



Improving the PSO method for global optimization problems

Ioannis G. Tsoulos¹ · Alexandros Tzallas¹ · Evaggelos Karvounis¹

Received: 18 November 2019 / Accepted: 5 February 2020 / Published online: 15 February 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

The paper introduces two modifications for the well-known PSO method to solve global optimization problems. The first modification deals with the termination of the method and the second with the bounding of the so-called velocity in order to prevent the method from creating particles outside the domain range of the objective function. The modified method was tested on a series of global optimization problems from the relevant literature and the results are reported.

Keywords Particle swarm optimization · Stochastic methods · Termination rules

1 Introduction

The common problem of estimating the global minimum of a multi-dimensional function is given as

$$x^* = \arg \min_{x \in S} f(x) \quad (1)$$

where $S \subset R^n$ is defined by:

$$S = [a_1, b_1] \otimes [a_2, b_2] \otimes \dots \otimes [a_n, b_n] \quad (2)$$

This problem appears in many scientific areas such as Yapou et al. (1998), Duan et al. (1992), chemistry Wales and Scheraga (1999), Pardalos et al. (1994), economics Gaing (2003) etc. During the past years many methods have been proposed in the relevant literature to tackle the global optimization problem. These methods usually are divided into two main categories: deterministic and stochastic. The first category includes techniques such as interval methods Lin and Stadtherr (2004), the TRUST method Barhen et al. (1997), fuzzy optimization methods Angelov (1994) etc. Typically, the deterministic methods are more hard to implement than stochastic methods and in some cases they require a—pory knowledge of the objective function. On the other hand, the stochastic methods are more easy to implement and hence there are a great variety of stochastic optimization methods such as: Simulated Annealing Kirkpatrick et al. (1983),

Controlled Random Search Price (1977), Genetic Algorithms Goldberg (1989), Michalewicz (1996) and Particle Swarm Optimization Kennedy and Eberhart (1999). The Particle Swarm Optimization method (PSO) is a , population-based method. The method creates a population of candidate solutions (swarm of particles) that are evolved simulating the movement of actual particles. For that purpose, each particle maintains its current position \vec{x} and a corresponding velocity \vec{u} . The PSO method has been applied with success in a wide area of problems such as problems from physics de Moura Meneses et al. (2009), Shaw and Srivastava (2007), medicine Wachowiak et al. (2004), Marinakis (2008), economics Park et al. (2010), electronics Hosseini et al. (2019) etc. During the recent years, many modifications of the original PSO method have been suggested such as hybrid techniques Liu et al. (2005), Shi et al. (2005), methods to improve the calculation of the parameters of the method (such as inertia, velocity) Tang and Fang (2015), Yasuda and Iwasaki (2004), Shahzad et al. (2009), methods that adapt the control parameters of the PSO in order to learn only feasible solutions Isiet and Gadala (2019), chaotic quantum-behaved particle swarm optimization techniques Mariani et al. (2012), Araujo and Coelho (2008) etc.

Hybrid methods tends to locate more accurately the global minimum but usually the required an additional amount of function calls. On the other hand, methods that used modified versions of the velocity tends to avoid the explosion problem of the PSO technique but in some cases they are trapped to a local minimum instead of locating the global one.

✉ Ioannis G. Tsoulos
itsoulos@uoi.gr

¹ Department of Informatics and Telecommunications,
University of Ioannina, 47100 Arta, Greece

This article proposes a new stopping rule specifically designed for PSO methods and a technique to bound the velocity avoiding calculation of the objective function outside of the domain definition S (Eq. 2). Various experiments were conducted on a series of well-known test functions from the relevant literature in order to demonstrated the efficiency of the proposed modifications and the results are listed.

The rest of this article is organized as follows: in Sect. 2 the general description of the PSO method is provided as well as the proposed modifications, in Sect. 3 the test functions are described with the conducted experiments and finally in Sect. 4 some conclusions are derived.

2 Method description

The general scheme of the PSO method is listed in Algorithm 1. The algorithm has m particles and each particle has n elements where n is the dimension of the objective function. Each of the m particles is associated with two vectors: the position x_i of the particle and the velocity u_i of the particle. At every iteration a new position x_i is calculated as a combination of the old position x_i , the associated velocity u_i , the previous best location of the particle p_i and the best location of all particle p_{best} . The new point x_i is updated using

$$x_i = x_i + u_i \quad (3)$$

In most cases the following scheme is used to update the j element of the velocity u_i

$$u_{ij} = \omega u_{ij} + r_1 c_1 (p_{ij} - x_{ij}) + r_2 c_2 (p_{\text{best}_j} - x_{ij}) \quad (4)$$

where

1. The variables r_1, r_2 are random numbers in the range $[0,1]$.
2. The parameters c_1, c_2 are constant numbers usually in the range $[1,2]$.
3. The variable ω is called inertia and typically is in the range $[0,1]$. This article uses an update mechanism for the inertia proposed in Shi and Eberhart (1998) and it is given below:

$$\omega = \omega_{\text{max}} - \frac{k}{k_{\text{max}}} (\omega_{\text{max}} - \omega_{\text{min}}) \quad (5)$$

where k is the current number of iterations and k_{max} the maximum number of allowed iterations. Also, $\omega_{\text{min}}, \omega_{\text{max}}$ are user defined and for this article $\omega_{\text{min}} = 0.4, \omega_{\text{max}} = 0.9$.

Nevertheless, it is clear that Eq. 3 can produce elements that are not belong to set S . This article proposes in Sect. 2.3 a technique that can prevent such cases.

The PSO algorithm terminates when the termination criteria are hold. In these article two stopping criteria are used: a termination check based on the variance of the best discovered minimum that was also proposed in the previous work Tsoulos (2008) and a new termination check based on the changes of the best located positions. The first procedure is defined in Sect. 2.1 and the second in Sect. 2.2.

Algorithm 1 The generic PSO algorithm**1. Initialization.**

- (a) **Initialize** $k = 0$ as the number of current iteration.
- (b) **Set** m as the total number of particles.
- (c) **Initialize** the positions x_1, x_2, \dots, x_m , with $x_i \in S \subset R^n$
- (d) **Initialize** the velocities of the particles u_1, u_2, \dots, u_m . Each element j of the u_i is randomly initialized in the range $[-d_j, d_j]$ with d_j :

$$d_j = \frac{1}{2} (b_j - a_j) \quad (6)$$

- (e) **For** $i = 1..m$ do $p_i = x_i$.
- (f) **Set** $p_{\text{best}} = \arg \min_{i \in 1..m} f(x_i)$

2. Termination Check. Check the termination criteria in order to stop the algorithm.**3. For** $i = 1..m$ **Do**

- (a) **Update** the velocity u_i as a function of $x_i, p_i, p_{\text{best}}$
- (b) **Set** $x_i = x_i + u_i$
- (c) **Evaluate** the fitness of the particle $i, f(x_i)$, where $f()$ is the objective function.
- (d) If $f(x_i) \leq f(p_i)$ then $p_i = x_i$

4. End For**5. Set** $p_{\text{best}} = \arg \min_{i \in 1..m} f(x_i)$ **6. Set** $k = k + 1$.**7. Goto** Step 2**2.1 Termination rule based on variance**

At every iteration k the variance of $f(p_{\text{best}})$ is measured. Denote this variance with $\sigma^{(k)}$. If there is no any new minimum found for a number of generations, then it is highly possible that the algorithm has found the global minimum and hence it should terminate. The algorithm terminates when

$$\sigma^{(k)} \leq \frac{\sigma^{(\text{klast})}}{2} \quad (7)$$

where klast is the last iteration where a new minimum was found. As an example consider the function Rastrigin defined as

$$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$$

The global minimum in the range $[-1, 1]^2$ is the value -2.0 . Now consider a PSO with 100 particles and let the algorithm terminates when the current number of iterations $k \geq 200$. The progress of minimization for the Rastrigin function is outlined in Fig. 1. The algorithm located the global minimum at the early stages of the run and before iteration 20. Now consider the plot of the variance of best located value at Fig. 2. The variance decreases very smoothly and hence it can be used as a criterion to terminate the algorithm and preventing unnecessary function calls.

2.2 Termination rule based on similarity

Let us denote with $p^{(k)}$ the table with the best locations of each particle at iteration k . Also set $E = 0$ at the beginning of the algorithm. At every iteration $k > 1$ we measure the quantity

$$g = \left\| p^{(k)} - p^{(k-1)} \right\| \quad (8)$$

If $g < e$, where e a small positive number, then $E = E + 1$. In other words the similarity between the best location between two successive iterations is measured. The algorithm terminates when $E \geq E_{\max}$, where E_{\max} a small positive integer number. To outline the efficiency of this stopping rule consider again the Rastrigin problem and the same test run with 100 particles and $K_{\max} = 200$ as in Sect. 2.1 In Fig. 3 the graph of the quantity g is plotted against the number of iterations. As it can be noticed the value of g very soon tends to zero and hence this quantity can be used as a stopping rule.

2.3 Velocity bound technique

A common problem of the PSO method is the production of points that are outside of the domain range of the objective function. This problem produced when the changes in the velocity performed without control and it is usually called explosion. The velocity bound technique suggested here is used to prevent this effect with some changes in the basic PSO algorithm.

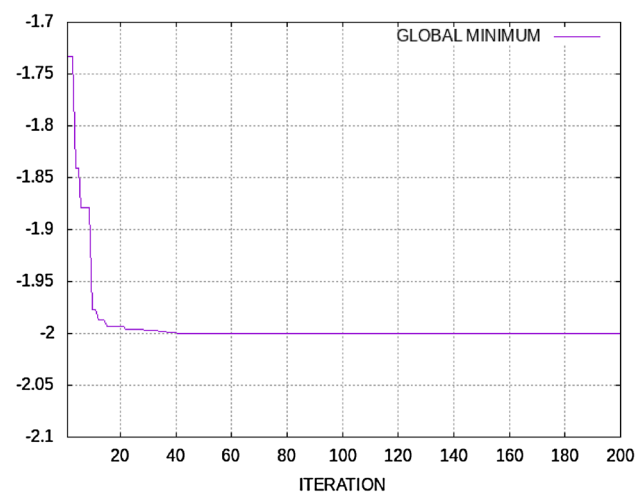


Fig. 1 Plot of the progress of minimization for the Rastrigin function

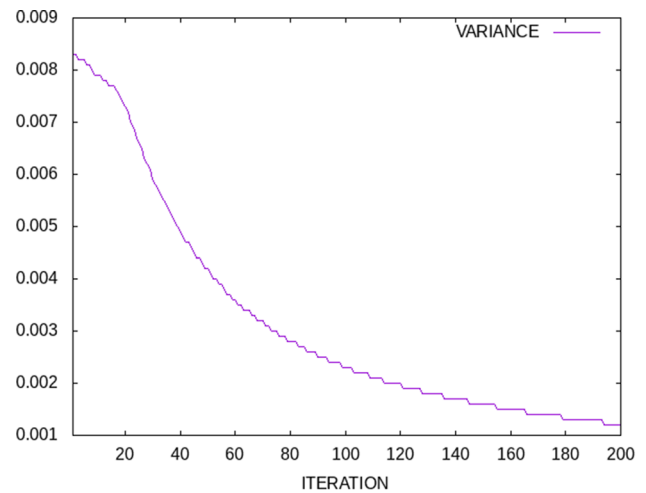


Fig. 2 The plot of the variance of the Rastrigin function

The first change is performed when the velocities are initialized. The new procedure for velocity initialization is shown in Algorithm 2.3. This procedure ensures that at least for the first iteration of the PSO method every point $x_i \in S$. The second change is to alter the Eq. 4 in order to ensure that every $x_i \in S$. The proposed change for the update of the velocities is listed in algorithm 2.3. This algorithm does not perform any changes in the velocity that will responsible for producing points x_i outside of the domain range.

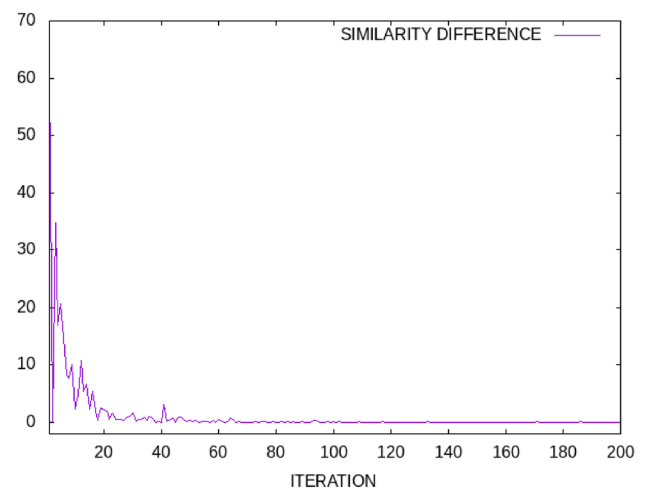


Fig. 3 Similarity check plot for the Rastrigin function

Algorithm 2 Initialization of velocities with bounding technique.

1. **For** $i=1..m$ **Do**
 - (a) **For** $j=1..n$ **Do**
 - i. Produce t_j randomly in $\left[-\frac{b_j-a_i}{2}, \frac{b_j-a_i}{2}\right]$
 - ii. **If** $x_{ij} + t_j \in [a_j, b_j]$ **then** $u_{ij} = t_j$ **else goto** step 1(a)i.
 - (b) **EndFor**
 2. **EndFor**
-

Algorithm 3 The new algorithm for altering the velocity vector.

1. **For** $i=1..m$ **Do**
 - (a) **For** $j=1..n$ **Do**
 - i. **Set** $t_j = \omega u_{ij} + r_1 c_1 (p_{ij} - x_{ij}) + r_2 c_2 (p_{\text{best},j} - x_{ij})$
 - ii. **If** $t_j \in [a_j, b_j]$ **then** $u_{ij} = t_j$
 - (b) **EndFor**
 2. **EndFor**
-

3 Experiments

In order to measure the efficiency of the proposed modifications a series of test functions provided in Ali et al. (2005) were used. These problems are listed in Table 1.

The proposed modifications have been evaluated in terms of number of function calls and successful discover of global minimum using four different experiments:

1. With the variance stopping rule only. In this test only the stopping rule that is based on variance was used. This test is denoted as **VARIANCE** in the experiments.
2. With the similarity stopping rule only, where only the stopping rule based on similarity was used. This test is denoted as **SIMILARITY** in the experiments.
3. A test that combines the variance stopping rule and the new bounding technique of velocities. This test is named as **VBOUND** in the experiments.
4. A test that combines both the similarity stopping rule and the new bounding technique. This test is named **SBOUND** in the experiments.

The parameters of the PSO method used in the relevant experiments are listed in Table 2. The results for the previous four experiments are listed in Table 3. The column

FUNCTION represents the objective function. The numbers in cells stand for the average number of function calls from 30 independent runs using different seeds for the random generator each time. The numbers in parentheses denote the fraction of runs where the global minimum was located. If this number is missing then the global minimum was discovered in every independent run (100% success). The last row of the table contains the total number of function calls for all experiments as well as the average number of success rate. At the end of every run the local search optimization method BFGS of Powell Powell (1989) was applied.

Also, the original pso method using the similarity stopping rule and the new bounding technique has been compared with Quantum behaved PSO method Sun et al. (2006) for the above test functions. The results from this comparison are listed in Table 4. The proposed stopping rule of similarity has been applied successfully in Quantum Pso method and as a consequence the method terminated successfully in the majority of cases. The Quantum PSO method requires less number of function calls for a series of optimization problems in two dimensions but for problems of higher dimension the proposed technique performs better.

From the conducted experiments it is clear that the SIMILARITY rule outperforms VARIANCE stopping rule in almost any test function in terms of function calls

Table 1 Test functions used in the experiments

Name	n	Expression
AP	2	$f(x) = \frac{1}{4}x_1^4 - \frac{1}{2}x_1^2 + \frac{1}{10}x_1 + \frac{1}{2}x_2^2$
BF1	2	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1) - \frac{4}{10} \cos(4\pi x_2) + \frac{7}{10}$
BF2	2	$f(x) = x_1^2 + 2x_2^2 - \frac{3}{10} \cos(3\pi x_1) \cos(4\pi x_2) + \frac{3}{10}$
BRANIN	2	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos(x_1) + 10$
CAMEL	2	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_1^4$
CB3	2	$f(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$
EXP8	8	$f(x) = -\exp\left(-0.5 \sum_{i=1}^8 x_i^2\right)$
EXP32	32	$f(x) = -\exp\left(-0.5 \sum_{i=1}^{32} x_i^2\right)$
RASTRIGIN	2	$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$
CM	4	$f(x) = \sum_{i=1}^n x_i^2 - \frac{1}{10} \sum_{i=1}^n \cos(5\pi x_i)$
Griewank2	2	$f(x) = 1 + \frac{1}{200} \sum_{i=1}^2 x_i^2 - \prod_{i=1}^2 \frac{\cos(x_i)}{\sqrt{ i }}$
Hansen	2	$f(x) = \sum_{i=1}^5 i \cos[(i-1)x_1 + i] \sum_{j=1}^5 j \cos[(j+1)x_2 + j]$
Rosenbrock8	8	$f(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$
Rosenbrock32	32	$f(x) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$
Gkls250	2	$f(x) = \text{Gkls}(x,n,w)$ with $n = 2, w = 50$ local minima Gaviano et al. (2003)
Gkls350	3	$f(x) = \text{Gkls}(x,n,w)$ with $n = 2, w = 50$ local minima Gaviano et al. (2003)
Shekel5	4	$f(x) = -\sum_{i=1}^5 \frac{1}{(x-a_i)(x-a_i)^f + c_i}$
Shekel7	4	$f(x) = -\sum_{i=1}^7 \frac{1}{(x-a_i)(x-a_i)^f + c_i}$
Shekel10	4	$f(x) = -\sum_{i=1}^{10} \frac{1}{(x-a_i)(x-a_i)^f + c_i}$
Shubert	2	$f(x) = -\sum_{i=1}^2 \sum_{j=1}^5 j \{ \sin((j+1)x_i) + 1 \}$
Test2n4	4	$f(x) = \frac{1}{2} \sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i$
Test2n5	5	$f(x) = \frac{1}{2} \sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i$
Test2n6	6	$f(x) = \frac{1}{2} \sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i$
Sinu8	8	$f(x) = -(2.5 \prod_{i=1}^n \sin(x_i - z) + \prod_{i=1}^n \sin(5(x_i - z)))$
Sinu32	32	$f(x) = -(2.5 \prod_{i=1}^n \sin(x_i - z) + \prod_{i=1}^n \sin(5(x_i - z)))$
Potential3	9	The Lennard Jones potential Lennard-Jones (1924) for 3 atoms.
Potential5	15	The Lennard Jones potential Lennard-Jones (1924) for 5 atoms

Table 2 Parameters of the PSO method

Parameter	Value
m	200
k_{\max}	200
E_{\max}	3
c_1	1
c_2	1

and success rate. Also, the new bounding technique for the velocity reduces dramatically (in most cases) the number of required function calls for both stopping rules.

Table 3 Experiments with the two stopping rules and the bound technique

Function	VARIANCE	SIMILARITY	VBOUND	SBOUND
AP	29500.3	22026	13075.8	12426.6
BF1	28123.2(0.80)	23766.5	16634.5(0.77)	13465.5
BF2	24505.8(0.70)	23921.1	16617.5(0.77)	13502.7
BRANIN	26523.9	22682	8111.83(0.97)	7161.37
CAMEL	29538.1	22815.8	14639.6	11911.6
CB3	29439.5	21100.7	15804.6	11876.7
CM	36351.4(0.97)	19012.7	12042.7	6110.7
SHUBERT	26979.9(0.97)	27690.1	5861.37(0.90)	6049.8
EXP8	37692.1	19466.9	3460.23	2087.6
EXP32	30722	20464.33	19465.66	10644.33
GKLS250	28421.1	18933.5	10856.9	9585.53
GKLS350	36241.6	18969.7	11202.8	6090.43
GRIEWANK2	32053.7(0.80)	27391.1	14936.2(0.73)	14986.5
HANSEN	29347.4(0.97)	28303	6580.27	6013.17
RASTRIGIN	35609.1(0.97)	21319.5	16720.6(0.90)	10690.8
ROSENBROCK8	35542.9	25470.6	1747.07	1157.4
ROSENBROCK32	21912.4	10632.73	5829.53	3882.9
SINU8	36720.2	18602.2	1977.17(0.90)	1159.97(0.93)
SINU32	7962.27(0.30)	5526.53(0.10)	2701.54(0.67)	1701.67(0.67)
SHEKEL5	34806.7(0.50)	21014.1(0.50)	4588.8(0.43)	3423.13(0.43)
SHEKEL7	32809.9(0.63)	22194.8(0.63)	5544.13(0.60)	3934.77(0.60)
SHEKEL10	30857.6(0.67)	22985.8(0.77)	7392.1(0.67)	4475(0.67)
TEST2N4	35947.6(0.97)	20068	2805.2(0.90)	1823.97
TEST2N5	27065(0.80)	19911.9	2009.1(0.73)	1232.2(0.83)
TEST2N6	26430.2(0.73)	19555	5432.17(0.77)	3044.19(0.90)
POTENTIAL3	11190.9	25463.2	1848.7	1882.33
POTENTIAL5	5314.5	17200.6	1714.43	1721.93
Total	767608.27(0.88)	566488.39(0.93)	229510.5(0.88)	170042.76(0.93)

4 Conclusions

Two modifications for the general PSO method have been proposed in this article: a stopping rule and a bounding technique for the velocity of the particles. Both modifications are general enough and they can be incorporated in any PSO variant. These modifications does not require any additional information from the objective function and they

do not affect the speed of the algorithm since they have small overhead in the computation time. The experimental results indicates that the new stopping rule outperforms another statistical based stopping rule used in the past years and also the conjunction of the new stopping rule with the bounding technique reduces the number of function evaluations needed to discover the global minimum.

Table 4 The proposed method compared against Quantum PSO using the similarity stopping rule

Function	QUANTUM	SBOUND
AP	18160.1	12426.6
BF1	2779.33 (0.93)	13465.5
BF2	2727.57 (0.93)	13502.7
BRANIN	10496.7	7161.37
CAMEL	15895.9	11911.6
CB3	2479.43	11876.7
CM	2158.4 (0.93)	6110.7
SHUBERT	13082.4 (0.93)	6049.8
EXP8	2161.17	2087.6
EXP32	2840.63	10644.33
GKLS250	2392.03	9585.53
GKLS350	15168.5 (0.90)	6090.43
GRIEWANK2	4198.37 (0.93)	14986.5
HANSEN	19378.2 (0.80)	6013.17
RASTRIGIN	4314.63 (0.97)	10690.8
ROSENBROCK8	3461.47	1157.4
ROSENBROCK32	3631.73	3882.9
SINU8	19332.8 (0.67)	1159.97 (0.93)
SINU32	6591.97 (0.03)	1701.67 (0.67)
SHEKEL5	19066.2 (0.10)	3423.13 (0.43)
SHEKEL7	18824 (0.13)	3934.77 (0.60)
SHEKEL10	4336.5 (0.07)	4475 (0.67)
TEST2N4	16318.1 (0.57)	1823.97
TEST2N5	13001.8 (0.43)	1232.2 (0.83)
TEST2N6	14224.4 (0.20)	3044.19 (0.90)
POTENTIAL3	5128	1882.33
POTENTIAL5	10745.8	1721.93
Total	252896.03 (0.77)	170042.76 (0.93)

Funding This work is partly funded by the project entitled HuMORIST-Hospital MOnitorIng SysTem, co-financed by the European Union and Greek national funds through the Operational Program for Research and Innovation Smart Specialization Strategy (RIS3) of Ipeiros (Project Code: HPI1AB-00260).

References

Ali MM, Khompatraporn C, Zabinsky ZB (2005) A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *J Glob Optimiz* 31:635–672

Angelov P (1994) A generalized approach to fuzzy optimization. *Int J Intell Syst* 1994:261–268

Araujo E, Coelho LS (2008) Particle swarm approaches using Lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vacuum system. *Appl Soft Comput* 8:1354–1364

Barhen J, Protopopescu V, Reister D (1997) TRUST: a deterministic algorithm for global optimization. *Science* 16:1094–1097

de Moura Meneses AA, Machado MD, Schirru R (2009) Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor. *Progr Nuclear Energy* 51:319–326

Duan Q, Sorooshian S, Gupta V (1992) Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour Res* 28:1015–1031

Gaing Z-L (2003) Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Trans Power Syst* 18:1187–1195

Garg H (2016) A hybrid PSO-GA algorithm for constrained optimization problems. *Appl Math Comput* 274:292–305

Gaviano M, Ksasov DE, Lera D, Sergeyev YD (2003) Software for generation of classes of test functions with known local and global minima for global optimization. *ACM Trans Math Softw* 29:469–480

Goldberg D (1989) Genetic algorithms in search. Optimization and machine learning. Addison-Wesley Publishing Company, Reading

Hosseini SA, Hajipour A, Tavakoli H (2019) Design and optimization of a CMOS power amplifier using innovative fractional-order particle swarm optimization. *Appl Soft Comput* 85:105831

Isiet M, Gadala M (2019) Self-adapting control parameters in particle swarm optimization. *Appl Soft Comput* 83:105653

Kennedy J, Eberhart RC (1999) The particle swarm: social adaptation in information processing systems. In: Corne D, Dorigo M, Glover F (eds) *New ideas in optimization*. McGraw-Hill, Cambridge, pp 11–32

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220:671–680

Lennard-Jones JE (1924) On the determination of molecular fields. *Proc R Soc Lond A* 106:463–477

Lin Y, Stadtherr MA (2004) Advances in interval methods for deterministic global optimization in chemical engineering. *J Glob Optimiz* 29:281–296

Liu B, Wang L, Jin YH, Tang F, Huang DX (2005) Improved particle swarm optimization combined with chaos. *Chaos Solitons Fractals* 25:1261–1271

Mariani VC, Duck ARK, Guerra FA (2012) Leandro dos Santos Coelho, R.V. Rao, Heat exchanger design, Shell and tube heat exchanger (STHE), Economic optimization, Particle swarm optimization, Quantum particle swarm optimization, Chaos theory. *Appl Therm Eng* 42:119–128

Marinakis Y (2008) Magdalene Marinaki, Georgios Dounias, particle swarm optimization for pap-smear diagnosis. *Expert Syst Appl* 35:1645–1656

Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer, Berlin

Pardalos PM, Shalloway D, Xue G (1994) Optimization methods for computing global minima of nonconvex potential energy functions. *J Glob Optimiz* 4:117–133

Park J-B, Jeong Y-W, Shin J-R, Lee KY (2010) An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Trans Power Syst* 25:156–166

Powell MJD (1989) A tolerant algorithm for linearly constrained optimization calculations. *Math Program* 45:547–566

Price WL (1977) Global optimization by controlled random search. *Comput J* 20:367–370

Shahzad F, Baig AR, Masood S, Kamran M, Naveed N (2009) Opposition-based particle swarm optimization with velocity clamping (OVCP SO). In: Yu W, Sanchez EN (eds) *Advances in computational intelligence. Advances in intelligent and soft computing*, vol 116. Springer, Berlin, Heidelberg

Shaw R, Srivastava S (2007) Particle swarm optimization: a new tool to invert geophysical data. *Geophysics* 2007:72

Shi Y, Eberhart RC (1998) Parameter Selection in particle swarm optimization. In: *Evolutionary Programming VII. Lecture Notes in Computer Science*, vol 1447. Springer, Berlin, pp 591–600

Shi XH, Liang YC, Lee HP, Lu C, Wang LM (2005) An improved GA and a novel PSO-GA based hybrid algorithm. *Inf Process Lett* 93:255–261

- Sun J, Xu W, Fang W, Algorithm Quantum-Behaved Particle Swarm Optimization, with Controlled Diversity. In: Alexandrov VN, van Albada GD, Sloot PMA, Dongarra J (eds) Computational science–ICCS 2006. ICCS, (2006) Lecture Notes in Computer Science, vol 3993. Springer, Berlin, Heidelberg, p 2006
- Tang R-L, Fang Y-J (2015) Modification of particle swarm optimization with human simulated property. *Neurocomputing* 153:319–331
- Tsoulos IG (2008) Modifications of real code genetic algorithm for global optimization. *Appl Math Comput* 203:598–607
- Wachowiak MP, Smolikova R, Yufeng Z, Zurada JM, Elmaghraby AS (2004) An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE Trans Evol Comput* 8:289–301
- Wales DJ, Scheraga HA (1999) Global optimization of clusters, crystals, and biomolecules. *Science* 27:1368–1372
- Yapo PO, Gupta HV, Sorooshian S (1998) Multi-objective global optimization for hydrologic models. *J Hydrol* 204:83–97
- Yasuda K, Iwasaki N (2004) Adaptive particle swarm optimization using velocity information of swarm. In: 2004 IEEE international conference on systems, man and cybernetics (IEEE Cat. No.04CH37583), The Hague, pp 3475–3481, vol 4

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