A New Approach for Momentum Particle Swarm Optimization

Rohan Mohapatra, Rohan R. Talesara, Saloni Govil, Snehanshu Saha, Soma S. Dhavala, and TSB Sudarshan

1 Introduction

Real-world optimization problems are mostly, multidimensional, and multi-objective. Compared to unconstrained optimization, constrained optimization problems pose many challenges. Constraints are difficult to model in the problem and are generally represented by equality or non-equality. Also, the constraints modify the search space. Consequently, any solution to optimize the problem must do so in the bounds of the constraints. A traditional way to approach constrained optimization problems is to use mathematical programming. However, it is restrictive in the sense that it is heavily dependent on the problem being optimized and how it is modelled. If the problem is not modelled well, mathematical programming may

R. Mohapatra (✉) · R. R. Talesara · S. Govil
PES University, Bangalore, India
e-mail: rohanmohapatra@pesu.pes.edu

R. R. Talesara
e-mail: rohantalesara@pesu.pes.edu

S. Govil
e-mail: salonigovil@pesu.pes.edu

S. Saha
Department of Computer Science and Information Systems and APPCAIR, BITS Pilani, K K Birla, Goa, Sancoale, India
e-mail: snehanshu.saha@ieee.org

S. S. Dhavala
Center for Astroinformatics, Modeling and Simulation (CAMS), Department of Computer Science, PES University, Bengalore, India
e-mail: soma.dhavala@gmail.com

T. Sudarshan
Department of Computer Science, PES University, Bengalore, India
e-mail: sudarshan@pes.edu

S. Patnaik et al. (eds.), *Advances in Machine Learning and Computational Intelligence*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-15-5243-4_5

not yield good results. Another widely used approach is to calculate the gradient and use it to optimize the problem, as in the gradient descent algorithm. However, this approach requires one to find the derivative of the function, which may be difficult in real-world problems which are generally constrained, multidimensional, non-continuous, and non-convex. As a result of the shortcomings of traditional optimization algorithms, there has been significant interest in using meta-heuristic algorithms for optimization. In particular, there has been significant interest in nature-inspired and evolutionary algorithms such as genetic algorithm [1], ant colony algorithm, simulated annealing, and particle swarm optimization. It is important to note that meta-heuristic algorithms generally give approximate solutions and not exact solutions. The particle swarm optimization algorithm was proposed in 1995 as a swarm intelligence-based evolutionary algorithm for optimization problems. It is inspired by the flocking behavior of birds and is characterized by two equations: velocity update and position update. The velocity update is controlled by both a global and local component. PSO, however, has its own shortcomings. It may get stuck in the local-extremum, and is prone to pre-mature convergence. Various modified versions of PSO [2] have been proposed to overcome its shortcomings and make it more robust and fast. Garg [3] presented a hybrid of genetic algorithm and PSO, where the PSO is used for improving the velocity vector and the GA has been used for modifying the decision vectors using genetic operators. Tripathi et al. [4] proposed the time-variant PSO where the coefficients of the PSO velocity update equation (inertia weight and acceleration coefficients) are allowed to change every iteration. This has been said to help the algorithm explore the search space more efficiently. Sun et al. [5] proposed a quantum-behaved PSO, wherein the particles follow quantum mechanics as opposed to Newtonian mechanics. In quantum mechanics, it is not possible to determine both the position and velocity of a particle simultaneously, which leads to radically different PSO equations. Nebro et al. [6] present speed constrained multi-objective PSO (SMPSO) which provides a mechanism to control the velocity of the particles when it becomes too high. It also includes a turbulence factor and an external archive to store non-dominated solutions found during the search. Xiang et al. [7] presented the momentum PSO algorithm which uses a momentum term in the velocity update equation. However, the momentum term is defined such that the effect of the previous velocity terms is diminishingly small, and hence, the momentum term does not provide great acceleration compared to the original PSO algorithm. To this end, in this article, we propose a new momentum PSO, wherein the momentum term is influenced by the previous velocities to a greater extent. The proposed algorithm has been tested on 20 benchmark test optimization functions, and results show that the proposed algorithm performs better than both weighted PSO and momentum PSO in almost all cases. Performance here is measured in terms of how fast the optimal value is reached. Further, the proposed PSO is used to optimize two habitability scores, namely, Cobb–Douglas habitability score [8, 9] and constant elasticity earth similarity approach [10] by maximizing their respective production functions. These scores are used to assess extra-solar planets and assign them scores based on their habitability and earth similarity. CDHS considers the planet's radius, mass, escape velocity, and surface temperature, while CEESA includes a fifth parameter, the orbital eccentricity of the planet. Ultimately, we also show that the proposed PSO is equivalent to gradient descent with momentum. The contents of the paper are organized as follows: Sect. 2 explains the weighted PSO and momentum PSO, Sect. 3 explains the proposed PSO algorithm and its equivalence to gradient descent with momentum. Section 4 explains the two habitability scores, namely CDHS and CEESA. Section 5 presents the results of running the proposed algorithm against 20 benchmark test optimization functions, CDHS and CEESA, with graphs and tables comparing it to both weighted PSO and momentum PSO.

2 Particle Swarm Optimization with Its Variants

2.1 Particle Swarm Optimization Algorithm with Inertial Weight

The particle swarm optimization algorithm [11] is an optimization algorithm inspired by the flocking behaviour of birds. It is characterized by a population of particles in space, which aim to converge to an optimal point. The movement of the particles in space is characterized by two equations, namely velocity and position update equations, which are as follows:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_i^{best} - x_i^t) + c_2 r_2 (g^{best} - x_i^t) \tag{1}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2}$$

where ω, c_1, $c_2 \geq 0$. Here, x_i^t refers to the position of particle i in the search space at time t, v_i^t refers to the velocity of particle i at time t, p_i^{best} is the personal best position of particle i, and g_i^{best} is the best position amongst all the particles of the population. The movement of the particles is decided by updating the current position of the particles with the velocity as shown in (3).

2.2 Particle Swarm Optimization Algorithm with Momentum

The back-propagation algorithm is one of the most frequently used algorithms for training multilayer feed-forward neural networks. It uses gradient descent method to minimize the error between actual output and expected output, but it tends to stagnate or oscillate in the superficial local minima and fail to converge to global minimum. The momentum term was introduced to deal with this issue. It helps by acting as a low pass filter to smoothen.

Inspired by this, a momentum is introduced in the velocity updating equation of *PSO*. Thus, the new equation along with the momentum term was introduced by the following equation,

$$v_i^{t+1} = (1 - \lambda)(v_i^t + c_1 r_1 (p_i^{best} - x_i^t) + c_2 r_2 (g^{best} - x_i^t)) + \lambda v_i^{t-1}$$

where c_1, c_2, x_i^t, v_i^t, p_i^{best}, g_i^{best} mean the same as described in the previous section. The momentum factor is indicated by λ.

3 New Approach to Momentum Particle Swarm Optimization

In this section, we propose a rather new approach to momentum particle swarm optimization. The problem with the currently available *m-PSO* that weighted average which is computed takes care of both exploration and exploitation simultaneously. Since PSO tries to search the space by exploring, it makes more sense to give more weight to the exploration part of the equation. Another problem with the above term is that it takes longer iteration to reduce error and reach the minimum.

To counter the above-said problems, we mathematically formulate a new particle swarm optimization with momentum as follows:

$$v_i^{t+1} = M_i^{t+1} + c_1 r_1 (p_i^{best} - x_i^t) + c_2 r_2 (g^{best} - x_i^t) \tag{3}$$

where

$$M_i^{t+1} = \beta M_i^t + (1 - \beta) v_i^t \tag{4}$$

Here β is the momentum factor, and M_i^{t+1} indicates the effect of the momentum. The above equation can be rewritten as the following by combining (4) and (5),

$$v_i^{t+1} = \beta M_i^t + (1 - \beta) v_i^t + c_1 r_1 (p_i^{best} - x_i^t) + c_2 r_2 (g^{best} - x_i^t) \tag{5}$$

We understand that PSO is composed of two phases: the exploration phase and the exploitation phase.

$$\underbrace{v_i^t}_{\text{Exploration}} + \underbrace{c_1 r_1 (p_i^{best} - x_i^t) + c_2 r_2 (g^{best} - x_i^t)}_{\text{Exploitation}}$$

With above proposed approach, the exploration phase is determined by the **exponential weighted average of the previous velocities seen so far** only. The negligible weights applied in the momentum PSO do not help much in providing us that acceleration required.

3.1 Exponentially Weighted Average

In the previous section, we defined the momentum (5) and we also mentioned that it was an exponentially weighted average of the previous velocities seen so far. We prove it by expanding M_i^t(5); we get the following derivation.

$$M_i^t = \beta M_i^{t-1} + (1-\beta)v_i^{t-1} \tag{6}$$

$$M_i^t = \beta[\beta M_i^{t-2} + (1-\beta)v_i^{t-2}] + (1-\beta)v_i^{t-1} \tag{7}$$

$$M_i^t = \beta^2 M_i^{t-2} + \beta(1-\beta)v_i^{t-2} + (1-\beta)v_i^{t-1} \tag{8}$$

$$M_i^t = \beta^2[\beta M_i^{t-3} + (1-\beta)v_i^{t-3}] + \beta(1-\beta)v_i^{t-2} + (1-\beta)v_i^{t-1} \tag{9}$$

$$M_i^t = \beta^3 M_i^{t-3} + \beta^2(1-\beta)v_i^{t-3} + \beta(1-\beta)v_i^{t-2} + (1-\beta)v_i^{t-1} \tag{10}$$

Generalizing M_i^t, it can be written as the follows,

$$\begin{aligned} M_i^t =& \beta^n M_i^{t-n} + \beta^{(n-1)}(1-\beta)v_i^{t-n} + \beta^{(n-2}(1-\beta)v_i^{t-(n-1)} + \ldots \\ &+ \beta(1-\beta)v_i^{t-2} + (1-\beta)v_i^{t-1} \end{aligned}$$

3.2 Equivalence to Stochastic Gradient Descent Rule

For functions of multiple variables, the Taylor expansion with remainder is f(x) = f(a) + f'(a)(x - a) + $E_n(x)$.

The gradient descent weight update with momentum is given by,

$$w^{(t)} = w^{(t-1)} + \eta V_{dw}^t \tag{11}$$

$$V_{dw}^t = \beta V_{dw}^{t-1} + (1-\beta)\frac{\partial f}{\partial w} \tag{12}$$

Combining the equations and dividing (13) by $(1-\beta)$, we get

$$w^{(t)} = w^{(t-1)} + \alpha V_{dw}^{t-1} + \eta \frac{\partial f}{\partial w} \tag{13}$$

Here, η is the learning rate and V_{dw}^{t-1} is the momentum applied to the weight update.

Let us apply Taylor expansion to the above to get,

$$w^{(t)} = w^{(t-1)} + \eta f(w\prime) + \eta \sum_{j=1}^{n} \frac{\partial f}{\partial w_j}(w\prime)(w_j^{t-1} - w_j') + E_n(x) + \alpha V_{dw}^{t-1} \tag{14}$$

at some optimum value $w\prime$. The equation of the exponentially weighted momentum particle swarm optimization is,

$$v_i^t = M_i^t + c_1 r_1 (p_i^{best} - x_i^{t-1}) + c_2 r_2 (g^{best} - x_i^{t-1}) \tag{15}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{16}$$

Combining (16) and (17) and expanding, we get

$$x_i^t = x_i^{t-1} + \beta M_i^{t-1} + (1-\beta) v_i^{t-1} + c_1 r_1 (p_i^{best} - x_i^{t-1}) + c_2 r_2 (g^{best} - x_i^{t-1}) \tag{17}$$

The term $c_2 r_2 (g^{best} - x_i^{t-1})$ in PSO is the social factor. We equate this social factor with the error term in the Taylor expansion. We get

$$k_1 (x-a)^{n+1} \leq E_n(x) \leq k_2 (x-a)^{n+1} \tag{18}$$

$$E_n(x) = k(x-a)^{n+1} \tag{19}$$

Expanding the above at optimum value w', and equating to the social factor we get,

$$k(w-w')(w-w')^n = c_2 r_2 (g^{best} - x_i^{t-1}) \tag{20}$$

We interpret as follows, and k plays the role or c_2, and if we consider that $w\prime$ is the optimum value, then the corresponding value in PSO is g^{best}

Comparing the weight update rule (15) and the PSO Eq. (18), considering a single particle we find,

$$\begin{aligned}
&\eta \equiv (1-\beta) \\
&f(w\prime) \equiv v^{t-1} \\
&\eta \sum_{j=1}^{n} \frac{\partial f}{\partial w_j}(w\prime)(w_j^{t-1} - w_j') \equiv -c_1 r_1 (p_i^{best} - x_i^{t-1}) \\
&\alpha \equiv \beta
\end{aligned}$$

By using the above equations for mathematical convenience, we find that M_i^{t-1} works the same way as the momentum term in the (14) i.e. V_{dw} which helps in smoothing and faster convergence to the minimum. Additionally, $c_1 r_1$ serves the same purpose of gradient of the loss function.

$$\eta = (1 - \beta) = (1 - \alpha) \tag{21}$$

This implies the learning rate is tuned by another parameter β. Therefore,

$$\frac{\partial f}{\partial w_j} = \frac{-c_1 r_1}{\eta} = \frac{-c_1 r_1}{1 - \alpha} \tag{22}$$

So, whenever the gradient of the function needs to be computed, the above equivalence can be extended to functions which are non-differentiable!

This can be experimentally proved for various equations. Taking one such example,

$$f(x) = x^2 - 10x + 17 \tag{23}$$

Differentiating (24), we obtain

$$f\prime(x) = 2x - 10 \tag{24}$$

Setting the initial parameters for m-PSO as $c_1 = 0.8,\ c_2 = 0.9,\ \beta = 0.7$ and solving for *30* particles, the algorithm converges towards the global minima in *29* iterations.

4 Representing the Problem

We apply the proposed momentum PSO on both benchmark test optimization functions and habitability optimization problems in astronomy. The benchmark functions considered are standard optimization functions described in Sect. 4.1. The optimization problems tested against, namely CDHS and CEESA, have been described in Sects. 4.2, 4.3. The results of applying the proposed momentum PSO on the above test functions and problems have been discussed in Sect. 5, with supporting values, tables, and graphs.

Usually, a problem may be constrained or unconstrained depending on the search space that has been tackled with. An unconstrained problem's space is the full search space for the particle swarm. The difficulty arises only when it's constrained.

Theophilus, Saha et al. [10] describe a way to handle constrained optimization. We use the same method to represent the test functions and represent few standard optimization problems.

We also consider two habitability scores that are the Cobb–Douglas habitability (CDH) score and the constant elasticity earth similarity approach (CEESA) score. Estimating these scores involves maximizing a production function while observing a set of constraints on the input variables.

4.1 Standard Test Optimization Problems

In this section, we briefly describe the benchmark optimization functions chosen to evaluate our proposed algorithm and compare its performance to that of the weighted PSO and momentum PSO described in Sect. 2. For the purpose of assessing the performance of the proposed algorithm, we have considered single-objective unconstrained optimization functions Rastrigin, Ackley, Sphere, Rosenbrock, Beale, Goldstein-Price, Booth, Bukin N.6, Matyas, Levi N.13, Himmelblau's, Three-hump camel, Easom, Cross-in-tray, Eggholder, Holder table, McCormick, Schaffer N.2, Schaffer N.4, Styblinski–Tang as well as constrained optimization functions Rosenbrock constrained with a cubic line, Rosenbrock constrained to a disc and Mishra's bird.

Rosenbrock function has a long, closed, parabolic-shaped valley where the global minima are present. Finding that valley is an easy task, but converging to the global minima is challenging. Goldstein-Price, Cross-in-Tray, Holder table, and Himmelblau's are 2-dimensional continuous non-convex multimodal functions with four global minima. Three-hump camel is a 2-dimensional continuous non-convex multimodal function with single global minima and three local ones. Easom function has 2 dimensions and is unimodal consisting of multiple local minima but a single global minima where the global minima has a small area relative to the search space. Mishra's Bird is a constrained optimization function which is 2-dimensional and non-convex with two global minima. It is one of the most challenging constraint optimization problem introduced so far.

Rest all the benchmark optimization functions considered except Matyas have multiple local minima and only a single global one. Matyas has no local minima, except the global minima. The results for the above-mentioned benchmark optimization functions are summarized in Tables 1 and 2.

4.2 Representing CDHS

The Cobb–Douglas habitability score is based on the Cobb–Douglas production function. The Cobb–Douglas [12] is a production function very popularly used in economics. It represents the relationship between the values of two or more inputs (particularly physical capital and labour) and the amount of output that can be produced by those inputs.

Table 1 Unconstrained test optimization functions formulas

Name	Formula
Ackley function	$-20\exp[-0.2\sqrt{0.5(x^2+y^2)}]-\exp[0.5(\cos(2x\pi)+\cos(2y\pi))]+e+20$
Rosenbrock 2D function	$(1-x)^2+100(y-x^2)^2$
Beale function	$(1.5-x+xy)^2+(2.25-x+xy^2)^2+(2.625-x+xy^3)^2$
Goldstein–Price function	$[1+(x+y+1)^2(19-14x+3x^2-14y+6xy+3y^2)]+[30+(2x-3y)^2(18-32x+12x^2+48y-36xy+27y^2)]$
Booth function	$(x+2y-7)^2+(2x+y-5)^2$
Bukin function N.6	$100\sqrt{\lvert y-0.01x^2\rvert}+0.01\lvert x+10\rvert$
Matyas function	$0.26(x^2+y^2)-0.48xy$
Lévi function N.13	$\sin^2(3x\pi)+(x-1)^2(1+\sin^2(3y\pi))+(y-1)^2(1+\sin^2(2y\pi))$
Himmelblau's function	$(x^2+y-11)^2+(x+y^2-7)^2$
Three-hump camel function	$2x^2-1.05x^4+\frac{x^6}{6}+xy+y^2$
Easom function	$-\cos(x)\cos(y)\exp(-((x-\pi)^2+(y-\pi)^2))$
Cross-in-tray function	$-0.0001\left[\lvert\sin(x)\sin(y)\exp\left(\lvert 100-\frac{\sqrt{x^2+y^2}}{\pi}\rvert\right)\rvert+1\right]^{0.1}$

The general form of Cobb–Douglas production function is:

$$Q = A\prod_{i=1}^{L} x_i^{\lambda_i} \quad x = (x_1, \ldots, x_L) \tag{25}$$

where

- Q = quantity of output
- A = efficiency parameter
- L = total number of goods
- $x_1, \ldots, x_L$ = (non-negative) quantities of good consumed, produced, etc.
- λ_i = elasticity parameter for good i

The Cobb–Douglas habitability score can be constituted from two components, *the interior score* (Y_i) and *the surface score* (Y_s). Both have to maximized, so the objective function can be represented as follows:

$$Y_i = R^{\alpha}.D^{\beta} \tag{26}$$

$$Y_s = V_e^{\gamma}.T_s^{\delta} \tag{27}$$

Table 2 Unconstrained test optimization functions

Name	Global minimum	m-PSO optimized value	Iterations	Proposed momentum PSO optimized value	Iterations
Ackley function	0	0.001	59	0.001	47
Rosenbrock 2D function	0	$2*10^{-8}$	215	67.14	124
Beale function	0	$4.38*10^{-7}$	68	0	136
Goldstein–Price function	3	3	61	3	53
Booth function	0	$1.07*10^{-7}$	90	$4.09*10^{-7}$	49
Bukin function N.6	0	0.05	503	0.047	191
Matyas function	0	0	55	0	30
Lévi function N.13	0	0	83	0	56
Himmelblau's function	0	$2.51*10^{-7}$	95	0	48
Three-hump camel function	0	$2*10^{-8}$	59	0	36
Easom function	−1	$-3*10^{-10}$	45	−1	40
Cross-in-tray function	−2.062	−2.06	43	−2.064	30

where $R,\ D,\ V_e$ and T_s are density, radius, escape velocity, and surface temperature for a particular exoplanet, respectively, and $\alpha,\ \beta,\ \gamma$, and δ are elasticity coefficients.

The final CDH score can be represented as weighted sum of the interior score and surface score as follows,

$$\underset{\alpha,\ \beta,\ \gamma,\ \delta}{maximize} \qquad Y = w_i Y_i + w_s Y_s$$

$$subject\ to\ \ 0 < \phi < 1, \forall\phi \in \{\alpha,\ \beta,\ \gamma,\ \delta\},$$

$$\begin{aligned} \alpha + \beta - 1 - \tau &\leq 0 \\ 1 - \alpha - \beta - \tau &\leq 0 \\ \gamma + \delta - 1 - \tau &\leq 0 \\ 1 - \gamma - \delta - \tau &\leq 0 \end{aligned}$$

It can be subjected to two scales of production: constant return to scale (CRS) and decreasing return to scale (DRS). The above two Eqs. (26 and 27) are concave under constant returns to scale (CRS), when $\alpha + \beta = 1$ and $\gamma + \delta = 1$, and also under decreasing returns to scale (DRS), $\alpha + \beta < 1$ and $\gamma + \delta < 1$.

4.3 Representing CEESA

The constant elasticity earth similarity approach score is based on the CES production function. The general form of CES production function is:

$$Q = F.\left[\sum_{i=1}^{n} a_i X_i^r\right]^{\frac{1}{r}}$$

where

- Q = quantity of output
- F = factor of productivity
- a_i = share parameter of input i, $\sum_{i=1}^{n} a_i = 1$
- X_i = quantities of factors of production (i = 1, 2… n)
- $s = \frac{1}{1-r}$ = elasticity of substitution

The objective function for CEESA to estimate the habitability score of an exoplanet is:

$$\underset{r,d,t,v,e,\rho,\eta}{\text{maximize}} \quad Y = (r.R^\rho + d.D^\rho + t.T^\rho + v.V^\rho + e.E^\rho)^{\frac{\eta}{\rho}}$$

$$\text{Subject to } 0 < \phi < 1, \forall\phi \in \{r, d, t, v, e\},$$
$$0 < \rho \leq 1,$$
$$0 < \eta < 1,$$
$$(r + d + t + v + e) - 1 - \tau \leq 0,$$
$$1 - (r + d + t + v + e) - \tau \leq 0$$

where E represents orbital eccentricity, and τ is tolerance. Two scales of production are used: constant return to scale (CRS) and decreasing return to scale (DRS). Under DRS, $0 < \eta \leq 1$. Under CRS, $\eta = 1$; hence, the objective function reduces to:

$$\underset{r,d,t,v,e,\rho,\eta}{\text{maximize}} \quad Y = (r.R^\rho + d.D^\rho + t.T^\rho + v.V^\rho + e.E^\rho)^{\frac{1}{\rho}}$$

5 Experiments and Discussions

5.1 Experimental Setup

The confirmed exoplanets catalogue was used for dataset maintained by the Planetary Habitability Laboratory (PHL). We use the parameters described in Tables 3, 4, 5. Surface temperature and eccentricity are not recorded in Earth Units; we normalized these values by dividing them with earth's surface temperature (288 K) and eccentricity (0.017). The PHL-EC records are empty for those exoplanets whose surface temperature is not known. We drop these records from the experiment.

We conveniently first test the **proposed swarm algorithm** on test optimization functions mentioned below. We used *n = 1000* with a *target error* $= 1 * 10^-6$ and *50* particles. Then, the algorithm was used to optimize the CDHS and CEESA objective functions.

Table 3 Constrained test optimization functions

Name	Formula	Global minimum	mPSO Optimized Value	Iterations	Proposed momentum PSO Optimized Value	Iterations
Mishra bird function	$\sin(y)e^{\|[(1-\cos x)^2]\|} + \cos(x)e^{\|[(1-\sin y)^2]\|} + (x-y)^2$ *subjected to* $(x+5)^2 + (y+5)^2 < 25$	−106.76	−106.76	121	−106.76	55
Rosenbrock function constrained with a cubic and a line function	$(1-x)^2 + 100(y-x^2)^2$ *subjected to* $(x-1)^3 - y + 1 \leq 0$ *and* $x + y - 2 \leq 0$	0	0.99	109	0.99	64
Rosenbrock function constrained to a disc	$(1-x)^2 + 100(y-x^2)^2$ *subjected to* $x^2 + y^2 \leq 2$	0	0	69	0	39

Table 4 Estimated Cobb–Douglas habitability scores under CRS

Name	Algorithm	α	β	Y_i	γ	δ	Y_s	Iterations	CDHS
TRAPPIST-1 b	Momentum PSO	0.99	0.01	1.09	0.01	0.99	1.38	130	1.234
	Proposed momentum PSO	0.99	0.01	1.09	0.01	0.99	1.38	75	1.234
TRAPPIST-1 c	Momentum PSO	0.99	0.01	1.17	0.01	0.99	1.21	65	1.19
	Proposed momentum PSO	0.99	0.01	1.17	0.01	0.99	1.21	80	1.19
TRAPPIST-1 d	Momentum PSO	0.01	0.99	0.9	0.01	0.99	1.02	94	0.96
	Proposed momentum PSO	0.01	0.99	0.9	0.01	0.99	1.02	64	0.96
TRAPPIST-1 e	Momentum PSO	0.99	0.01	0.92	0.01	0.99	0.88	30	0.9096
	Proposed momentum PSO	0.99	0.01	0.92	0.2	0.8	0.88	69	0.9096
TRAPPIST-1 f	Momentum PSO	0.99	0.01	1.04	0.95	0.05	0.8	93	0.92
	Proposed momentum PSO	0.99	0.01	1.04	0.7	0.3	0.8	59	0.92
TRAPPIST-1 g	Momentum PSO	0.99	0.01	1.13	0.99	0.01	1.09	117	0.92
	Proposed momentum PSO	0.99	0.01	1.13	0.99	0.01	1.09	66	0.92
TRAPPIST-1 h	Momentum PSO	0.01	0.99	0.81	0.99	0.01	0.68	88	0.7449
	Proposed momentum PSO	0.036	0.963	0.807	0.99	0.01	0.68	86	0.7438

5.2 *Test Optimization Functions*

The particle swarm optimization mentioned in Sect. 2 was tested on the unconstrained test optimization functions defined in Table 2 and constrained test optimization functions in Table 3.

Table 5 Estimated CEESA scores under CRS

Name	Algorithm	r	d	t	v	e	ρ	Iterations	CEESA Score
TRAPPIST-1 b	Momentum PSO	0.556	0.000	0.398	0.045	0.000	0.629	76	1.193
	Proposed Momentum PSO	0.107	0.314	0.578	0.001	3.704	0.999	92	1.126
TRAPPIST-1 c	Momentum PSO	0.117	0.384	0.273	0.225	0.000	0.999	54	1.161
	Proposed Momentum PSO	0.053	0.348	0.212	0.386	0.000	0.999	60	1.161
TRAPPIST-1 d	Momentum PSO	0.127	0.306	0.566	0.000	0.000	0.167	71	0.948
	Proposed Momentum PSO	0.413	0.283	0.304	0.000	0.000	0.820	38	0.882
TRAPPIST-1 e	Momentum PSO	0.455	0.486	0.033	0.027	5.182	0.504	82	0.868
	Proposed Momentum PSO	0.264	0.004	0.626	0.105	0.000	0.936	7	0.897
TRAPPIST-1 f	Momentum PSO	0.718	0.000	0.276	0.006	0.000	0.969	77	0.972
	Proposed Momentum PSO	0.382	0240	0.093	0.284	5.392	0.719	46	0.836
TRAPPIST-1 g	Momentum PSO	0.256	0.232	0.009	0.501	0.002	0.106	4	1.038
	Proposed Momentum PSO	0.337	0.251	0.000	0.412	0.000	0.986	66	1.066
TRAPPIST-1 h	Momentum PSO	0.434	0.259	0.001	0.306	0.000	0.538	73	0.743
	Proposed Momentum PSO	0.295	0.181	0.294	0.230	0.000	0.999	56	0.709

Table 6 Equivalence of new approach to stochastic gradient descent: The table is proof that using m-PSO, we are able to compute the derivative of functions in a non-classical, iterative manner. This could be easily be applied to functions whose derivatives are difficult to find, analytically. We call this "derivative-free" optimization

Parameter Set 1 : $\eta = 0.1,\ \beta = 0.9,\ c_1 = 0.8,\ c_2 = 0.9$									
Parameter Set 2 : $\eta = 0.3,\ \beta = 0.7,\ c_1 = 0.8,\ c_2 = 0.9$									
Function	Derivative	Number of particles	SGD Iterations		Proposed momentum PSO Iterations		Weighted PSO Iterations		
			Set 1	Set 2	Set 1	Set 2	Set 1	Set 2	
$\sin x + \cos^2 x$	$\cos x - 2\sin x \cos x$	15 30 60	16	5	162 91 87	36 27 31	42 42 40	45 37 33	
x^2	$2x$	15 30 60	68	35	65 62 53	27 26 23	24 24 24	32 26 25	
$x^2 - 10x + 17$	$2x - 10$	15 30 60	37	26	66 66 55	29 29 23	25 26 25	29 23 18	
$\sqrt{x^2+2}$	$\frac{x}{\sqrt{x^2+2}}$	15 30 60	535	209	64 57 54	29 25 22	23 22 22	28 25 22	
$\|x\|$	$\frac{x}{\|x\|}$	15 30 60	3472	1600	105 107 89	41 42 41	52 48 43	39 44 40	

5.3 A Comparative Study of PSO Variants on CDHS Scores

The particle swarm optimization mentioned in Sect. 2 was used to optimize the CDHS score to find if the planets are habitable or not. The 3 variants of PSO mentioned in 2 were compared, the Table 4 gives an overview that the proposed momentum particle swarm optimization converges faster to the global minima compared to the other variants. A representational graph attached below gives a clearer picture. The three particle swarm optimizers mentioned in Sect. 2 have been tested with *maximum-iterations = 1000*, *number-of-particles = 50* and *threshold-error to be at* 10^{-6} (Fig. 1).

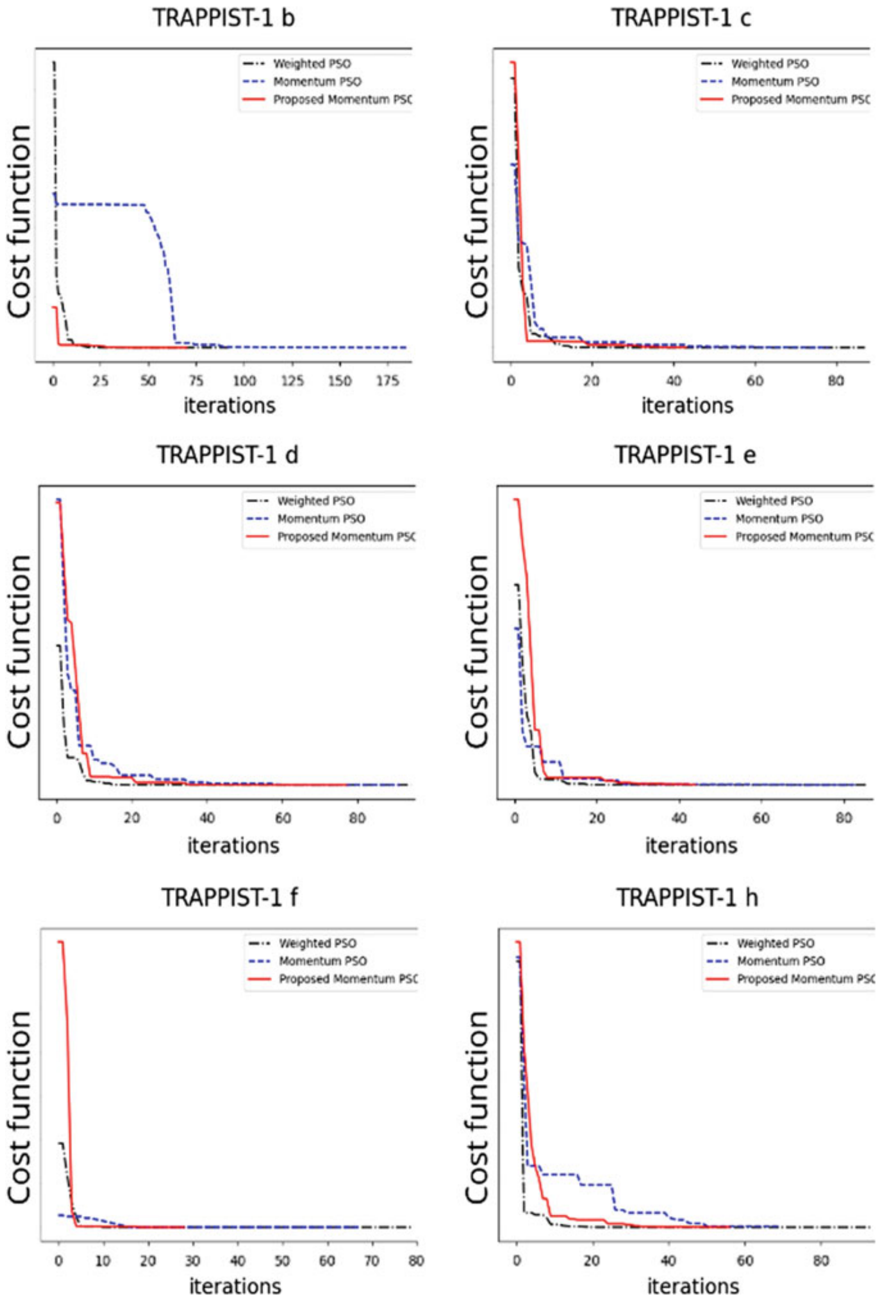

Fig. 1 CDHS: Comparison between 3 variants

6 Conclusion

The paper presents a momentum-enhanced particle swarm approach to solve unconstrained and constrained optimization problems. We also establish the equivalence between the classical stochastic gradient descent/ascent type approaches and the new approach proposed here. This throws some interesting insights to derivative-free optimization, apart from demonstrating empirical evidence of our method surpassing standard PSO in terms of speed of convergence. We conclude by noting that the proposed method can be extended to energy Hamiltonian approach to Momentum PSO as energy can be computed from momentum easily. We also note that acceleration from momentum can also be derived, and therefore, controlling acceleration becomes easier. This is handy when excessive acceleration may push the algorithm further away from minima/maxima. Approximation of derivatives is thus a subtask that can also be accomplished by our approach.

References

1. D.E. Goldberg, *Genetic Algorithms in Search* (Addison-Wesley Longman Publishing, Optimization and Machine Learning, 1989)
2. S. Chen, J. Montgomery, Particle swarm optimization with threshold convergence. in *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, (2013)
3. H. Garg, A hybrid pso-ga algorithm for constrained optimization problems. Appl. math. comput. **274**, 292–305 (2015)
4. P.K. Tripathi, S. Bandyopadhyay, S.K. Pal, Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. Inf. Sci. **177**, 5033–5049 (2007)
5. J. Sun, B. Feng, W. Xu, Particle swarm optimization with particles having quantum behavior. in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 1, (2004), pp. 325–331
6. A.J. Nebro, J.J. Durillo, J. Garcia-Nieto, C.C. Coello, F. Luna, E. Alba, Smpso: A new pso-based metaheuristic for multi-objective optimization. in *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM 2009—Proceedings*, (2009), pp. 66–73
7. T. Xiang, J. Wang, X. Liao, An improved particle swarm optimizer with momentum. in *2007 IEEE Congress on Evolutionary Computation*, (2007), pp. 3341– 3345
8. K. Bora, S. Saha, S. Agrawal, M. Safonova, S. Routh, A. Narasimhamurthy, Cd-hpf: New habitability score via data analytic modeling. Astron. Comput. **17**, 129–143 (2016)
9. S. Saha, S. Basak, K. Bora, M. Safonova, S. Agrawal, P. Sarkar, J. Murthy, Theoretical validation of potential habitability via analytical and boosted tree methods: An optimistic study on recently discovered exoplanets. Astronomy and Comput., **23**, 141–150 (2018)
10. A. Theophilus, S. Saha, S. Basak, J. Murthy, A novel exoplanetary habitability score via particle swarm optimization of ces production functions. (2018)
11. J. Kennedy, R. Eberhart, Particle swarm optimization. in *Proceedings of ICNN'95—International Conference on Neural Networks*, vol. 4, (1995), pp.1942–1948
12. C.W. Cobb, P.H. Douglas, A theory of production. The American Economic Review, vol. 18, No. 1, Supplement, Papers and Proceedings of the Fortieth Annual Meeting of the American Economic Association (1928), pp. 139–165