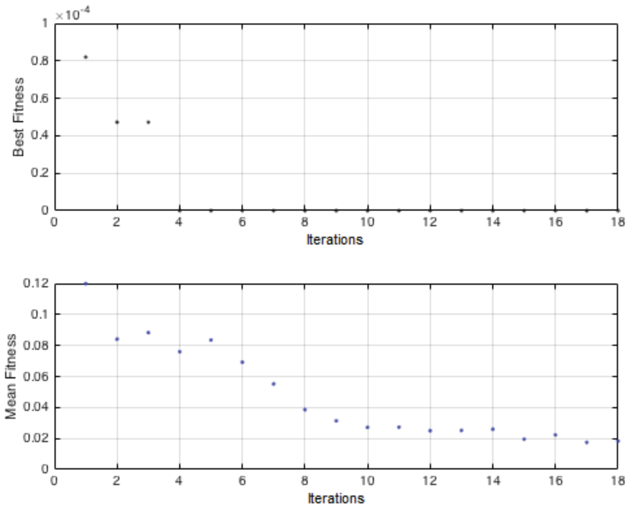

Fig. 18 QPSO response to the two-dimensional Sphere function

TABLE III
PARAMETERS OF THE PSO AND QPSO ALGORITHMS FOR THE SPHERE FUNCTION

Number of particles to be optimized	100
Maximum number of steps in the algorithm	5000
Social Parameter	2
Cognitive Parameter	2
Stop Criterion (Error)	10^{-12}

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

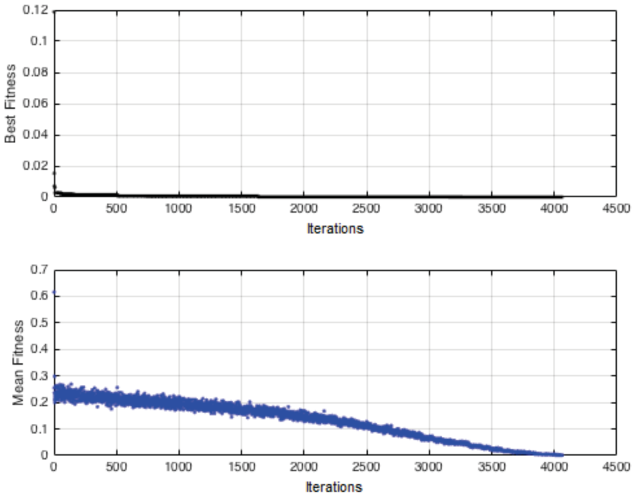

Fig. 19 PSO response to the five-dimensional Sphere function

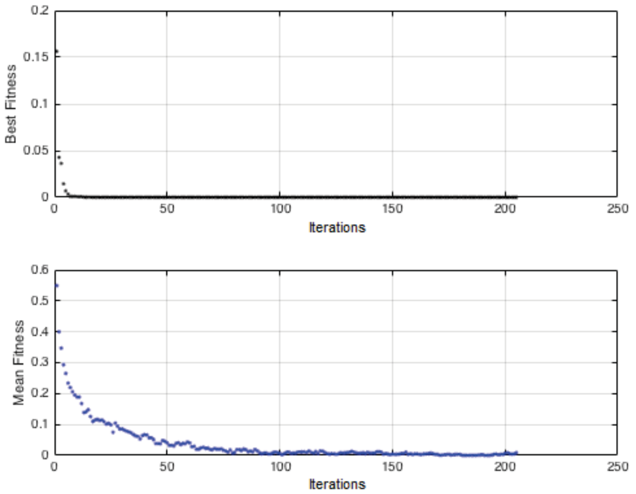

Fig. 20 QPSO response to the five-dimensional Sphere function

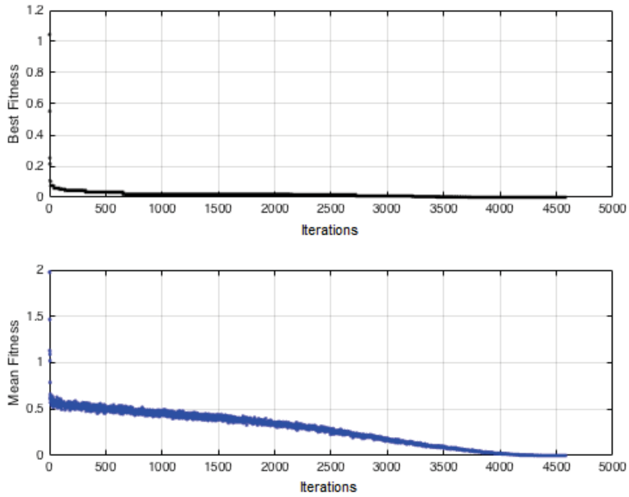

Fig. 21 PSO response to the ten-dimensional Sphere function

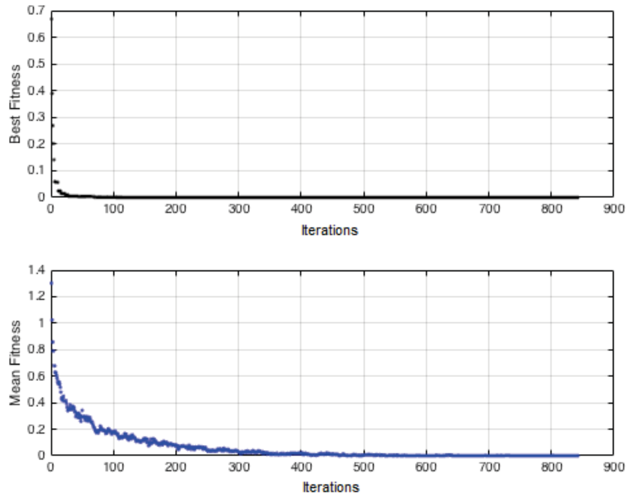

Fig. 22 QPSO response to the ten-dimensional Sphere function

in Figs. 19 and 20.

The PSO algorithm reached the stopping criterion with 4062 iterations and the QPSO just needed 205. PSO reached $8.07 \cdot 10^{-13}$ and QPSO $6.32 \cdot 10^{-13}$. 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 $2.83 \cdot 10^{-13}$ while the PSO needed 4575 iterations to reach $7.78 \cdot 10^{-13}$, 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

[1] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951. (Online). Available: http://dx.doi.org/10.1214/aoms/1177729586.

[2] N. Barricelli, "Esempi numerici di processi di evoluzione," *Methodos*, no. 21-22, pp. 45–68, 1954, cited By 55.

[3] I. Rechenberg, "Cybernetic solution path of an experimental problem," *Evolutionary Computation: The Fossil Record*, pp. 301–310, 1968, cited By 2.

[4] L. J. Fogel, *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming*. New York, NY, USA: John Wiley & Sons, Inc., 1999.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983. (Online). Available: http://science.sciencemag.org/content/220/4598/671.

[6] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers and Operations Research*, vol. 13, no. 5, pp. 533 – 549, 1986, applications of Integer Programming. (Online). Available: http://www.sciencedirect.com/science/article/pii/0305054886900481.

[7] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[9] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company, 2004.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," 1995.

[11] L. M., "Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality," 2002.

[12] P. S. Arrhenius, "Xxxi. on the influence of carbonic acid in the air upon the temperature of the ground," *Philosophical Magazine*, vol. 41, no. 251, pp. 237–276, 1896. (Online). Available: http://dx.doi.org/10.1080/14786449608620846.

[13] A. Taghvaei, P. G. Mehta, and S. P. Meyn, "Error estimates for the kernel gain function approximation in the feedback particle filter," in *2017 American Control Conference (ACC)*, May 2017, pp. 4576–4582.

[14] A. Taghvaei and P. G. Mehta, "Gain function approximation in the feedback particle filter," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 5446–5452.

[15] G. A. Hoffmann, "Function approximation with learning networks in the financial field and its application to the interest rate sector," in *Proceedings of 1995 Conference on Computational Intelligence for Financial Engineering (CIFEr)*, Apr 1995, pp. 178–182.

[16] I. Schalk-Schupp, F. Faubel, M. Buck, and A. Wendemuth, "Approximation of a nonlinear distortion function for combined linear and nonlinear residual echo suppression," in *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sept 2016, pp. 1–5.

[17] J. E. A., T. P. I. Ahamed, and R. T., "A function approximation approach to reinforcement learning for solving unit commitment problem with photo voltaic sources," in *2016 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, Dec 2016, pp. 1–6.

[18] A. B. de Souza, "Fundamentos de otimizacao por inteligencia de enxames: uma visao geral," *Sba: Controle e Automacao Sociedade Brasileira de Automatica*, vol. 20, pp. 271 – 304, 09 2009.

[19] Z. Meng, P. Feng, P. Chao, L. Weixing, and G. Qi, "Trajectory optimization using time-separating strategy with improved pso on mechanical arms," in *2017 36th Chinese Control Conference (CCC)*, July 2017, pp. 2669–2674.

[20] C. H. R. Jethmalani, S. P. Simon, K. Sundareswaran, P. S. R. Nayak, and N. P. Padhy, "Auxiliary hybrid pso-bpnn-based transmission system loss estimation in generation scheduling," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1692–1703, Aug 2017.

[21] L. Gong, W. Cao, and J. Zhao, "An improved pso algorithm for high accurate parameter identification of pv model," in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, June 2017, pp. 1–5.

[22] T. Guan and F. Zhuo, "An improved sa-pso global maximum power point tracking method of photovoltaic system under partial shading conditions," in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, June 2017, pp. 1–5.

[23] L. Benchikhi, M. Sadgal, and A. El-Fazziki, "An optimization approach of parameters in image processing based on pso: Case of quality control," in *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, Oct 2013, pp. 1–6.

[24] K. Wang, X. Yan, Y. Yuan, X. Jiang, G. Lodewijks, and R. R. Negenborn, "Pso-based method for safe sailing route and efficient speeds decision-support for sea-going ships encountering accidents," in *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, May 2017, pp. 413–418.

[25] M. K. Pamba R.V., Sherly E., "Evaluation of frequent pattern growth based fuzzy particle swarm optimization approach for web document clustering." in *Computational Science and Its Applications ICCSA 2017. ICCSA 2017. Lecture Notes in Computer Science, vol 10404.*, 2017.

[26] M. S. K. P. Jatana N., Suri B. and C. A. R., "Particle swarm based evolution and generation of test data using mutation testing." in *Computational Science and Its Applications ICCSA 2016. ICCSA 2016. Lecture Notes in Computer Science, vol 9790*, 2016.

[27] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 1, June 2004, pp. 325–331 Vol.1.

[28] C. Zhang, Y. Xie, D. Liu, and L. Wang, "Fast threshold image segmentation based on 2d fuzzy fisher and random local optimized qpso," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1355–1362, March 2017.

[29] R. Faia, T. Pinto, and Z. Vale, "Optimization of electricity markets participation with qpso," in *2016 13th International Conference on the European Energy Market (EEM)*, June 2016, pp. 1–5.

[30] X. Xie, H. Wang, S. Tian, and Y. Liu, "Optimal capacity configuration of hybrid energy storage for an isolated microgrid based on qpso algorithm," in *2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, Nov 2015, pp. 2094–2099.

[31] L. A. Rastrigin, "Systems of extremal control," 1974.

[32] H. Mühlenbein, M. Schomisch, and J. Born, "Paper: The parallel genetic algorithm as function optimizer," *Parallel Comput.*, vol. 17, no. 6-7, pp. 619–632, Sep. 1991. (Online). Available: http://dx.doi.org/10.1016/S0167-8191(05)80052-3.

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.