Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

On a hybrid particle swarm optimization algorithm

Sharandeep Singh *, Narinder Singh, S. B. Singh

Department of Mathematics, Punjabi University, Patiala, Punjab 147002, India

A R T I C L E I N F O

Article history:
Received 18 August 2016
Received in revised form
10 November 2016
Accepted 13 December 2016

Keywords:
Inertia Constant
Gbest and Pbest Performance
Velocity updating

A B S T R A C T

The research on proposing various variants of Particle Swarm Optimization Technique is continued for last several decades. Efforts are being made to develop a most efficient algorithm. In this paper a newly developed Hybrid Particle Swarm Optimization Algorithm. (It will be known as PARIPSO) has been proposed. This algorithm has been constructed by taking contribution of gbest as 65% and contribution of pbest as 35% which is novel philosophy to update velocity equation. The proposed algorithm has been tested on several benchmark problems. The results thus obtained have been compared with those obtained using Standard Particle Swarm Optimization (SPSO) and Mean Particle Swarm Optimization (MPSO). On the basis of results obtained it is concluded that the proposed algorithm performs better than SPSO and MPSO in most of the cases in the terms of efficiency, time computation, reliability, accuracy and stability.

1. Introduction

Particle Swarm Optimization (PSO) was developed by Kennedy and Eberhart (1995) and Kennedy et al. (2001), based on the swarm behavior such as fish and bird schooling in nature. The Particle Swarm Optimization algorithm is comprised of a collection of particles that move around the search space influenced by their own best past location and the best past location of the whole swarm or a close neighbor. In each iteration a particle's velocity is updated using (Eq. 1):

$$v_i(k+1) = v_i(k) + (c_1 \times rand() \times (p_i^{best} - p_i(k))) + (c_2 \times rand(\ \ ) \times (p_{gbest} - p_i(k))) \quad (1)$$

where $v_i(k+1)$is the new velocity for the i^{th} particle, c_1 and c_2 are the weighting coefficients for the personal best and global best positions respectively, $p_i(k)$ is the i^{th} particle's position at time k, p_i^{best} is the i^{th} particle's best known position, and p_{gbest} is the best position known to the swarm.

The rand () function generate a uniformly random numbers in [0, 1]. Variants on this update equation consider best positions within a particles local neighborhood at time t.

A particle's position is updated using (Eq. 2):

$$p_i(k+1) = p_i(k) + v_i(k+1) \quad (2)$$

Particle, $p_i(k)$ is the i^{th} particle's position at time k and $v_i(k)$is the old velocity for the i^{th} particle.

Shi and Eberhart (1998) have introduced the inertia weight in this theory. An Inertia weight is related with the speed of last iteration, and the velocity update equation for the change of the speed is the following (Eq. 3):

$$v_i(k+1) = w \times v_i(k) + (c_1 \times rand() \times (p_i^{best} - p_i(k))) + (c_2 \times rand() \times (p_{gbest} - p_i(k))) \quad (3)$$

Clerc (1999) introduced the concept of construction factor. The following is the formula for its position and speed changing (Eq. 4):

$$v_i(k+1) = x \times (w \times v_i(k) + (c_1 \times rand() \times (p_i^{best} - p_i(k))) + (c_2 \times rand() \times (p_{gbest} - p_i(k))) \quad (4)$$

$x = \frac{2}{|2-\emptyset-\sqrt{\emptyset^2-4\emptyset}|}$ is called the contraction factor, $\emptyset = c_1 + c_2 > 4$. Generally, ϕ is equal to 4.1, so $x = 0.729$. The experimental results of equation (4) as compared with the PSO algorithm with inertia weights, the convergence speed in the PSO algorithm with the convergence agent is much quicker. In fact, when the proper values of w, $c_1$and c_2 is decided, the

* Corresponding Author.
Email Address: sssinghsharan@gmail.com (S. Singh)
https://doi.org/10.21833/ijaas.2016.12.013
2313-626X/© 2016 The Authors. Published by IASE.

two calculation methods are identical. So, the PSO algorithm with convergence agent can be regarded as a special example of the particle swarm optimization algorithm with inertia weights. Meanwhile, the properly selected parameters in the algorithms can improve the function of the methods.

1.1. Literature review

Several interesting variation of PSO algorithm have recently been proposed by various researcher.

Multi-Swarm Cooperative Particle Swarm Optimizer is developed by Niu et al. (2007). MCPSO is based on a master–slave model, in which a population consists of one master swarm and several slave swarms. The slave swarms execute a single PSO or its variants independently to maintain the diversity of particles, while the master swarm evolves based on its own knowledge and also the knowledge of the slave swarms. According to the co-evolutionary relationship between master swarm and slave swarms, two versions of MCPSO are proposed, namely the competitive version of MCPSO (COM-MCPSO) and the collaborative version of MCPSO (COL-MCPSO), where the master swarm enhances its particles based on an antagonistic scenario or a synergistic scenario, respectively. The performance of the proposed algorithms has been compared with the standard PSO (SPSO) and its variants to demonstrate the superiority of MCPSO.

Quadratic Interpolation Particle Swarm Optimization is developed by Pant et al. (2007). The QIPSO algorithm makes use of a multiparent, quadratic crossover/reproduction operator defined in the BPSO algorithm. The author has compared the results with Basic Particle Swarm Optimization.

Mean particle swarm optimization for function optimization has been introduced by Deep and Bansal (2009). The method is based on a novel philosophy by modifying the velocity update equation. This is done by replacing two terms of original velocity update equation by two new terms based on the linear combination of pbest and gbest. Its performance is compared with the standard PSO (SPSO) by testing on benchmark problems. Based on the numerical and graphical analyses of results it is shown that the MeanPSO outperforms the SPSO, in terms of efficiency, reliability, accuracy and stability.

Competitive Learning Model in introduced by Murugesan and Palaniswami (2012). The hybridization of this Algorithm using Swarm Intelligent techniques further improves the efficiency of the Algorithm. Various works on the hybridization of Particle Swarm Optimization (PSO) with Simple Competitive Learning (SCL) have been proposed and are found to be efficient in Image Segmentation.

A Modified Hybrid Particle Swarm Optimization (MHPSO) algorithm has been developed by Said Labed et al. (2011). This approach is combined by some principles of Particle Swarm Optimization (PSO), the Crossover operation of the Genetic Algorithm and 2-opt improvement heuristic. The main feature of this approach is that it allows avoiding a major problem of met heuristics by the parameters setting.

A New Disc-Based Particle Swarm Optimization is developed by Yadav and Deep (2012). With the help of this approach authors have solved complex optimization problems. The reliability of the algorithms is validated statistically on several benchmark problems and also compared with the existing versions of PSO.

One half global best position particle swarm optimization has been introduced by Singh and Singh (2011). The performance of this algorithm has been tested through numerical and graphical results. The results obtained are compared with the standard PSO (SPSO) for scalable and non-scalable problems. The results indicate that new approach is better as comparison to SPSO in the terms of efficiency, reliability, accuracy and stability.

Personal best position particle swarm optimization has been introduced by Singh and Singh (2012). In the proposed approach a novel philosophy of modifying the velocity update equation of Standard Particle Swarm Optimization approach has been used. The modification has been done by vanishing the gbest term in the velocity update equation of SPSO and thus relying on pbest only. The performance of the proposed algorithm (Personal Best Position Particle Swarm Optimization, PBPPSO) has been tested on several benchmark problems. It is concluded that the PBPPSO performs better than SPSO in terms of accuracy and quality of solution.

A new version of particle swarm optimization algorithm has been developed by Singh et al. (2012). The algorithm has been developed by combining two different approaches of PSO i.e., Standard Particle Swarm Optimization and Mean Particle Swarm Optimization. Numerical experiments for scalable and non-scalable well known test problems have shown the superiority of newly proposed Hybrid Particle Swarm Optimization (HPSO) approach, compared to the classical SPSO algorithm in terms of convergence, speed and quality of obtained solutions.

A Modified PSO algorithm has also been developed by Ghatei et al. (2012). In this approach, the range for achieved answers is defined that is the same parameter used in the GDA called "water level". Amount of this range reduces or increases regarding to algorithm's property being used in terms of minimum or maximum during the time. This algorithm has been tested on some standard functions and its performance has been compared with standard PSO. Test results indicate that the proposed method significantly improves the ability of PSO of escaping from the local optimal raise and increases the accuracy and the convergence rate.

2. New proposed algorithm: Pari PSO

The Objective of developing a new algorithm was to reduce the number of clocks in finding the minimum functional value and hence making the method more economic. To achieve it lot of numerical experiments were performed. In this algorithm the velocity equation has been updated as (Eq. 5):

$$v_i(k+1) = v_i(k) + (c_1 \times rand() \times \left(0.35 \times p_i^{best} - p_i(k)\right)) + (c_2 \times rand() \times \left(0.65 \times p_{gbest} - p_i(k)\right)) \quad (5)$$

In the velocity update equation of this new PSO the first term represents the current velocity of the particle and can be thought of as a momentum term. The second term is proportional to the vector$(0.35 \times p_i^{best} - p_i(k))$, is responsible for the attractor of particle's current position and positive direction of its own best position (pbest). The third term is proportional to the vector $\left(0.65 \times p_{gbest} - p_i(k)\right)$), which is responsible for the attractor of particle's current position.

The (original) process for implementing the global version of Pari PSO is as follows:

ALGORITHM- Pari PSO
For each particle
Initialize particle
END
Do
For each particle
Calculate fitness value
If the fitness value is better than its peronal best
set current value as the new ***pbest***
End
Choose the particle with the best fitness value of all as ***gbest***
For each particle
Calculate particle velocity according Equation
$v_i(k+1) = w \times v_i(k) + (c_1 \times rand() \times \left(0.35 \times p_i^{best} - p_i(k)\right)) + (c_2 \times rand() \times \left(0.65 \times p_{gbest} - p_i(k)\right))$
Update particle position according equation
$p_i(k+1) = p_i(k) + v_i(k+1)$
End While maximum iterations or minimum error criteria is not attained
END ALGORITHM

2.1. Remark

The name of this algorithm has been coined by the first author in the lingering memories of his beloved daughter Late Ms. Pari.

2.2. SPSO parameters settings

The parameter in the SPSO given in the literature is:

1. The number of particles should be low, around 20-40
2. The speed a particle can move (maximum change in its position per iteration) should be bounded, such as to a percentage of the size of the domain.
3. A local bias (local neighborhood) factor can be introduced where neighbors are determined based on Euclidean distance between particle positions.
4. Particles may leave the boundary of the problem space and may be penalized, be reflected back into the domain or biased to return back toward a position in the problem domain. Alternatively, a wrapping strategy may be used at the edge of the domain creating a loop, torrid or related geometrical structures at the chosen dimensionality.
5. An inertia or momentum coefficient can be introduced to limit the change in velocity (Weights 0.4 to 0.9 and momentum coefficient 1.4 to 2.0).
6. The maximum number of function evaluations is fixed to be 30,000.
7. The dynamic range for each element of a particle is defined as (-100,100), that is, the particle cannot move out of this range in each dim and thus Xmax = 100.
8. Maximum Error = 0.1 to 0.9
9. In the proposed method we have to test the same parameters as in SPSO

2.3. The test problems

Every new techniques of PSO has to be tested on some benchmark problems. Keeping this in view the proposed algorithms has been tested on 28 benchmark problems (15 Scalable and 13 Non-Scalable Problems). All these problems vary in difficulty levels and problem size. The performance of SPSO, MPSO and Pari PSO is evaluated on these benchmarks problems. These problems have been divided in two kinds of problem sets.

Problem Set I: Scalable Problems: - Those problems in which the dimension of the problems can be increased / decreased at will. In general, the complexity of the problem increases as the problem size is increased.

Problem Set II: Non-Scalable Problems- In which the problem size is fixed, but the problems have many local as well as global optima.

2.4. Analysis

In SPSO, MPSO and newly proposed algorithm Pari PSO the balance between the local and global exploration abilities is mainly controlled by the inertia weight. The experimental results have been performed to illustrate this. By setting the maximum velocity to be two, it was found that SPSO, MPSO and Pari PSO with an inertia weight in the range [0.4, 0.9] on average has a better and bad performance; that is, it has a large chance to find the global optimum within a reasonable number of iterations. A time decreasing inertia weight is found to be better than a fixed inertia weight 0.8 and acceleration coefficients 1.6.

A number of criteria are used to evaluate the performance of SPSO, MPSO with Pari PSO. The percentage of success is used to evaluate the reliability. The average number of function evaluations of successful runs and the average computational time of the successful runs, are used to evaluate the cost.

For Problem Set-I. The quality of the solution obtained is measured by the minimum, mean and standard deviation of the objective function values out of thirty runs. This is shown in Table 2, 8 and 14. The corresponding information for Problem Set-II is shown in Tables 5, 11 and 17 respectively. Similarly we obtained the time decreasing performance of the SPSO, MPSO and Pari PSO in Table 3, 6, 9, 12, 15, 18 respectively.

We are testing the new approach Pari Particle Swarm Optimization Algorithm on the parameter setting: inertia weight 0.6 and 0.7, swarm size 30 dim, function evaluation 30,000, acceptable error 0.9 and acceleration coefficient 1.4 and 1.5. On this parameter setting the results obtained by the new approach has been listed in the Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11. These results have also been illustrated through Figs. 1 to 6. The results indicate that the new approach does not yield the global optimal point in all 100% cases.

Table 1: Parameter setting for Pari PSO

Inertia Weight	0.7
Confidence	1.5
Dim	30
Swarm Size	30
Maximum Evaluation	30,000
Acceptable Error	0.9

Finally, we are testing the new approach on the parameter setting: inertia weight 0.8, swarm size 30 dim, function evaluations 30,000, acceptable error 0.9 and acceleration coefficient 1.6. For this parameter setting results indicate that Pari PSO is most efficient for finding the global optimal point as comparison to SPSO and MPSO in the terms of cases in the terms of efficiency, time computation, reliability, accuracy and stability many several types of benchmarks problems.

Table 2: Comparison of minimum objective function value obtained in 50 runs by SPSO, MPSO and Pari PSO for 15 Scalable problems set-I

Problem	Minimum Function Value			Mean Function Value			Standard Deviation			Rate of Success		
No.	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO
1	0.333237	0.116912	0.104504	0.466472	0.366841	0.335568	0.034995	0.100515	0.096336	100.00%	100.00%	100.00%
2	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
3	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
4	67.46491 9	0.074771	0.026964	131.9798	0.335697	0.300736	32.34162 2	0.116899	0.126024	0.00%	100.00%	100.00%
5	0.388931	0.080987	0.090896	0.475017	0.331975	0.323439	0.027427	0.105874	0.115487	100.00%	100.00%	100.00%
6	0.000458	0.000201	0.000000	0.136791	0.173732	0.146715	0.132152	0.153407	0.150484	100.00%	100.00%	100.00%
7	0.000007	0.000274	0.000033	0.139596	0.133538	0.137262	0.131760	0.144473	0.146399	100.00%	100.00%	100.00%
8	0.000026	0.000000	0.000003	0.005763	0.004310	0.003206	0.010851	0.005661	0.004902	100.00%	100.00%	100.00%
9	0.000008	0.000001	0.000080	0.065178	0.106307	0.075560	0.097276	0.135533	0.103360	100.00%	100.00%	100.00%
10	0.002075	0.000599	0.009711	0.144592	0.181111	0.153810	0.115705	0.138862	0.115868	100.00%	100.00%	100.00%
11	0.346877	0.024249	0.000355	0.454562	0.254027	0.196083	0.038501	0.135333	0.131251	100.00%	100.00%	100.00%
12	0.008442	0.003052	0.006482	0.223617	0.200235	0.248183	0.150935	0.144104	0.139184	100.00%	100.00%	100.00%
13	0.003015	0.164147	0.363567	0.227848	8.539143	36.74694	0.165879	7.736604	47.32171	100.00%	6.00%	2.00%
14	0.001623	0.000557	0.000497	0.105389	0.115323	0.103464	0.095023	0.106661	0.088842	100.00%	100.00%	100.00%
15	0.000009	0.000000	0.000002	0.003656	0.003064	0.002487	0.005729	0.003789	0.004110	100.00%	100.00%	100.00%

3. Experimental results and discussion

The Performance of the proposed PSO model is tested on a number of analytical benchmark functions which have been extensively used to compare PSO-type meta-heuristic algorithms in the literature. This paper utilizes the benchmark function set, shown in Set-I and Set-II.

Table 3: Testing of newly proposed approach, Pari PSO in terms of Time

Sr. No	Number of Clocks		
1	SPSO	MPSO	Pari PSO
2	4258	546	530
3	4867	4867	4820
4	7846	8049	7924
5	7129	811	826
6	1903	405	390
7	296	296	296
8	265	280	265
9	280	280	296
10	297	296	296
11	234	249	280
12	2074	452	468
13	296	327	312
14	577	5319	5476
15	280	312	312
16	296	296	296

The new algorithm was tested on a set of 28 benchmark Problems (15 Scalable and 13 Non-Scalable). The scalable and non-scalable problems were chosen as the test problems. The Standard Particle Swarm Optimization implementation was written in C and compiled using the Borland C++ Version 4.5 compiler. For the purpose of comparison, all the simulation use the parameter setting of the SPSO implementation except the inertia weight w, acceleration coefficient, swarm size and maximum velocity allowed. The swarm size (number of particles) varies from 20 to 30, inertia weight from 0.4 to 0.9 and acceleration coefficient between 1.4 and 2.0. The dynamic range for each element of a particle has been defined as (-100, 100), i.e., the particle cannot move out of this range in each dim and thus Xmax = 100. The maximum number of iterations allowed is 30,000. If the SPSO, Mean PSO and PariPSO implementation cannot find an acceptable solution within 30,000 iterations, it is ruled that it fails to find the global optimum in this run.

A number of criteria have been used to evaluate the performance of SPSO, Mean PSO and PariPSO. The percentage of success is used to evaluate the reliability. The average number of function evaluations of successful runs and the average computational time of the successful runs, are used to evaluate the cost. For problem SET-I, the conclusion has been drawn on the basis of the minimum mean, success rate and standard deviation of the objective function values in fifty runs. The corresponding information for problem SET-II has been drawn on similar basis.

Table 4: Comparison of minimum objective function value obtained in 50 runs by SPSO, MPSO and Pari PSO for 13 Non-Scalable Problems Set-II

Problem No.	Minimum Function Value			Mean Function Value			Standard Deviation			Rate of Success		
	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO
1	0.500000	0.500000	0.500000	0.500000	0.500000	0.500001	0.000000	0.000000	0.000002	100.00%	0.00%	0.00%
2	0.004908	0.013329	0.000544	0.212664	0.167218	0.176938	0.150964	0.133254	0.138689	100.00%	100.00%	100.00%
3	0.001029	0.004946	0.009126	0.249016	0.240726	0.212053	0.159486	0.140210	0.152504	100.00%	100.00%	100.00%
4	0.000533	0.031645	0.001223	0.236961	0.251522	0.265394	0.137953	0.140993	0.145386	100.00%	100.00%	100.00%
5	0.002502	0.001426	0.000817	0.178632	0.208039	0.242072	0.143712	0.138999	0.149717	100.00%	100.00%	100.00%
6	73046.59	73046.59	73046.59	73046.59	73046.59	73046.59	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
7	13.4467	27.3725	27.9998	46.7363	28.0679	28.306709	33.1909	0.214735	0.150201	0.00%	0.00%	0.00%
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
9	1.380465	1.380465	1.380465	1.380465	1.380465	1.380465	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
10	0.067997	0.043758	0.088462	0.583033	0.459506	0.388234	0.215576	0.219519	0.193844	100.00%	100.00%	100.00%
11	0.003378	0.007090	0.002439	0.263803	0.210116	0.278202	0.207517	0.176338	0.214453	100.00%	100.00%	100.00%
12	0.005549	0.008986	0.018278	0.369263	0.316727	0.421738	0.270722	0.230383	0.277308	100.00%	100.00%	100.00%
13	0.002655	0.006073	0.000060	0.319323	0.263009	0.338979	0.229666	0.181617	0.176046	100.00%	100.00%	100.00%

The new approach has been tested on different types of parameters. When we are testing the new approach for swarm size 30 dim, function evaluation 30,000, inertia weight 0.6 and 0.7, acceptable error 0.9 and acceleration coefficient 1.4 and 1.5, new approach PariPSO, SPSO and MPSO are failed to find the global optimal result on several scalable and non-scalable problems.

For the parameter setting swarm size 30 dim, function evaluation 30,000, inertia weight 0.8, acceptable error 0.9 and acceleration coefficient 1.6 the proposed algorithm has been tested on the given benchmark problems. With the help of this parameter setting we have obtained the optimal solution in most of the cases. These results have been shown in the Tables 14, 15, 17 and 18.

Table 5: Testing of newly proposed approach, Pari PSO in terms of time

Sr. No	Number of Clocks		
	SPSO	MPSO	Pari PSO
1	795	5413	5413
2	343	358	296
3	324	264	308
4	333	362	363
5	341	359	286
6	29144	27610	27971
7	5415	5598	5558
8	4898	4976	4976
9	17284	16707	7175
10	468	312	343
11	265	296	296
12	296	280	296
13	280	296	234

Table 6: Comparison of minimum objective function value obtained in 50 runs by SPSO, MPSO and Pari PSO for 15 scalable problems set-I

Problem No.	Minimum Function Value			Mean Function Value			Standard Deviation			Rate of Success		
	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO
1	0.619676	0.234792	0.246675	0.922738	0.690200	0.694141	0.204204	0.149918	0.152567	70.00%	100.00%	100.00%
2	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
3	0.094588	0.025141	0.036357	0.148249	0.055662	0.062988	0.022320	0.011667	0.012798	100.00%	100.00%	100.00%
4	17.929427	1.000000	1.000000	26.948429	1.000000	1.000000	2.620415	0.000000	0.000000	0.00%	0.00%	0.00%
5	0.543236	0.166967	0.179300	0.803042	0.637185	0.562523	0.091277	0.189205	0.204035	100.00%	100.00%	100.00%
6	0.000001	0.000036	0.000006	0.279604	0.235347	0.281010	0.273092	0.244694	0.262177	100.00%	100.00%	100.00%
7	0.000040	0.000080	0.000094	0.227999	0.267945	0.250953	0.254545	0.273861	0.268882	100.00%	100.00%	100.00%
8	0.000000	0.000001	0.000003	0.005115	0.002858	0.004850	0.007175	0.005120	0.008650	100.00%	100.00%	100.00%
9	0.000001	0.000039	0.000085	0.148325	0.082883	0.119724	0.208067	0.148494	0.174187	100.00%	100.00%	100.00%
10	0.000732	0.004534	0.006678	0.212338	0.159204	0.208941	0.181537	0.135146	0.174190	100.00%	100.00%	100.00%
11	0.428080	0.059734	0.004511	0.788042	0.389512	0.446050	0.108482	0.236531	0.275453	100.00%	100.00%	100.00%
12	0.008529	0.000828	0.013052	0.431102	0.396643	0.398125	0.244669	0.264548	0.251589	100.00%	100.00%	100.00%
13	0.015651	0.030990	0.074653	0.438592	5.363785	104.003060	0.260853	4.798968	81.704200	100.00%	16.00%	4.00%
14	0.00162	0.001080	0.000826	0.128907	0.129157	0.096919	0.121869	0.124008	0.088872	100.00%	100.00%	100.00%
15	0.000000	0.000001	0.000002	0.003158	0.002099	0.003639	0.004651	0.003652	0.005484	100.00%	100.00%	100.00%

The results of Table 14 indicate that the new approach has solved thirteen scalable problems with 100% success and it also outperforms SPSO and MPSO in all these problems. All approaches has been failed to find global optimal solution for one scalable problem. In the fifteenth problem the performance of PariPSO is better to SPSO not to MPSO.

Moreover the results of Table 15 shows that PariPSO is finding the optimal point in less number of clocks on thirteen scalable problems as comparison to SPSO and MPSO. Thus it takes less CPU time and hence becomes most economic amongst these methods. Only in one problem it takes less time to SPSO but more time in comparison to MPSO.

The results of Table 17 show that the new approach has solved eight non scalable problems with 100% success and outperformed SPSO and MPSO. All these approaches failed to solve two non-scalable problems successfully. In the remaining three problems the proposed approach is better to SPSO but not to MPSO.

Lastly, the results of Table 18 shows that three non-scalable problems has been solved successfully using PariPSO in less number of clocks compared to SPSO and MPSO. For five non scalable problems the new approach is finding the global optimal point in less number of clocks as comparison to SPSO but not to MPSO. In two problems PariPSO proved to be better than MPSO but worse than SPSO. In the remaining two all the algorithms failed to find the optimal solution.

Table 7: Testing of newly proposed approach Pari PSO in terms of time

Sr. No	Number of Clocks		
	SPSO	MPSO	Pari PSO
1	3340	458	427
2	4984	5005	5074
3	202	250	229
4	7306	7470	7422
5	769	289	320
6	220	219	228
7	272	285	297
8	284	291	247
9	325	257	249
10	270	303	308
11	964	356	357
12	282	277	287
13	474	5184	5505
14	317	277	279
15	310	297	299

Table 8: Comparison of minimum objective function value obtained in 50 runs by SPSO, MPSO and Pari PSO for 13 Non-scalable problems Set-II

Problem No.	Minimum Function Value			Mean Function Value			Standard Deviation			Rate of Success		
	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO
1	0.500003	0.500003	0.500003	0.526973	0.582082	0.542196	0.047372	0.097857	0.054360	100.00%	100.00%	100.00%
2	0.002786	0.002983	0.001972	0.308700	0.285012	0.306678	0.240717	0.238237	0.249587	100.00%	100.00%	100.00%
3	0.007013	0.001252	0.009322	0.329615	0.306953	0.355605	0.250265	0.240706	0.269558	100.00%	100.00%	100.00%
4	0.021654	0.002922	0.002611	0.470666	0.352727	0.379539	0.256132	0.237292	0.261926	100.00%	100.00%	100.00%
5	0.003533	0.001515	0.001515	0.349886	0.411309	0.354547	0.238325	0.267259	0.220808	100.00%	100.00%	100.00%
6	73046.59	73046.59	73046.59	73046.59	73046.59	73046.59	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
7	3.575332	28.10045	28.5865	26.79365 3	28.57612	28.75951 3	13.62731 3	0.326619	0.069283	0.00%	0.00%	0.00%
8	0.010057	0.032551	0.007865	0.491922	0.525217	0.545747	0.281136	0.226387	0.240063	100.00%	100.00%	100.00%
9	0.480465	0.480466	0.480479	0.507009	0.492759	0.494226	0.030349	0.016595	0.020826	100.00%	100.00%	100.00%
10	0.055860	0.022383	0.021841	0.510907	0.395098	0.405757	0.260473	0.218544	0.238010	100.00%	100.00%	100.00%
11	0.007338	0.005380	0.007508	0.203349	0.151939	0.177446	0.170201	0.121302	0.168608	100.00%	100.00%	100.00%
12	0.015703	0.001915	0.000381	0.327649	0.304038	0.311233	0.237534	0.228422	0.229011	100.00%	100.00%	100.00%
13	0.006582	0.001053	0.000042	0.258751	0.214439	0.278492	0.258751	0.152282	0.177249	100.00%	100.00%	100.00%

Table 9: Testing of Newly Proposed Approach Pari PSO in terms of Time

Sr. No	Number of Clocks		
	SPSO	MPSO	Pari PSO
1	265	218	202
2	218	343	203
3	234	202	234
4	202	202	234
5	249	234	202
6	28719	27440	27643
7	5163	5475	5444
8	265	234	234
9	312	265	312
10	358	343	343
11	280	296	343
12	234	296	265
13	265	280	265

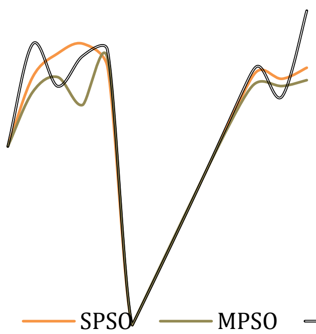

Fig. 1: Comparison of results obtained SPSO , MPSO and Pari PSO for the set of 15 scalable problems SET-I- Table 2

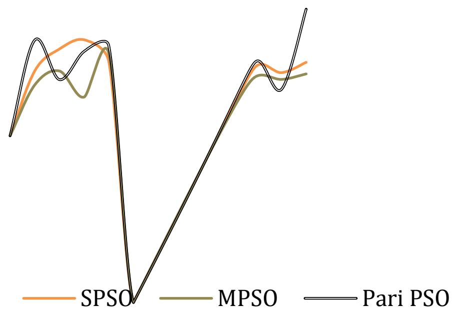

Fig. 2: Comparing of results obtained SPSO, MPSO and Pari PSO for the set of 13 Non-scalable problems SET-I- Table 4

On the basis of results obtained authors conclude that PariPSO is most economic, efficient and faster in comparison to SPSO and MPSO for a defined set of parameters.

4. Conclusion

A new version of the Particle Swarm Optimization (PSO) has been introduced in this paper. The method will be known as Pari Particle Swarm Optimization (Pari PSO).

Table 10: Comparison of minimum objective function value obtained in 50 runs by SPSO, MPSO and Pari PSO for 15 Non-scalable problems Set-I

Problem	Minimum Function Value			Mean Function Value			Standard Deviation			Rate of Success		
No.	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO
1	6.216648	0.354164	0.323747	8.453022	0.717572	0.698956	1.261263	0.135261	0.151211	0.00%	100.00%	100.00%
2	0.900000	0.900000	0.900000	0.900000	0.900000	0.900000	0.000000	0.000000	0.000000	100.00%	100.00%	100.00%
3	1.000305	1.000000	1.000000	1.002817	1.000000	1.000000	0.004139	0.000000	0.000000	0.00%	0.00%	0.00%
4	5654.218	0.059967	0.037287	18951.95	0.546463	0.533749	9417.662	0.214455	0.215857	0.00%	100.00%	100.00%
5	0.739977	0.223565	0.133032	1.195796	0.669006	0.632472	0.613865	0.157349	0.194815	54.00%	100.00%	100.00%
6	0.000089	0.000672	0.000007	0.301340	0.243549	0.281184	0.262897	0.249320	0.263720	100.00%	100.00%	100.00%
7	0.000029	0.000046	0.000180	0.247606	0.268767	0.258546	0.252291	0.252866	0.257292	100.00%	100.00%	100.00%
8	0.000004	0.000007	0.000001	0.005216	0.009008	0.007259	0.007772	0.021359	0.018882	100.00%	100.00%	100.00%
9	0.000111	0.000006	0.000064	0.151264	0.118851	0.103578	0.225395	0.190112	0.141065	100.00%	100.00%	100.00%
10	0.007634	0.001831	0.001831	0.216753	0.184000	0.192249	0.180577	0.169350	0.175228	100.00%	100.00%	100.00%
11	15.644990	0.031034	0.002714	390.1446	0.451726	0.384586	1391.722	0.251974	0.261574	0.00%	100.00%	100.00%
12	0.038170	0.024035	0.001911	0.434670	0.410699	0.360954	0.253007	0.248216	0.241765	100.00%	100.00%	100.00%
13	0.031976	0.217234	0.020919	0.446246	11.73805	11.83678	0.252932	11.97660	12.13561	100.00%	4.00%	8.00%
14	0.001623	0.001623	0.001623	0.114224	0.089594	0.097428	0.099862	0.081496	0.093799	100.00%	100.00%	100.00%
15	0.000003	0.000005	0.000001	0.004755	0.007066	0.002943	0.007811	0.018895	0.003906	100.00%	100.00%	100.00%

Table 11: Testing of newly proposed approach, Pari PSO in terms of time

Sr. No	Number of Clocks		
	SPSO	MPSO	Pari PSO
1	7472	998	889
2	1076	982	951
3	7893	8018	8002
4	7238	2480	1778
5	6723	546	561
6	312	296	296
7	202	265	202
8	218	202	187
9	280	249	202
10	202	405	202
11	4758	592	546
12	202	390	202
13	764	5288	5194
14	203	187	202
15	421	202	187

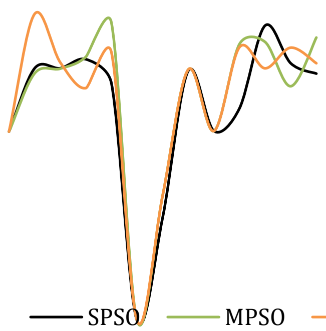

Fig. 4: Comparison of results obtained using SPSO, MPSO and Pari PSO for the set of 13 Non-scalable problems SET-I- Table 8

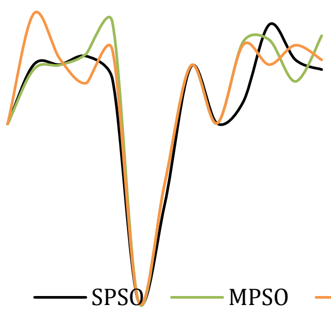

Fig. 3: Comparison of results obtained using SPSO, MPSO and Pari PSO for the set of 15 scalable problems SET-I-Table 6

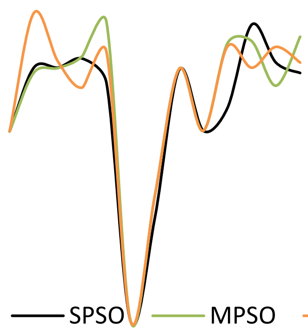

Fig. 5: Comparison of results obtained using SPSO, MPSO and Pari PSO for the set of 15 Scalable Problems SET-I-Table 10

Table 12: Comparison of Minimum Objective function value obtained in 50 runs by SPSO, MPSO and Pari PSO for 13 Non-Scalable Problems Set-II

Problem No.	Minimum Function Value			Mean Function Value			Standard Deviation			Rate of Success		
	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO
1	0.500169	0.500044	0.500001	0.559173	0.567709	0.542520	0.097106	0.085016	0.060343	100.00%	100.00%	100.00%
2	0.009310	0.012805	0.000305	0.400386	0.385434	0.320214	0.272983	0.260114	0.279873	100.00%	100.00%	100.00%
3	0.009239	0.009525	0.006395	0.367867	0.323189	0.343069	0.244524	0.244053	0.229518	100.00%	100.00%	100.00%
4	0.005260	0.004538	0.032348	0.470888	0.403137	0.447511	0.253210	0.267541	0.271799	100.00%	100.00%	100.00%
5	0.023872	0.000531	0.003430	0.345565	0.356886	0.373028	0.225290	0.228851	0.282113	100.00%	100.00%	100.00%
6	73046.59	73046.596	73046.596	73046.59	73046.596	73046.596	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
7	111.2349	27.466171	27.866916	601.5607	28.043699	28.267751	468.6419	0.191601	0.164995	0.00%	0.00%	0.00%
8	0.010858	0.010057	0.010057	0.481820	0.473086	0.514414	0.268329	0.256105	0.241689	100.00%	100.00%	100.00%
9	0.480474	0.480478	0.480566	0.508201	0.497816	0.501146	0.029480	0.023301	0.024134	100.00%	100.00%	100.00%
10	0.114681	0.002013	0.002462	0.618387	0.449032	0.493212	0.203637	0.268925	0.268903	100.00%	100.00%	100.00%
11	0.000632	0.001778	0.009317	0.281051	0.209253	0.182759	0.244625	0.220955	0.182759	100.00%	100.00%	100.00%
12	0.006776	0.028729	0.002534	0.371321	0.302112	0.374412	0.238325	0.211787	0.238193	100.00%	100.00%	100.00%
13	0.012970	0.001342	0.006773	0.277142	0.286922	0.328632	0.172771	0.206169	0.183676	100.00%	100.00%	100.00%

Table 13: Testing of newly proposed approach, Pari PSO in terms of time

Sr. No	Number of Clocks		
	SPSO	MPSO	Pari PSO
1	287	312	249
2	327	250	296
3	325	292	347
4	321	307	332
5	273	223	313
6	28969	28563	28657
7	5350	5506	5475
8	312	327	296
9	297	280	296
10	514	343	374
11	218	296	265
12	265	280	296
13	280	312	327
14	287	312	249
15	327	250	296

Table 14: Comparison of Minimum Objective function value obtained in 50 runs by SPSO, MPSO and Pari PSO for 15 Non-Scalable Problems Set-I

Problem No.	Minimum Function Value			Mean Function Value			Standard Deviation			Rate of Success		
	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO
1	6.216648	0.354164	0.323747	8.453022	0.717572	0.698956	1.261263	0.135261	0.151211	0.00%	100.00%	100.00%
2	0.900000	0.900000	0.900000	0.900000	0.900000	0.900000	0.000000	0.000000	0.000000	100.00%	100.00%	100.00%
3	1.000305	1.000000	1.000000	1.002817	1.000000	1.000000	0.004139	0.000000	0.000000	0.00%	0.00%	0.00%
4	5654.218	0.059967	0.037287	18951.95	0.546463	0.533749	9417.662	0.214455	0.215857	0.00%	100.00%	100.00%
5	0.739977	0.223565	0.133032	1.195796	0.669006	0.632472	0.613865	0.157349	0.194815	54.00%	100.00%	100.00%
6	0.000089	0.000672	0.000007	0.301340	0.243549	0.281184	0.262897	0.249320	0.263720	100.00%	100.00%	100.00%
7	0.000029	0.000046	0.000180	0.247606	0.268767	0.258546	0.252291	0.252866	0.257292	100.00%	100.00%	100.00%
8	0.000004	0.000007	0.000001	0.005216	0.009008	0.007259	0.007772	0.021359	0.018882	100.00%	100.00%	100.00%
9	0.000111	0.000006	0.000064	0.151264	0.118851	0.103578	0.225395	0.190112	0.141065	100.00%	100.00%	100.00%
10	0.007634	0.001831	0.001831	0.216753	0.184000	0.192249	0.180577	0.169350	0.175228	100.00%	100.00%	100.00%
11	15.644990	0.031034	0.002714	390.1446	0.451726	0.384586	1391.722	0.251974	0.261574	0.00%	100.00%	100.00%
12	0.038170	0.024035	0.001911	0.434670	0.410699	0.360954	0.253007	0.248216	0.241765	100.00%	100.00%	100.00%
13	0.031976	0.217234	0.020919	0.446246	11.73805	11.83678	0.252932	11.97660	12.13561	100.00%	4.00%	8.00%
14	0.001623	0.001623	0.001623	0.114224	0.089594	0.097428	0.099862	0.081496	0.093799	100.00%	100.00%	100.00%
15	0.000003	0.000005	0.000001	0.004755	0.007066	0.002943	0.007811	0.018895	0.003906	100.00%	100.00%	100.00%

Table 15: Testing of Newly Proposed Approach, Pari PSO in terms of Time

Sr. No	Number of Clocks		
	SPSO	MPSO	Pari PSO
1	7472	998	889
2	1076	982	951
3	7893	8018	8002
4	7238	2480	1778
5	6723	546	561
6	312	296	296
7	202	265	202
8	218	202	187
9	280	249	202
10	202	405	202
11	4758	592	546
12	202	390	202
13	764	5288	5194
14	203	187	202
15	421	202	187

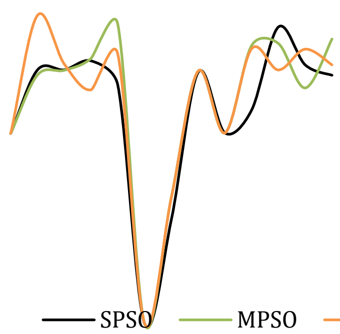

Fig. 6: Comparison of results obtained using SPSO, MPSO and Pari PSO for the set of 13 Non-Scalable Problems SET-I- Table 12

Table 16: Parameter Setting for Pari PSO

Inertia Weight	**0.8**
Confidence	1.6
Dim	30
Swarm Size	30
Maximum Evaluation	30,000
Acceptable Error	0.9

The performance of this approach has been compared with SPSO and MPSO in the terms of efficiency, time computation, reliability, accuracy and stability and number of clocks.

The test results shows that the proposed approach significantly improves the ability of PSO to find global optimal solution and also increases the accuracy or convergence rate.

Table 17: Comparison of Minimum Objective function value obtained in 50 runs by SPSO, MPSO and Pari PSO for 13 Non-Scalable Problems Set-II

Problem No.	Minimum Function Value			Mean Function Value			Standard Deviation			Rate of Success		
	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO	SPSO	MPSO	Pari PSO
1	0.500169	0.500044	0.500001	0.559173	0.567709	0.542520	0.097106	0.085016	0.060343	100.00%	100.00%	100.00%
2	0.009310	0.012805	0.000305	0.400386	0.385434	0.320214	0.272983	0.260114	0.279873	100.00%	100.00%	100.00%
3	0.009239	0.009525	0.006395	0.367867	0.323189	0.343069	0.244524	0.244053	0.229518	100.00%	100.00%	100.00%
4	0.005260	0.004538	0.032348	0.470888	0.403137	0.447511	0.253210	0.267541	0.271799	100.00%	100.00%	100.00%
5	0.023872	0.000531	0.003430	0.345565	0.356886	0.373028	0.225290	0.228851	0.282113	100.00%	100.00%	100.00%
6	73046.59	73046.596	73046.596	73046.59	73046.596	73046.596	0.000000	0.000000	0.000000	0.00%	0.00%	0.00%
7	111.2349	27.466171	27.866916	601.5607	28.043699	28.267751	468.6419	0.191601	0.164995	0.00%	0.00%	0.00%
8	0.010858	0.010057	0.010057	0.481820	0.473086	0.514414	0.268329	0.256105	0.241689	100.00%	100.00%	100.00%
9	0.480474	0.480478	0.480566	0.508201	0.497816	0.501146	0.029480	0.023301	0.024134	100.00%	100.00%	100.00%
10	0.114681	0.002013	0.002462	0.618387	0.449032	0.493212	0.203637	0.268925	0.268903	100.00%	100.00%	100.00%
11	0.000632	0.001778	0.009317	0.281051	0.209253	0.182759	0.244625	0.220955	0.182759	100.00%	100.00%	100.00%
12	0.006776	0.028729	0.002534	0.371321	0.302112	0.374412	0.238325	0.211787	0.238193	100.00%	100.00%	100.00%
13	0.012970	0.001342	0.006773	0.277142	0.286922	0.328632	0.172771	0.206169	0.183676	100.00%	100.00%	100.00%

Table 18: Testing of Newly Proposed Approach, Pari PSO in terms of Time

Sr. No	Number of Clocks		
	SPSO	MPSO	Pari PSO
1	287	312	249
2	327	250	296
3	325	292	347
4	321	307	332
5	273	223	313
6	28969	28563	28657
7	5350	5506	5475
8	312	327	296
9	297	280	296
10	514	343	374
11	218	296	265
12	265	280	296
13	280	312	327

References

Clerc M (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In the Proceedings of the 1999 Congress on Evolutionary Computation (CEC '99), IEEE, 3. https://doi.org/10.1109/CEC.1999.785513

Deep K and Bansal JC (2009). Mean particle swarm optimisation for function optimisation. International Journal of Computational Intelligence Studies, 1(1): 72-92.

Ghatei S, Khajei RP, Maman MS and Meybodi MR (2012). A modified PSO using great deluge algorithm for optimization. Journal of Basic and Applied Scientific Research, 2(2): 1362-1367.

Kennedy J and Eberhart RC (1995). Particle Swarm optimization. In the Proceedings of IEEE International Conference on Neural Networks. https://doi.org/10.1109/ICNN.1995.488968

Kennedy J, Kennedy JF, Eberhart RC, and Shi Y (2001). Swarm intelligence. 1st Edition, Morgan Kaufmann Publishers, USA.

Labed S, Gherboudj A, and Chikhi S (2011). A modified hybrid particle swarm optimization algorithm for multidimensional knapsack problem. Journal of Theoretical and Applied Information Technology, 39(2): 11-16.

Murugesan KM and Palaniswami S (2012). Hybrid exponential particle swarm optimization k-means algorithm for efficient image segmentation. Journal of Computer Science, 8(11): 1874-1879.

Niu B, Zhu Y, He X, and Wu H (2007). MCPSO: A multi-swarm cooperative particle swarm optimizer. Applied Mathematics and Computation, 185(2): 1050-1062.

Pant M, Radha T, and Singh VP (2007). A new particle swarm optimization with quadratic interpolation. In the International Conference on Computational Intelligence and Multimedia Applications, IEEE, 1: 55-60. https://doi.org/10.1109/ICCIMA.2007.95

Shi Y and Eberhart R (1998). A modified particle swarm optimizer. In the IEEE International Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence: 69-73. https://doi.org/10.1109/ICEC.1998.699146

Singh N and Singh SB (2011). One Half Global Best Position Particle Swarm Optimization.

International Journal of Scientific and Engineering Research, 2(8): 1-10.

Singh N and Singh SB (2012). Personal Best Position Particle Swarm Optimization. Journal of Applied Computer Science and Mathematics, 12(6) :69-76.

Singh N, Singh S, and Singh SB (2012). HPSO: A New Version of Particle Swarm Optimization. Journal of Artificial Intelligence, 3(3): 123-134,

Yadav A and Deep K (2012). A New Disc Based Particle Swarm Optimization. In the Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS '11), Springer India: 23-30.