Research Article

An Efficient Algorithm for Unconstrained Optimization

**Sergio Gerardo de-los-Cobos-Silva,¹ Miguel Ángel Gutiérrez-Andrade,¹
Roman Anselmo Mora-Gutiérrez,² Pedro Lara-Velázquez,¹
Eric Alfredo Rincón-García,² and Antonin Ponsich²**

¹*Departamento de Ingeniería Eléctrica, Universidad Autónoma Metropolitana-Iztapalapa, Avenida San Rafael Atlixco 186, Colonia Vicentina, Delegación Iztapalapa, 09340 México, DF, Mexico*

²*Departamento de Sistemas, Universidad Autónoma Metropolitana-Azcapotzalco, Avenida San Pablo 180, Colonia Reynosa Tamaulipas, 02200 México, DF, Mexico*

Correspondence should be addressed to Eric Alfredo Rincón-García; rigaeral@correo.azc.uam.mx

Received 29 April 2015; Revised 4 August 2015; Accepted 6 August 2015

Academic Editor: Peide Liu

Copyright © 2015 Sergio Gerardo de-los-Cobos-Silva et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an original and efficient PSO algorithm, which is divided into three phases: (1) stabilization, (2) breadth-first search, and (3) depth-first search. The proposed algorithm, called PSO-3P, was tested with 47 benchmark continuous unconstrained optimization problems, on a total of 82 instances. The numerical results show that the proposed algorithm is able to reach the global optimum. This work mainly focuses on unconstrained optimization problems from 2 to 1,000 variables.

1. Introduction

In general, an optimization problem can be defined as

$$\min_{x \in X} h(x). \quad (1)$$

In recent decades several heuristic optimization methods have been developed. These techniques are able to find solutions close to the optimum, where exact or analytic methods cannot produce optimal solutions within reasonable computation time. This is especially true when a global optimum is surrounded by many local optima, a situation known as deep valleys or black holes.

Different heuristic solution methods have been developed, among others, Tabu Search (TS) [1], Simulated Annealing (SA) [2], genetic algorithms (GA) [3], Scatter Search (SS) [4], and particle swarm optimization (PSO) [5].

In particular, for PSO there are so many different versions, a treaty on its taxonomy can be found in [6]. For a review of its variants, see [7]. With respect to unconstrained optimization, [8] proposed a modified algorithm to ensure the rational

flight of every particle's dimensional component; two parameters of fitness function evaluation, particle-distribution-degree, and particle-dimension-distance were introduced in order to avoid premature convergence. Reference [9] proposed a two-layer PSO (TLPSO) to increase the diversity of the particles so that the drawback of getting trapped in a local optimum was avoided. Reference [10] introduced a hybrid approach combining particle swarm optimization with genetic algorithm. Reference [11] presented the particle swarm optimization with flexible swarm; the algorithm was tested over 14 benchmark functions for 300,000 iterations for each function. Reference [12] presented two hybrids, a real-coded genetic algorithm-based PSO (RGA-PSO) method, and an artificial immune algorithm-based PSO (AIA-PSO) method. These algorithms were tested over 14 benchmark functions, with 20,000 objective function evaluations of the internal PSO approach for a problem with $N \leq 3$ decision variables and 60,000 objective function evaluations of the internal PSO approach for a problem with $5 \leq N \leq 30$. Reference [13] proposed IRPEO, an improved real-coded population-based EO (extremal optimization) method, that was compared with PSO and PSO-EO; the algorithms were

FIGURE 1: Griewank function.

FIGURE 2: Different functions and their *hardness [18].

TABLE 1: Time and efficiency.

	Function(dimension)	Part.	Mean Eff.	Time (sec)	Mean Iter.
1	Ackley(10)	60	0.994679	6.99	424.63
2	Ackley(120)	3	1.000000	1.46	322.08
3	Ackley(2)	24	0.998171	0.70	58.58
4	Ackley(30)	12	1.000000	1.39	185.25
5	Ackley(6)	36	0.998272	2.37	227.00
6	Beale(2)	60	1.000000	0.58	22.88
7	Bohachevsky 1(2)	240	1.000000	4.86	21.88
8	Bohachevsky 2(2)	240	1.000000	4.71	20.42
9	Booth(2)	240	1.000000	5.01	18.71
10	Branin(2)	24	0.999960	0.50	58.46
11	Bukin6(2)	12	0.875325	2.96	499.00
12	Colville(4)	3	0.791743	2.56	749.00
13	CrossLegTable(2) ** (0.83)	60	0.833426	8.07	198.42
14	Cross-in-Tray(2)	3	0.999908	1.61	422.38
15	Damavandi(2) ** (0.25)	60	0.989719	12.42	490.25
16	DevilleierGlasser02(5) ** (0.00)	12	1.000000	2.60	499.00
17	Drop Wave(2)	3	1.000000	0.78	223.54
18	Easom(2)	24	1.000000	0.27	17.54
19	Eggholder(2)	240	0.980028	32.87	499.00
20	Goldstein-Price(2)	60	1.000000	0.52	19.00
21	Griewank(10)	12	1.000000	1.23	185.50
22	Griewank(100)	24	1.000000	0.73	87.00
23	Griewank(2)	24	1.000000	1.53	191.04
24	Griewank(30)	12	1.000000	1.38	218.21
25	Griewank(6)	36	1.000000	1.37	86.13
26	Hartmann(3)	24	0.999999	0.57	43.50
27	Hartmann(6)	12	0.986473	2.21	399.50
28	Holder Table(2)	24	0.999863	1.79	201.79
29	Levy(12)	3	0.985893	1.75	499.00
30	Levy(120)	24	0.999918	6.07	499.00
31	Levy(2)	240	1.000000	5.53	17.92
32	Levy13(2)	240	1.000000	4.60	20.25
33	Matyas(2)	240	1.000000	4.92	17.92
34	McCormick(2)	12	0.999947	3.32	499.00
35	Michalewicz(2)	24	0.999998	0.38	39.33
36	Michalewicz(5)	120	0.957199	13.90	407.80
37	Powell(10)	12	1.000000	2.24	385.54
38	Powell(12)	3	1.000000	1.52	466.29
39	Powell(120)	3	1.000000	1.62	489.13
40	Powell(2)	3	1.000000	1.28	499.00
41	Power Sum(4)	240	0.994381	100.60	477.00
42	Rastrigin(10)	24	1.000000	1.91	252.75

**The overall success of different global optimizers according to Gavana [18].

tested over 12 benchmark functions with 10,000–50,000 iterations for problems with dimension $N = 10, 30$.

In this paper, we present a PSO-3P [14] based algorithm that uses three phases to guide the search through the solution space. In order to test the performance of PSO-3P, it was applied to 47 benchmark continuous unconstrained

optimization problems, over a total of 82 instances. After some computational experiments, we observed that PSO-3P is able to escape from suboptimal entrapments.

In particular, the proposed PSO-3P algorithm was able to reach the global optimum for the Griewank function with 120,000 variables, $x_i \in [-600, 600]$, in 40 seconds in average,

TABLE 2: Time and efficiency.

	Function(dimension)	Part.	Mean Eff.	Time (sec)	Mean Iter.
43	Rastrigin(100)	24	1.000000	2.07	262.50
44	Rastrigin(3)	24	0.999920	5.56	542.21
45	Rastrigin(30)	24	1.000000	2.03	265.33
46	Rastrigin(6)	24	1.000000	1.84	245.08
47	Rosenbrock(10)	24	1.000000	1.64	218.04
48	Rosenbrock(100)	24	0.966926	2.32	674.92
49	Rosenbrock(2)	24	1.000000	0.24	26.33
50	Rosenbrock(30)	12	0.999706	1.68	258.35
51	Rosenbrock(6)	24	1.000000	2.77	241.67
52	Rotated Hyper-Ellipsoid(12)	3	1.000000	1.21	361.04
53	Rotated Hyper-Ellipsoid(120)	3	1.000000	1.25	323.78
54	Rotated Hyper-Ellipsoid(2)	240	1.000000	4.66	20.33
55	Schaffer2(2)	240	1.000000	4.68	21.17
56	Schwefel 26(100)	120	0.924108	16.26	490.50
57	Schwefel 26(2)	120	0.994103	15.93	499.00
58	Shekel(4)	60	0.999580	5.36	322.50
59	Shubert(2)	240	1.000000	4.45	14.08
60	Six-Hump Camel(2)	12	0.999974	2.73	499.00
61	Sphere(12)	3	1.000000	1.02	303.42
62	Sphere(120)	3	1.000000	1.08	315.79
63	Sphere(2)	240	1.000000	4.58	16.96
64	Styblinski-Tang(12)	3	0.999900	1.81	499.00
65	Styblinski-Tang(120)	3	0.999884	1.96	499.00
66	Styblinski-Tang(2)	24	0.999905	0.24	25.17
67	Sum of Different Powers(12)	3	1.000000	0.73	213.79
68	Sum of Different Powers(120)	3	1.000000	0.98	277.65
69	Sum of Different Powers(2)	240	1.000000	15.37	193.75
70	Sum Squares(12)	3	1.000000	0.91	262.38
71	Sum Squares(120)	3	1.000000	1.03	297.25
72	Sum Squares(2)	24	1.000000	0.16	15.38
73	Three-Hump Camel(2)	240	1.000000	4.54	17.29
74	Trid(10)	48	0.993918	10.11	499.00
75	Trid(6)	240	0.996842	32.12	499.00
76	Xin She Yang02(12)	48	1.000000	5.18	157.92
77	Xin She Yang02(120)	48	1.000000	6.86	254.54
78	Xin She Yang02(2)	48	1.000000	4.83	148.83
79	Xin She Yang03(2)** (1.08)	3	1.000000	0.55	160.33
80	Zakharov(12)	3	1.000000	1.15	357.21
81	Zakharov(120)	3	0.999958	1.35	430.13
82	Zakharov(2)	240	1.000000	4.55	17.75

**The overall success of different global optimizers according to Gavana [18].

using only 3 particles and 90 iterations (see Figures 1(a) and 1(b)). This instance was solved using Matlab and run on a Notebook with an Intel Atom N280 processor at 1.66 Ghz.

The remainder of this paper is divided as follows: a background of PSO is presented in the following section. The general guidelines of PSO-3P are described in Section 3. Numerical examples are provided in Section 4. Finally, Section 5 includes conclusions and future research.

2. PSO

The particle swarm optimization is a metaheuristic based on swarm intelligence and has its roots in artificial life, social psychology, engineering, and computer science. PSO differs from evolutionary computation (c.f. [15]) because the population members or agents, also called particles, are “flying” through the problem hyperspace.

TABLE 3: Results with 3 and 6 particles.

	Function(Dim.)	Part.	Time (sec)	Perc.	Mean Iter.	EOF
1	Ackley(10)	3	1.4	20.8	455.6	1366.8
2	Ackley(120)	3	1.5	62.5	385.5	1156.5
3	Ackley(2)	3	1.2	37.5	379.2	1137.6
4	Bohachevsky 1(2)	3	0.8	100	236.9	710.7
5	Bohachevsky 2(2)	3	0.8	100	241.8	725.4
6	Bohachevsky 3(2)	3	0.8	100	247.5	742.5
7	Branin(2)	6	3.4	4.2	484.4	2906.4
8	Colville(4)	3	1.6	4.2	485	1455
9	CrossLegTable(2) (0.83)	3	0.8	75	223.6	670.8
10	Cross-in-Tray(2)	3	1.7	16.7	460	1380
11	DevillierGlasser02(5) (0.00)	3	0.6	100	151.9	455.7
12	Drop Wave(2)	3	1.2	100	142	426
13	Easom(2)	3	1.5	8.3	466.5	1399.5
14	Goldstein-Price(2)	3	1.6	4.2	483	1449
15	Griewank(10)	3	0.3	100	88.5	265.5
16	Griewank(100)	3	0.4	100	88.3	264.9
17	Griewank(2)	3	0.7	100	209.5	628.5
18	Griewank(30)	3	0.9	75	193.7	581.1
19	Griewank(6)	3	0.3	100	88.2	264.6
20	Hartmann(3)	3	1.8	4.2	485.5	1456.5
21	Holder Table(2)	6	2.6	4.2	478.9	2873.4
22	Levy(12)	6	2.4	8.3	475.5	2853
23	Levy(120)	6	2.9	12.5	474.8	2848.8
24	Levy(2)	6	2.3	8.3	465.3	2791.8
25	Levy13(2)	6	2.4	4.2	484.3	2905.8
26	Matyas(2)	3	0.8	100	237.6	712.8
27	Michalewicz(2)	3	2	54.2	367	1101
28	Powell(10)	3	1.7	4.2	497.3	1491.9
29	Powell(12)	3	0.6	95.8	200.3	600.9
30	Powell(120)	3	0.7	100	207.6	622.8
31	Powell(2)	3	0.7	95.8	261.5	784.5
32	Rastrigin(10)	3	0.8	100	261.8	785.4

PSO is an adaptive method that uses agents or particles moving through the search space using the principles of evaluation, comparison, and imitation [15].

PSO is based on the use of a set of particles or agents that correspond to states of an optimization problem, where each particle moves across the solution space in search of an optimal position or at least a good solution. In PSO, agents communicate with each other, and the agent with the best position (measured according to an objective function) influences the others by attracting them towards itself.

The population is started by assigning an initial random position and speed for each element. At each iteration, the velocity of each particle is randomly accelerated towards its best position (where the value of the fitness function or objective function improves) and also considering the best positions of their neighbors.

To solve a problem, PSO uses a dynamic management of particles; this approach allows breaking cycles and diversifying the search. In this work, r -particle swarm is represented at time t under the form

$$\theta_{1t}, \theta_{2t}, \dots, \theta_{rt} \quad (2)$$

with $\theta_{jt} \in D$, $j = 1, 2, \dots, r$; then a movement of the swarm is defined according to

$$\theta_{j,t+1} = \theta_{jt} + V_{j,t+1}, \quad (3)$$

where the velocity $V_{j,t+1}$ is given in

$$\begin{aligned} V_{j,t+1} = & \alpha V_{jt} + \text{rand}(0, \varphi_1) [\theta'_{j,t} - \theta_{j,t}] \\ & + \text{rand}(0, \varphi_2) [\theta'_{g,t} - \theta_{j,t}], \end{aligned} \quad (4)$$

TABLE 4: Results considering 3 and 6 particles.

	Function(dim)	Part.	Time (sec)	Perc.	Mean Iter.	EOF
33	Rastrigin(100)	3	1.3	100	289.8	869.4
34	Rastrigin(3)	6	2.8	8.3	487.1	2922.6
35	Rastrigin(30)	3	1.3	95.8	291.4	874.2
36	Rastrigin(6)	6	1.8	100	272.5	1635
37	Rosenbrock(10)	3	1.1	66.7	331	993
38	Rosenbrock(2)	3	0.5	79.2	140.3	420.9
39	Rosenbrock(30)	3	1.2	50	379.2	1137.6
40	Rosenbrock(6)	3	0.3	100	87.5	262.5
41	Rotated Hyper-Ellipsoid(12)	3	0.6	100	179	537
42	Rotated Hyper-Ellipsoid(120)	3	0.9	95.8	221.5	664.5
43	Rotated Hyper-Ellipsoid(2)	3	0.8	100	231	693
44	Schaffer2(2)	3	0.7	100	221	663
45	Shekel(4)	3	0.8	100	216.5	649.5
46	Six-Hump Camel(2)	3	1.7	8.3	468	1404
47	Sphere(12)	3	0.5	95.8	162.6	487.8
48	Sphere(120)	3	0.5	95.8	166.9	500.7
49	Sphere(2)	3	0.7	100	217.4	652.2
50	Styblinski-Tang(2)	3	1.5	20.8	416.9	1250.7
51	Sum of Different Powers(12)	3	0.5	95.8	144.1	432.3
52	Sum of Different Powers(120)	3	0.5	100	128.3	384.9
53	Sum of Different Powers(2)	3	0.9	100	270.2	810.6
54	Sum Squares(12)	3	0.5	100	148.9	446.7
55	Sum Squares(120)	3	0.6	100	186.8	560.4
56	Sum Squares(2)	3	0.7	100	222.7	668.1
57	Three-Hump Camel(2)	3	0.7	100	209.8	629.4
58	Xin She Yang02(12)	3	1.2	95.8	351	1053
59	Xin She Yang02(120)	3	2.8	25	451.1	1353.3
60	Xin She Yang02(2)	3	0.6	95.8	170.1	510.3
61	Xin She Yang03(2) (1.08)	3	0.5	100	155.3	465.9
62	Zakharov(12)	3	0.6	95.8	178.3	534.9
63	Zakharov(120)	3	1.2	75	373.6	1120.8
64	Zakharov(2)	3	1.6	4.2	488.9	1466.7

where D is space of feasible solutions, $V_{j,t}$ is speed at time t of the j th particle, $V_{j,t+1}$ is speed at time $t + 1$ of the j th particle, $\theta_{j,t}$ is j th particle at time t , $\theta'_{g,t}$ is the particle with the best value found so far (i.e., before time t), $\theta'_{j,t}$ is best position found so far by the j th particle (before time t), $\text{rand}(0, \varphi)$ is random number uniformly distributed over the interval $[0, \varphi]$, and α is inertia weight factor.

The PSO algorithm is described in Algorithm 1.

3. PSO-3P

In this section, the main characteristics of the proposed algorithm, called PSO-3P, are described. The PSO-3P is based on a traditional PSO heuristic. However, the position of the particles can be modified using different strategies, which are sequentially applied in three phases of the searching process.

In phase 1, called stabilization, according to the description presented in Section 2, the PSO-3P algorithm generates

randomly a set of particles in the solution space. Then, during itF_1 iterations the position of the particles is modified using (3) and (4). Thus, at the end of this phase the particles are concentrated, or stabilized, in a promising region.

When phase 1 is completed, a breadth-first search strategy, called phase 2, is incorporated. In this phase, if the global best solution is not improved after c consecutive iterations, a random particle is created and, with a probability bigger than 0.5, takes the place of one particle randomly selected in the swarm. This process of creation and replacement is repeated $n\text{Pop} * \text{prop}$ times. However, the particle with the best known position is preserved. Thus, the population is dispersed in the solution space, but it can be attracted to the best region visited so far. This diversification strategy is considered during itF_2 iterations.

Finally, phase 3 is initialized. During itF_3 iterations the following depth-first search strategy is applied. If the global best solution is not improved after c consecutive iterations, $n\text{Pop} * \text{prop}$ particles are randomly created in a neighborhood

TABLE 5: Ackley problem.

Algorithm	Dim.	M. G/I	P.	EOF	Best	Average	Median	Time (s)
PSO-3P	10	500	3	515*	$7.99E - 15$	$7.56E - 05$	$7.99E - 15$	0.655
PSO-3P	30	500	3	482*	$7.99E - 15$	$1.40E - 05$	$7.99E - 15$	0.558
ELPSO [19]	30	100	1500	—	0.078	0.2279	0.3169	—
LDWPSO [19]	30	100	1500	—	0.0009	1.8896	2.0099	—
FIWPSO [19]	30	100	1500	—	1.1552	2.6922	2.4962	—
CIWPSO [19]	30	100	1500	—	1.3567	2.5694	2.4482	—
RIWPSO [19]	30	100	1500	—	0.9313	2.3145	2.3584	—
TVACPSO [19]	30	100	1500	—	1.157	2.1255	2.221	—
COPSO [19]	30	100	1500	—	2.1201	2.9663	2.8098	—
IRPEO [13]	30	50000	50	—	$1.15E - 04$	$2.91E - 04$	—	—
pPSA [20]	30	2000	30	—	$1.52E - 03$	$1.19E + 00$	$9.31E - 01$	—
PSO [21]	30	1000	—	—	$2.58E + 00$	—	—	—
SFLA [21]	30	1000	—	—	$2.40E + 00$	—	—	1.83
MSFLA [21]	30	1000	—	490000	$2.10E - 01$	—	—	6.25
MSFLA-EO [21]	30	1000	—	2900000	$0.00E + 00$	—	—	—
PSO-EO [22]	30	10000	30	—	$9.50E - 04$	$9.88E - 04$	—	0.26
LX-PM [23]	30	5000	300	86661	—	$1.01E - 10$	—	1.625
LX-MPTM [23]	30	5000	300	116731	—	$2.18E - 07$	—	2.243
LX-NUM [23]	30	5000	300	130041	—	$6.66E - 07$	—	2.473
HX-PM [23]	30	5000	300	178301	—	$1.01E - 10$	—	2.8
HX-MPTM [23]	30	5000	300	196401	—	$3.17E - 04$	—	3.565
HX-NUM [23]	30	5000	300	232901	—	$1.28E - 03$	—	4.134
ADM [24]	30	2348*	—	—	—	$3.28E - 01$	—	91.25
RM [24]	30	29824*	—	—	—	$1.42E - 03$	—	3389.03
PLM [24]	30	30000*	—	—	—	$3.47E - 04$	—	3053.57
NUM [24]	30	29590*	—	—	—	$5.91E - 04$	—	874.63
MNUM [24]	30	17290*	—	—	—	$1.82E - 01$	—	453.05
PM [24]	30	25554*	—	—	—	$4.46E - 01$	—	933.23
PSO-3P	50	500	3	511*	$7.99E - 15$	$9.60E - 05$	$7.99E - 15$	0.632
CSA [25]	50	2000	—	4000	—	$5.00E - 01$	—	—
ACSA [25]	50	2000	—	2500	—	$8.71E - 06$	—	—
PSO-3P	100	500	3	952*	$8.88E - 16$	$8.88E - 16$	$8.88E - 16$	1.040
ELPSO [19]	100	100	5000	—	0.0475	0.2255	0.1589	—
CPSO [19]	100	—	—	—	3.6588	4.7431	4.6822	—
HS [19]	100	—	—	—	12.876	13.3999	13.5271	—
GA [19]	100	—	—	—	10.2225	10.7037	10.5791	—
FSO [19]	100	—	—	—	0.0256	0.0275	0.0273	—
GSA [19]	100	—	—	—	1.7456	11.4327	6.0956	—
BSOA [19]	100	—	—	—	6.4311	8.0122	7.9955	—
ABC [19]	100	—	—	—	17.1928	17.5914	17.5324	—
PSO-3P	120	500	3	947*	$7.99E - 15$	$6.22E - 07$	$7.99E - 15$	1.638
PSO-3P	1000	500	3	470*	$7.99E - 15$	$4.52E - 05$	$7.99E - 15$	0.555

* Average.

of the best known solution and take the place of equal number of randomly selected particles in the swarm. Thus, phase 3 includes an intensification process in a promising region.

The main steps of the PSO-3P algorithm are described in Algorithm 2.

4. Computational Results

In order to evaluate the performance of the proposed PSO-3P algorithm, 47 unconstrained optimization problems taken

from [16–18] were used. An in-depth study for multiobjective optimization, and constrained optimization, will be presented in a future work.

These problems include functions that are scalable to arbitrary dimensions such as Ackley, Rastrigin, Rosenbrock, Sphere, and Zakharov. So far, these problems have been widely used as benchmarks for study with different methods by many researchers; see [13, 19–25]. Also functions that are especially difficult to solve were included; for example, according to [18], the overall success of different global

FIGURE 3: Wilcoxon sum rank test, where ELIPSO is represented by 1 [19], LDWIPSO is represented by 2 [19], FIWPSO is represented by 3 [19], CIWPSO is represented by 4 [19], RIWPSO is represented by 5 [19], TVACPSO is represented by 6 [19], COPSO is represented by 7 [19], IRPEO is represented by 8 [13], pPSA is represented by 9 [20], PSO is represented by 10 [21], SFLA is represented by 11 [21], MSFLA is represented by 12 [21], MSFLA-EO is represented by 13 [21], PSO-EO is represented by 14 [22], LX-PM is represented by 15 [23], LX-MPTM is represented by 16 [23], LX-NUM is represented by 17 [23], HX-PM is represented by 18 [23], HX-MPTM is represented by 19 [23], HX-NUM is represented by 20 [23], ADM is represented by 21 [24], RM is represented by 22 [24], PLM is represented by 23 [24], NUM is represented by 24 [24], MNUM is represented by 25 [24], PM is represented by 26 [24], CSA is represented by 27 [25], ACSA is represented by 28 [25], ELPSO is represented by 29 [19], CPSO is represented by 30 [19], HS is represented by 31 [19], GA is represented by 32 [19], FSO is represented by 33 [19], GSA is represented by 34 [19], BSOA is represented by 35 [19], and ABC is represented by 36 [19].

optimizers on DeVilliersGlasser02, Damavandi, CrossLeg-Table, XinSheYang03, Griewank, and XinSheYang02 was 0%, 0.25%, 0.83%, 1.08%, 6.08%, and 31.33%, respectively; see Figure 2.

In order to evaluate the performance of the algorithm we used the following efficiency measure:

efficiency

$$= \begin{cases} 1 - \text{abs}\left(\frac{h(x) - h(x^*)}{h(x^*)}\right) & \text{if } h(x^*) \neq 0 \\ 1 - \text{abs}(h(x)) & \text{if } h(x^*) = 0, \end{cases} \quad (5)$$

where x^* is the optimum of function h and x is the best solutions found by the algorithm after one run. When the efficiency was greater than 0.999999, it was rounded to 1.

The tuning of the operating parameters was realized using a brute-force approach, as described in [26]. Finally,

FIGURE 4: P values of the Wilcoxon sum rank test, where ELIPSO is represented by 1 [19], LDWIPSO is represented by 2 [19], FIWPSO is represented by 3 [19], CIWPSO is represented by 4 [19], RIWPSO is represented by 5 [19], TVACPSO is represented by 6 [19], COPSO is represented by 7 [19], IRPEO is represented by 8 [13], pPSA is represented by 9 [20], PSO is represented by 10 [21], SFLA is represented by 11 [21], MSFLA is represented by 12 [21], MSFLA-EO is represented by 13 [21], PSO-EO is represented by 14 [22], LX-PM is represented by 15 [23], LX-MPTM is represented by 16 [23], LX-NUM is represented by 17 [23], HX-PM is represented by 18 [23], HX-MPTM is represented by 19 [23], HX-NUM is represented by 20 [23], ADM is represented by 21 [24], RM is represented by 22 [24], PLM is represented by 23 [24], NUM is represented by 24 [24], MNUM is represented by 25 [24], PM is represented by 26 [24], CSA is represented by 27 [25], ACSA is represented by 28 [25], ELPSO is represented by 29 [19], CPSO is represented by 30 [19], HS is represented by 31 [19], GA is represented by 32 [19], FSO is represented by 33 [19], GSA is represented by 34 [19], BSOA is represented by 35 [19], ABC is represented by 36 [19], and PSO-3P is represented by 37.

the parameters were set as follows: $\phi_1 = \phi_2 = 2.05$ and $\phi = \phi_1 + \phi_2$; then,

$$\alpha = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4 * \phi}}; \quad (6)$$

$$\varphi_1 = \alpha * \phi_1;$$

$$\varphi_2 = \alpha * \phi_2.$$

For phase 2 and phase 3 the parameters were set as $c = 3$, $itF_1 = 75$, $itF_2 = 150$, $itF_3 = \text{Max It}$, and $\text{prop} = 0.25$.

The algorithm was implemented in Matlab R2008a and was run in an Intel Core i5-3210M processor computer at 2.5 GHz, running on Windows 8.

4.1. Unconstrained Optimization. In this section, we present the results obtained for problems with 120 or less variables. Data reported in each of the following tables derive from 24 independent executions of the algorithm for each function. The algorithm stops when a maximum of 1500 iterations has been done or the optimum has been found. In order to get a good mean efficiency, the number of particles used for each

- (1) Begin.
- (2) **while** *Termination criterion is not satisfied* **do**
- (3) Create a population of particles distributed in the feasible space.
- (4) Evaluate each position of the particles according to the objective function (fitness function).
- (5) If the current position of a particle is better than the previous one, update it.
- (6) Determine the best particle (according to the best previous positions).
- (7) Update the particle velocities $j = 1, 2, \dots, r$ according to (4).
- (8) Move the particles to new positions according to (3).
- (9) **end**

ALGORITHM 1: PSO algorithm.

- (1) Begin.
- (2) **while** *Termination criterion is not satisfied* **do**
- (3) Set variables c , itF_1 , itF_2 , itF_3 , $MaxIt$ and $prop$.
- (4) Create a population of $nPop$ random particles.
- (5) Set $cont = 0$ and $it = 1$. Evaluate each position of the particles according to the fitness function.
- (6) If the current position of a particle is better (respect to the fitness function) than the previous update it.
- (7) Determine the best particle (according to the best previous positions against the optimization criterion).
If a better particle cannot be founded, let $cont = cont + 1$.
- (8) Update the particle velocities $j = 1, 2, \dots, nPop$ according to (4).
- (9) (Phase 1: Stabilization) **if** $it \leq itF_1$ **then**
- (10) go to Step (34).
- (11) **end**
- (12) (Phase 2: Breadth-first search) **if** $itF_1 < it \leq itF_2$ **then**
- (13) **if** $cont = c$ **then**
- (14) Set $n = 1$. **while** $n \leq nPop * prop$. **do**
- (15) Create a random particle and, with a probability bigger than 0.5 substitute randomly a particle in the swarm.
- (16) Set $n = n + 1$.
- (17) **end**
- (18) Set $cont = 0$.
- (19) **end**
- (20) go to Step (34).
- (21) **end**
- (22) **end**
- (23) (Phase 3: Depth-first search). **if** $itF_2 < it \leq itF_3$ **then**
- (24) **if** $cont = c$ **then**
- (25) Set $n = 1$. **while** $n \leq nPop * prop$. **do**
- (26) Create a random particle in a variable neighborhood of $\theta'_{g,t}$ and substitute randomly a particle in the swarm.
- (27) Set $n = n + 1$.
- (28) **end**
- (29) **end**
- (30) Set $cont = 0$.
- (31) go to Step (34).
- (32) **end**
- (33) Select the best $nPop$ particles according to optimization criterion.
- (34) Set $it = it + 1$. Go to Step (3) until the termination criterion is satisfied.

ALGORITHM 2: PSO-3P algorithm.

function was tuned using a brute-force approach; thus it can range between 3 and 240 particles for different problems.

Tables 1 and 2 include, for each function, the number of particles (Part.), the average efficiency (Mean Eff.) using (5), the average time per run in seconds (Time (sec)), and the average number of iterations per run (Mean Iter.).

The performance of the PSO-3P is remarkable for DeVilliersGlasser02, Damavandi, and CrossLegTable functions, where the success rate was superior to those reported in [18].

4.2. Experiments Using Few Particles. As an additional study on the behavior of the proposed PSO-3P, results obtained

TABLE 6: Griewank problem.

Algorithm	Dim.	M. G/I	P.	EOF	Best	Average	Median	Time (s)
PSO-3P	10	500	3	282*	0	0	0	0.338
PSO-3P	30	500	3	513*	0	$1.61E-05$	0	0.86
ELIPSO [19]	30	100	1500	—	$1.63E-04$	$2.27E-04$	$2.55E-04$	—
LDWIPSO [19]	30	100	1500	—	0.0002	0.0135	0.0089	—
FIWPSO [19]	30	100	1500	—	0.0007	0.0118	0.0097	—
CIWPSO [19]	30	100	1500	—	0.0015	0.0124	0.0119	—
RIWPSO [19]	30	100	1500	—	0	0.0114	0.0083	—
TVACPSO [19]	30	100	1500	—	0	0.0209	0.0148	—
COPSO [19]	30	100	1500	—	0	0.0136	0.0136	—
IRPEO [13]	30	50000	50	—	$1.04E-05$	$1.92E-05$	—	—
pPSA [20]	30	2000	30	—	$3.12E-07$	$1.21E-02$	$9.86E-03$	—
PSO [21]	30	100	—	170000	—	0	—	0.52
SFLA [21]	30	100	—	120000	—	$1.00E-02$	—	0.45
MSFLA [21]	30	100	—	70000	—	0	—	0.28
MSFLA-EO [21]	30	100	—	50000	—	0	—	0.21
PSO-EO [22]	30	20000	30	—	0	0	—	0.23
LX-PM [23]	30	5000	300	100001	—	$1.94E-03$	—	1.851
LX-MPTM [23]	30	5000	300	143371	—	$1.14E-03$	—	2.723
LX-NUM [23]	30	5000	300	159971	—	$8.57E-03$	—	3.035
HX-PM [23]	30	5000	300	236391	—	$1.94E-03$	—	3.554
HX-MPTM [23]	30	5000	300	253931	—	$1.52E-03$	—	4.486
HX-NUM [23]	30	5000	300	252571	—	$3.68E-03$	—	4.457
ADM [24]	30	519*	—	—	—	$4.43E-03$	—	20.47
RM [24]	30	28743*	—	—	—	$1.36E-02$	—	3278.67
PLM [24]	30	277288*	—	—	—	$5.74E-03$	—	2987.69
NUM [24]	30	27915*	—	—	—	$2.45E-02$	—	819.56
MNUM [24]	30	15118*	—	—	—	$3.66E-02$	—	478.1
PM [24]	30	28864*	—	—	—	$1.02E+00$	—	900.25
PSO-3P	50	500	3	501*	0	$9.54E-06$	0	0.646
PSO [21]	50	100	—	1300000	—	0	—	4.32
SFLA [21]	50	100	—	3400000	—	$1.00E-02$	—	11.98
MSFLA [21]	50	100	—	460000	—	0	—	1.91
MSFLA-EO [21]	50	100	—	820000	—	0	—	2.06
CSA [25]	50	2000	—	4000	—	$1.01E+00$	—	—
ACSA [25]	50	2000	—	2500	—	$7.00E-01$	—	—
PSO-3P	100	500	3	485*	0	$1.44E-06$	0	0.589
ELPSO [19]	100	100	5000	—	0.1846	0.2612	0.2127	—
CPSO [19]	100	—	—	—	0.2347	0.2841	0.2793	—
HS [19]	100	—	—	—	244.5269	271.336	268.3095	—
GA [19]	100	—	—	—	86.0552	104.867	108.7177	—
FSO [19]	100	—	—	—	0.0136	0.0152	0.0151	—
GSA [19]	100	—	—	—	0.2126	1.1321	0.6543	—
BSOA [19]	100	—	—	—	60.9781	83.5311	84.5294	—
ABC [19]	100	—	—	—	722.3432	817.3863	844.5736	—
PSO-3P	120	500	3	498*	0	$1.44E-07$	0	0.624
PSO-3P	1000	500	3	507*	0	$1.18E-04$	0	0.617

* Average.

TABLE 7: Rastrigin problem.

Algorithm	Dim.	M. G/I	P.	EOF	Best	Average	Median	Time (s)
PSO-3P	10	500	3	878*	0	$4.72E - 02$	0	1.101
PSO-3P	30	500	3	510*	0	$3.21E - 07$	0	0.584
ELIPSO [19]	30	100	1500	—	3.8941	8.6403	8.8185	—
LDWIPSO [19]	30	100	1500	—	13.9294	25.67	26.8639	—
FIWPSO [19]	30	100	1500	—	22.8841	29.5503	28.3564	—
CIWPSO [19]	30	100	1500	—	7.9597	19.0042	19.8992	—
RIWPSO [19]	30	100	1500	—	11.9397	25.5707	25.3715	—
TVACPSO [19]	30	100	1500	—	9.9496	24.7745	25.3714	—
COPSO [19]	30	100	1500	—	30.8437	51.687	47.758	—
IRPEO [13]	30	50000	50	—	$2.47E - 05$	$6.19E - 05$	—	—
pPSA [20]	30	2000	30	—	$4.78E + 01$	$8.11E + 01$	$8.21E + 01$	—
PSO-EO [22]	30	20000	10	—	0	$0.00E + 00$	—	0.61
LX-PM [23]	30	5000	300	165471	—	$0.00E + 00$	—	2.997
LX-MPTM [23]	30	5000	300	350541	—	$1.23E - 12$	—	6.515
LX-NUM [23]	30	5000	300	—	—	$7.30E + 00$	—	—
HX-PM [23]	30	5000	300	167851	—	$0.00E + 00$	—	2.441
HX-MPTM [23]	30	5000	300	245281	—	$9.22E - 12$	—	4.219
HX-NUM [23]	30	5000	300	426651	—	$3.11E - 13$	—	7.358
ADM [24]	30	1456*	—	—	—	$1.36E + 01$	—	57.03
RM [24]	30	12978*	—	—	—	$1.30E - 02$	—	1638.42
PLM [24]	30	10467*	—	—	—	$4.31E - 03$	—	971.72
NUM [24]	30	16410*	—	—	—	$4.43E - 02$	—	516.03
MNUM [24]	30	6468*	—	—	—	$1.80E + 01$	—	214.95
PM [24]	30	20712*	—	—	—	$8.23E + 00$	—	709.02
PSO-3P	50	500	3	413*	0	$1.63E - 07$	0	0.482
PSO [21]	50	10000	—	—	—	$3.14E + 01$	—	—
SFLA [21]	50	10000	—	—	—	$5.96E + 00$	—	—
MSFLA [21]	50	10000	—	10000000	—	$2.00E - 03$	—	19.72
MSFLA-EO [21]	50	10000	—	46000000	—	$0.00E + 00$	—	9.41
CSA [25]	50	2000	—	4000	—	$9.72E - 07$	—	—
ACSA [25]	50	2000	—	2500	—	$8.56E - 09$	—	—
PSO-3P	100	500	3	510*	0	$8.48E - 05$	0	0.595
ELPSO [19]	100	100	5000	—	1.0448	5.5402	5.7829	—
CPSO [19]	100	—	—	—	49.8618	78.782	82.187	—
HS [19]	100	—	—	—	300.2257	415.5759	360.5646	—
GA [19]	100	—	—	—	720.2649	768.2887	775.294	—
FSO [19]	100	—	—	—	169.1787	214.1549	218.9371	—
GSA [19]	100	—	—	—	27.8589	29.8488	29.8488	—
BSOA [19]	100	—	—	—	135.4258	164.6546	146.3873	—
ABC [19]	100	—	—	—	735.1485	753.3054	756.6776	—
PSO-3P	120	500	3	466*	0	$9.16E - 06$	0	0.372
PSO-3P	1000	500	3	516*	0	$1.28E - 04$	0	0.4

* Average.

with small populations are investigated in this section. Indeed, for some instances, the global optimum was found using only 3 or 6 particles and, in average, less than 3000 evaluations of the objective function.

In this case, the algorithm was executed 24 times, and it stops when a maximum of 3000 iterations per run has been done or when the optimum has been found. Tables 3 and 4

include the functions that were solved to optimality, at least in one run among the 24. These tables show the name of the function in column 2. The number of particles used for each case is presented in column 3. Column 4 includes the average time per run in seconds. The percentage of runs in which the global optimum was obtained is included in column 5. The average number of iterations per run is displayed in column

TABLE 8: Rosenbrock problem.

Algorithm	Dim.	M. G/I	P.	EOF	Best	Average	Median	Time (s)
PSO-3P	10	500	3	587*	0	$3.71E - 01$	0	0.69
PSO-3P	30	500	3	522*	0	$1.22E - 02$	0	0.61
ELIPSO [19]	30	100	1500	—	1.1337	5.8172	5.8938	—
LDWIPSO [19]	30	100	1500	—	4.4676	52.7099	29.2173	—
FIWPSO [19]	30	100	1500	—	26.9238	52.5216	49.8166	—
CIWPSO [19]	30	100	1500	—	0.0186	8.6884	0.561	—
RIWPSO [19]	30	100	1500	—	0.2243	28.3862	24.8883	—
TVACPSO [19]	30	100	1500	—	$2.40E - 03$	$3.93E + 01$	$2.73E + 01$	—
COPSO [19]	30	100	1500	—	$1.28E + 00$	$4.55E + 01$	45.7893	—
pPSA [20]	30	2000	30	—	$1.43E + 01$	$6.37E + 01$	$2.65E + 01$	—
IRPEO [13]	30	50000	50	—	$5.36E - 02$	$8.64E - 01$	—	—
PSO-EO [22]	30	100000	30	—	$9.50E - 04$	$9.88E - 04$	—	1171
LX-PM [23]	30	5000	300	—	—	$1.58E + 01$	—	—
LX-MPTM [23]	30	5000	300	—	—	$1.85E + 01$	—	—
X-NUM [23]	30	5000	300	—	—	$1.85E + 01$	—	—
HX-PM [23]	30	5000	300	669751	—	$1.84E + 01$	—	5.453
HX-MPTM [23]	30	5000	300	—	—	$1.61E + 01$	—	—
HX-NUM [23]	30	5000	300	1368691	—	$1.25E + 01$	—	14.621
ADM [24]	30	30000*	—	—	—	$6.58E + 00$	—	41.46
RM [24]	30	29863*	—	—	—	$6.98E + 01$	—	1491.56
PLM [24]	30	29625*	—	—	—	$5.13E + 01$	—	647.87
NUM [24]	30	30000*	—	—	—	$6.67E + 01$	—	379.56
MNUM [24]	30	30000*	—	—	—	$4.03E + 00$	—	173.61
PM [24]	30	30000*	—	—	—	$4.04E + 03$	—	404.41
PSO-3P	40	40	3	953*	0	$1.91E - 01$	$0.00E + 00$	1.1
PSO [21]	40	20000	—	—	—	$4.02E + 01$	—	—
SFLA [21]	40	20000	—	—	—	$5.28E + 00$	—	—
MSFLA [21]	40	20000	—	—	—	$4.00E - 03$	—	—
MSFLA-EO [21]	40	20000	—	53000000	—	$2.00E - 03$	—	359
PSO-3P	50	500	3	565*	0	$2.02E + 00$	0	0.604
CSA [25]	50	2000	—	4000	—	$2.94E + 00$	—	—
ACSA [25]	50	2000	—	2500	—	$1.75E + 00$	—	—
PSO-3P	100	500	3	553*	0	$4.09E + 00$	$0.00E + 00$	0.65
ELPSO [19]	100	100	5000	—	3.7441	8.739	8.3733	—
CPSO [19]	100	—	—	—	70.9597	209.9321	225.3372	—
HS [19]	100	—	—	—	$2.94E + 05$	$3.19E + 05$	$3.17E + 05$	—
GA [19]	100	—	—	—	$2.13E + 04$	$2.61E + 04$	$2.58E + 04$	—
FSO [19]	100	—	—	—	96.4872	97.8001	97.7481	—
GSA [19]	100	—	—	—	93.0005	93.0768	93.0122	—
BSOA [19]	100	—	—	—	156.385	206.5988	215.7586	—
ABC [19]	100	—	—	—	$1.25E + 06$	$1.38E + 06$	$1.28E + 06$	—
PSO-3P	120	500	3	599*	0	$4.91E + 00$	0	0.69
PSO-3P	1000	750	3	1948*	0	$3.57E - 01$	$2.14E - 02$	2.576

* Average.

6. Finally, the number of evaluations of the objective function, EOF, is presented in column 7.

Even for this case, when the algorithm was restricted to use few particles and iterations, the success rate was satisfactory for DeVilliersGlasser02 and CrossLegTable functions. It is worth remembering that the number of particles, or iterations, reported in Table 1 is higher, since the data

presented in the previous section was generated after tuning the parameters of the algorithm for each function, in order to increase its efficiency.

4.3. High Dimensional Problems. Finally, results obtained for some scalable functions are presented in this section, in

TABLE 9: Sphere problem.

Algorithm	Dim.	M. G/I	P.	EOF	Best	Average	Median	Time (s)
PSO-3P	10	500	3	483*	0	4.75E-08	0	0.569
PSO-3P	30	500	3	489*	0	1.51E-08	0	0.56
ELPSO [19]	30	100	1500	—	2.979E-08	5.244E-08	5.170E-06	—
LDWIPSO [19]	30	100	1500	—	1.360E-08	6.110E-08	5.920E-08	—
FIWPSO [19]	30	100	1500	—	7.280E-08	1.701E-07	1.173E-07	—
CIWPSO [19]	30	100	1500	—	5.400E-08	1.565E-07	1.670E-07	—
RIWPSO [19]	30	100	1500	—	1.600E-08	4.810E-07	6.900E-08	—
TVACPSO [19]	30	100	1500	—	4.430E-10	1.077E-09	1.046E-09	—
COPSO [19]	30	100	1500	—	5.080E-09	9.550E-09	9.470E-09	—
IRPEO [13]	30	50000	50	—	1.88E-01	2.96E-01	—	—
pPSA [20]	30	2000	30	—	5.60E-06	7.88E-06	7.93E-06	—
PSO-EO [22]	30	20000	10	—	0	0.00E+00	—	0.61
LX-PM [23]	30	5000	300	43541	—	4.75E-11	—	0.505
LX-MPTM [23]	30	5000	300	59861	—	1.32E-07	—	0.724
LX-NUM [23]	30	5000	300	67661	—	3.34E-11	—	0.819
HX-PM [23]	30	5000	300	89751	—	4.75E-11	—	0.733
HX-MPTM [23]	30	5000	300	87031	—	5.48E-05	—	0.944
HX-NUM [23]	30	5000	300	107581	—	3.64E-04	—	1.165
ADM [24]	30	285*	—	—	—	0	—	11.45
RM [24]	30	29500*	—	—	—	3.31E-06	—	2577.08
PLM [24]	30	29678*	—	—	—	5.62E-07	—	2538.16
NUM [24]	30	29788*	—	—	—	1.13E-06	—	899.87
MNUM [24]	30	11585*	—	—	—	0.00E+00	—	336.02
PM [24]	30	28498*	—	—	—	1.20E-02	—	994.49
PSO-3P	50	500	3	469*	0	3.03E-08	0	0.539
PSO-3P	100	500	3	481*	0	1.27E-07	0	0.569
ELPSO [19]	100	100	5000	—	4.297E-05	6.035E-05	5.811E-05	—
CPSO [19]	100	—	—	—	2.708E-04	5.398E-04	4.095E-04	—
HS [19]	100	—	—	—	69.0111	76.0844	78.0859	—
GA [19]	100	—	—	—	23.7734	29.4081	31.2806	—
FSO [19]	100	—	—	—	2.001E-04	2.211E-04	2.216E-04	—
GSA [19]	100	—	—	—	0.0562	0.0672	0.0691	—
BSOA [19]	100	—	—	—	0.3406	0.3696	0.3720	—
ABC [19]	100	—	—	—	198.6946	230.8250	237.4058	—
PSO-3P	120	500	3	418*	0	1.39E-08	0	0.492
PSO-3P	1000	500	3	486*	0	3.64E-08	0	0.599

* Average.

order to investigate the performance of PSO-3P on high-dimensional problems. The test functions used in this framework are Ackley, Griewank, Rastrigin, Rosenbrock, Sphere, and Zakharov. The experiments include dimensions 10, 30, 50, 100, 120, and 1000. Results reported by other authors, as far as we know, only include up to 100 dimensions.

Tables 5–10 highlight the results of PSO-3P and different algorithms for these functions. The name of the algorithm is included in the first column; the second column shows the dimension of the problem (Dim.). The number of generations (G), or iterations (I), is incorporated in column three. The fourth column includes the number of particles; the number of evaluations of the objective (EOF), or fitness, function is

presented in column five. The sixth column shows the best value found by the algorithm; the average and median value are presented in columns seven and eight, respectively; finally, the last column includes the running time in seconds. We must remark that not all this information was reported in the reviewed literature; thus some cells remain empty.

For Ackley's function, see Table 5; the PSO-3P always found solutions very close to the global optimum; besides the median was very close to the global optimum for all dimensions (10, 30, 50, 100, 120, and 1000). Additionally, for the Ackley function with 30 dimensions, PSO-3P required less than 50% of the number of evaluations of the objective function compared to the other algorithms.

TABLE 10: Zakharov problem.

Algorithm	Dim.	M. G/I	P.	EOF	Best	Average	Median	Time (s)
PSO-3P	10	500	3	507*	0	2.33E - 05	0	0.603
PSO-3P	30	500	3	542*	0	8.79E - 05	0	0.643
ELPSO [19]	30	100	1500	—	0.0001	0.0003	0.0002	—
LDWIPSO [19]	30	100	1500	—	0.0001	0.0005	0.0004	—
FIWPSO [19]	30	100	1500	—	0.0008	0.0032	0.0028	—
CIWPSO [19]	30	100	1500	—	0.001	0.003	0.0021	—
RIWPSO [19]	30	100	1500	—	0.0003	0.0019	0.0012	—
TVACPSO [19]	30	100	1500	—	4.95E - 05	1.36E - 04	1.37E - 04	—
COPSO [19]	30	100	1500	—	0.0524	6.9459	2.1362	—
IRPEO [13]	30	50000	50	—	1.88E - 01	5.59E - 01	—	—
LX-PM [23]	30	5000	300	63271	—	1.95E - 20	—	1.563
LX-MPTM [23]	30	5000	300	65711	—	8.04E - 15	—	1.642
LX-NUM [23]	30	5000	300	78681	—	3.32E - 10	—	1.969
HX-PM [23]	30	5000	300	92891	—	1.95E - 20	—	1.959
HX-MPTM [23]	30	5000	300	94701	—	7.86E - 09	—	2.243
HX-NUM [23]	30	5000	300	210791	—	2.86E - 01	—	5.06
ADM [24]	30	1769*	—	—	—	0	—	70.38
RM [24]	30	14286*	—	—	—	4.05E + 00	—	1249.5
PLM [24]	30	15252*	—	—	—	3.08E - 05	—	1306.32
NUM [24]	30	6536*	—	—	—	4.88E + 01	—	239.33
MNUM [24]	30	27744*	—	—	—	0	—	687.27
PM [24]	30	5454*	—	—	—	7.09E + 01	—	203.16
PSO-3P	50	500	3	649*	0	2.11E - 05	0	0.783
PSO-3P	100	500	3	814*	0	8.14E - 02	0	0.944
ELPSO [19]	100	100	5000	—	0.2268	2.5016	2.2814	—
CPSO [19]	100	—	—	—	17.7263	40.9341	39.9646	—
HS [19]	100	—	—	—	287.5234	2694.91	488.3287	—
GA [19]	100	—	—	—	369.4271	412.8078	382.7739	—
FSO [19]	100	—	—	—	62.8325	74.4212	75.6089	—
GSA [19]	100	—	—	—	167.7459	171.3504	168.1632	—
BSOA [19]	100	—	—	—	91.0035	53.6407	130.7713	—
ABC [19]	100	—	—	—	1052.2	1278.7	1337.6	—
PSO-3P	120	500	3	768*	0	2.22E - 02	0	0.932
PSO-3P	1000	500	3	1080*	0	9.66E + 07	0	1.268

* Average.

For the Griewank, Rastrigin, Rosenbrock, Sphere, and Zakharov functions, see Tables 6–10; the PSO-3P always found the global optimum in at least one run, and the median always was the global optimum for all dimensions (10, 30, 50, 100, 120, and 1000).

Special attention should be given to the number of iterations and evaluations of the objective, fitness, and function. In average, the PSO-3P required 609 evaluations of the objective function to achieve the global optimum for dimensions 10, 30, 50, 100, 120, and 1000 and less than 0.74 seconds.

The normalized solutions for 17 instances are shown in Table 11; in this table values range between 0 and 1. If an algorithm found a solution close to the best reported value, it is associated with 0. On the other hand, if the best solution found by an algorithm is close to the worst reported value,

it is associate with 1. Based on Table 11, we observe that PSO-3P found the best results in 94% of the instances and only in 6% get the second best result. Also, PSO-3P is a competitive alternative to solve Ackley, Griewank, Rastrigin, Rosenbrock, Sphere, and Zakharov problems.

Results of the Wilcoxon rank sum test are shown in Figure 3; in this case, a dark square means that the algorithms are statically similar, while a clear square means the opposite. Results of the Wilcoxon rank sum test 3 show that PSO-3P is a metaheuristic similar to PSO-EO [22], LX-PM [23], HX-PM [23], and ACSA [25]. On the other hand, our method is different from the remaining methods. However, PSO-3P required less number of evaluations of the objective function than its counterparts for getting good results. P values involved in the Wilcoxon rank sum test are shown in the Figure 4.

TABLE 11: Normalized Score (NS) versus Problem (P) and dimension (Dim.).

NS versus P	Ackley			Griewank			Rastrigin			Rosenbrock			Sphere			Zakharov		
Dim.	30	50	100	30	50	100	30	50	100	30	40	50	30	100	30	100	30	100
1	7	27	36	26	27	36	9	10	32, 36	26	10	27	8	36	26	31		
0.9	3, 4																	
0.8	5		31															
0.7	6					28												
0.6	2		32, 34				7											24
0.5			35															
0.4	9						3											36
0.3			30			31	2, 5, 6		33									
0.2	26						4, 21, 25	11	35			PSO-3P	31					
0.1	1, 21, 25					32, 35	1, 17, 26		30		11			32	7, 22	34		
	8, 14–20					29, 30	8, 14–20	12, 13				28	29, 30	1–7, 9	29, 30	1–6, 8	29, 30	
0	22–24	28	29, 33	1–25	10–13	33, 34	22–24	27, 28	29, 34	14–25	12, 13		32–35	14–26	33–35	15–23, 25	33, 35	
	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P	PSO-3P

Note: ELIPSO is represented by 1 [19]; LDWIPSO is represented by 2 [19]; FIWPSO is represented by 3 [19]; CIWPSO is represented by 4 [19]; RIWPSO is represented by 5 [19]; TVACPSO is represented by 6 [19]; COPSO is represented by 7 [19]; IRPEO is represented by 8 [13]; pPSA is represented by 9 [20]; PSO is represented by 10 [21]; SFLA is represented by 11 [21]; MSFLA is represented by 12 [21]; MSFLA-EO is represented by 13 [21]; PSO-EO is represented by 14 [22]; LX-PM is represented by 15 [23]; LX-PM is represented by 16 [23]; LX-PM is represented by 17 [23]; HX-PM is represented by 18 [23]; HX-MPTM is represented by 19 [23]; HX-PM is represented by 20 [23]; ADM is represented by 21 [24]; RM is represented by 22 [24]; PLM is represented by 23 [24]; NUM is represented by 24 [24]; MNUM is represented by 25 [24]; PM is represented by 26 [24]; CSA is represented by 27 [25]; ACSA is represented by 28 [25]; ELPSO is represented by 29 [19]; CPSO is represented by 30 [19]; HS is represented by 31 [19]; GA is represented by 32 [19]; FSO is represented by 33 [19]; GSA is represented by 34 [19]; BSOA is represented by 35 [19]; ABC is represented by 36 [19].

5. Conclusions and Further Research

In this work, a novel PSO based algorithm is with three phases: (1) stabilization, (2) breadth-first search, and (3) depth-first search. The resulting PSO-3P algorithm was tested over a set of single-objective unconstrained optimization benchmark instances. The empirical evidence shows that PSO-3P is very efficient.

Tables 5–10 highlight the fact that PSO-3P gives good results and competitive solutions for some difficult previously reported problems.

Moreover, for all benchmark functions considered in this work, PSO-3P converged faster and required fewer iterations than many specialized algorithms, regardless of the dimension of the function. PSO-3P used on average 609 evaluations of the objective function to converge to the global optimum, which represent on average 66% of the number of evaluations reported for these problems by other algorithms, and an average running time of 0.74 seconds for problems of dimension 10 to 1000. As a matter of fact, some solutions are presented for problems of high dimension (120, 1000) that have not been reported before. Also, the numerical results of the Wilcoxon test show that the results obtained by PSO-3P are similar to those reported by IRPEO, LX-MPTM, MNUM, pPSA, LX-NUM, PSO, HX-PM, CSA, SFLA, HX-MPTM, ACSA, MSFLA, HX-NUM, PSO, MSFLA-EO, ADM, PSO-EO, RM, MSFLA, LX-PM, PLM, MSFLA-EO, and NUM. However, the PSO-3P needs less than 1000 evaluations of the objective function for generating good results; in contrast, the other methods need more than 2500 evaluations.

It was observed that, in all the cases studied, PSO-3P can reach the global optimum or a solution can be very close to it, with small number of iterations, as well as the ability to jump deep valleys, for unconstrained optimization with 2 to 1,000 variables.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Sergio Gerardo de-los-Cobos-Silva would like to thank D. Sto. and P. V. Gpe. for their inspiration, and his family Ma, Ser, Mon, Chema, and his Flaquita for all their support.

References

- [1] F. Glover, "Tabu Search, part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.
- [2] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [4] F. Glover, "A template for scatter search and path relinking," in *Artificial Evolution*, vol. 1363 of *Lecture Notes in Computer Science*, pp. 1–51, Springer, Berlin, Germany, 1998.
- [5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.
- [6] D. Sedighzadeh and E. Masehian, "Particle swarm optimization methods, taxonomy and applications," *International Journal of Computer Theory and Engineering*, vol. 1, no. 5, pp. 486–502, 2009.
- [7] I. Muhammad, H. Rathiah, and A. K. Noor Elaiza, "An overview of particle swarm optimization variants," *Procedia Engineering*, vol. 53, pp. 491–496, 2013.
- [8] Y. Zhao, W. Zu, and H. Zeng, "A modified particle swarm optimization via particle visual modeling analysis," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 2022–2029, 2009.
- [9] C.-C. Chen, "Two-layer particle swarm optimization for unconstrained optimization problems," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 295–304, 2011.
- [10] W. F. Abd-El-Wahed, A. A. Mousa, and M. A. El-Shorbagy, "Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems," *Journal of Computational and Applied Mathematics*, vol. 235, no. 5, pp. 1446–1453, 2011.
- [11] H. Kahraman and N. Allahverdi, "Particle swarm optimization with flexible swarm for unconstrained optimization," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 1, no. 1, pp. 8–13, 2013.
- [12] J.-Y. Wu, "Solving unconstrained global optimization problems via hybrid swarm intelligence approaches," *Mathematical Problems in Engineering*, vol. 2013, Article ID 256180, 15 pages, 2013.
- [13] G.-Q. Zeng, K.-D. Lu, J. Chen et al., "An improved real-coded population-based extremal optimization method for continuous unconstrained optimization problems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 420652, 9 pages, 2014.
- [14] S. G. De-los-Cobos-Silva, "SC—system of convergence: theory and foundations," *Revista de Matemática: Teoría y Aplicaciones*, vol. 22, no. 2, pp. 341–367, 2015.
- [15] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, Morgan Kaufmann Publishers, San Diego, Calif, USA, 2001.
- [16] A. R. Hedar, "Global Optimization Test Problems," 2014, http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm.
- [17] S. Surjanovic and D. Bingham, "Optimization Test Problems," 2014, <http://www.sfu.ca/~ssurjano/optimization.html>.
- [18] A. Gavana, *Global Optimization Benchmarks and AMIGO*, 2014, http://infinity77.net/global_optimization/test_functions.html.
- [19] A. R. Jordehi, "Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems," *Applied Soft Computing Journal*, vol. 26, pp. 401–417, 2015.
- [20] Z. Xinchao, "A perturbed particle swarm algorithm for numerical optimization," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 119–124, 2010.
- [21] X. Li, J. Luo, M.-R. Chen, and N. Wang, "An improved shuffled frog-leaping algorithm with extremal optimisation for continuous optimisation," *Information Sciences*, vol. 192, pp. 143–151, 2012.
- [22] M.-R. Chen, X. Li, X. Zhang, and Y.-Z. Lu, "A novel particle swarm optimizer hybridized with extremal optimization," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 367–373, 2010.

- [23] K. Deep and M. Thakur, "A new mutation operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 211–230, 2007.
- [24] P.-H. Tang and M.-H. Tseng, "Adaptive directed mutation for real-coded genetic algorithms," *Applied Soft Computing Journal*, vol. 13, no. 1, pp. 600–614, 2013.
- [25] P. Ong, "Adaptive cuckoo search algorithm for unconstrained optimization," *The Scientific World Journal*, vol. 2014, Article ID 943403, 8 pages, 2014.
- [26] M. Birattari, *Tuning Metaheuristics: A Machine Learning Perspective*, Springer, 2009.