



Hybrid Particle Swarm Optimization with Sine Cosine Algorithm and Nelder–Mead Simplex for Solving Engineering Design Problems

Hussam N. Fakhouri¹ · Amjad Hudaib¹ · Azzam Sleit¹

Received: 22 June 2019 / Accepted: 3 December 2019 / Published online: 2 January 2020
© King Fahd University of Petroleum & Minerals 2020

Abstract

This paper introduces a novel hybrid evolutionary algorithm that combines particle swarm optimization (PSO) algorithm with sine–cosine algorithm (SCA) and Nelder–Mead simplex (NMS) optimization technique. However, the algorithm of PSO has some drawbacks like locating local minima rather than global minima, low converge rate and low balance between exploration and exploitation. In this paper, the combination of PSO algorithm with update positions mathematical equation in SCA and NMS technique is presented in order to solve these problems. So a new hybrid strategy called PSOSCANMS is introduced. The SCA algorithm is based on the behavior of sine and cosine functions in the mathematical formula used for solutions. However, the NMS mathematical formulations attempt to replace the worst vertex with a new point, which depends on the worst point and the center of the best vertices. The combined effect of both mathematical formulations of PSO ensures a consistency of exploitation and exploration that makes the search in the search space more effective. Further, it escapes into the local minimum issue and resolves the low converge rate problem. In order to test PSOSCANMS's performance, a set of 23 well-known unimodal and multimodal functions have been benchmarked. Experimental results showed that PSOSCANMS is more successful than PSO and outperforms the other state-of-the-art compared algorithms over the tested optimization problems. Moreover, an engineering design problem such as spring compression, welded beam is also considered. The result of the problems in engineering design and application problems shows that the algorithm proposed is relevant in difficult cases involving unknown search areas.

Keywords Optimization · Meta-heuristic · Particle swarm · Sine–cosine optimizer · Nelder–Mead simplex

1 Introduction

In recent decades, several population-based random optimization techniques have been widely used for solving global optimization problems, for example evolutionary algorithms and swarm intelligence optimizations. The particle swarm optimization (PSO) meta-heuristic is part of a wider class of the swarm intelligence algorithms that aim to solve global optimization problems [1].

Meta-heuristic algorithms are needed for the optimization process in various fields of science including monitoring, classification, engineering, etc. In order to find optimum solutions, they try to satisfy the convergence needs. Fast

and successful convergence is needed especially for online (real-time) applications [2]. Convergence time has been shortened by developing new or hybrid optimization algorithms over time. In order to do this, two ways of achieving better results were used, new methods or a hybridizing of some optimization methods. An alternative (hybridization) is being followed in this study, and an effective method of benchmarking optimization is being examined. PSO is being used in several disciplines to many newly developed optimization algorithms [3]. An algorithm or inspirational formula can, however, increase the efficiency of the PSO standard. The hybrid or improved PSO architectures can therefore be developed. These modifications are necessary to improve convergence. The hybrid algorithm, however, is usually assessed by a set of benchmarking functions which examine the efficiency of the developed hybrid optimizer [2].

Many hybrid PSO algorithms were developed. The algorithm for original particle swarm has been modified by Mendes et al. [4], and the fully informed particle swarm is

✉ Hussam N. Fakhouri
h.fakhouri@ju.edu.jo

¹ Computer Science Department, King Abdullah II School of Information Technology, The University of Jordan, Amman, Jordan



designed to fully inform individuals. Different topologies for numerical function optimization of fully informed PSO were also examined [5]. The results were better based on their study, and fully informed PSO performed canonical PSO on the optimization of the benchmarking function [6, 7].

A hybrid PSO, based on the sine–cosine algorithm (SCOSCALF), has been developed to address optimization problems [8]. The exploration capability of the original PSO algorithm is increased by combining SCA and Levy flight in the PSOSCALF algorithm and is also avoided in the local minimum.

This article proposes a combination of the Nelder–Mead simplex (NMS) position update equations and the cosine optimization of the scene algorithm (SCA). The proposed hybridization algorithm significantly speeds up the convergence rate for a number of famous benchmark issues. It adds a user component to the algorithm for the optimization of particle swarming that maintains a suitable balance among exploration and exploitation. One of the population-based algorithms recently proposed for optimization solutions is the SCA optimization algorithm [9].

In the proposed PSOSCANMS algorithm, each new solution shall be calculated using sine and cosine function characteristics and then using Nelder–Mead simplex function characteristics. This algorithm enables exploration and exploitation capabilities to oscillate and change the value of these trigonometric functions. The Nelder–Mead simplex technique hybridization SCA thus allows for a further research into the design space to find an optimal solution.

The position of the particle is updated at the proposed PSOSCANMS with the best value between the three methods PSO, SCA and NMS for each iteration; PSO is the first to acquire new location of the particle, SCA is updated to the position of the particle when SCA's best value is better than the value found for PSO, and the new position of the particle is found by SCA equations. Otherwise, NMS will update the particle's position.

The main advantages of the hybridization of NMS and SCA with PSO are that they are easy to implement deterministic local optimization techniques which use only function evaluations to direct their search, and hence, they do not require the computation of derivative. Further they increased the search capability of PSO. Another main advantage is that the hybrid that is to overcome the problems that PSO has such as locating local minima rather than global minima, low convergence rate and low balance between exploration and exploitation. However, these problems are solved by changing the direction of particle movement at each iteration.

The novelty of this work is that the hybrid algorithm of NMS and SCA with PSO will offer five solutions at each step of the optimization process. The particle can choose the best solution between all these points rather than one point and one movement direction in PSO. This will extend the

exploration feature of particles to cover most of the search space. Moreover, NMS will cover three points around each particle in the search space and SCA will cover two more points using sine and cosine functions. This expansion will make each particle to choose the best position between five points rather than heading in one direction using one global point. However, if one of the five points has better solution than the one that PSO has, then the particle direction will be toward this area which contains a better value and possibly has the global minimum solution and not local minimum.

The next sections of this article are as follows: Sect. 2 describes the basics of the PSO algorithm. The SCA and the Nelder–Mead simplex algorithms are presented in the third and fourth sections. The details of the PSOSCANMS algorithm proposed are described in Sect. 3. Section 4 compares the ability of the suggested algorithm to optimize benchmarking functions with family PSO and other algorithms. Section 5 has been assessed for the ability of the PSOSCANMS method to resolve engineering problems. In Sect. 6, the paper is finalized.

2 Optimization Algorithms

2.1 Particle Swarm Optimization (PSO) Algorithm

The overall PSO's idea of how birds or fish schools move is based. The population is called a swarm in this algorithm and every spot is a particle. These swarms are solutions that are possible. The particles in the search area of the objective function are randomly initialized. Each particle makes a compromise between its own best position in history (i.e., pbest), its strongest position and its random look (i.e., gbest) in the course of consecutive iterations following initialization. In PSO, every particle has two characteristics (velocity vector V and vector X) and moves in the search space at a time of velocity adjustment based on the experience of the particle and the experience of the particle companion. The speed and position of the part according to Eqs. 1 and 2 shall be updated mathematically [10]:

$$\text{Vid}(t + 1) = w * \text{vid}(t) + c_1 * r_1 * (\text{pid}(t) - \text{xid}(t)) + c_2 * r_2 * (\text{pgd}(t) - \text{xid}(t)) \quad (1)$$

$$\text{Xid}(t + 1) = \text{xid}(t) + \text{vid}(t + 1) \quad (2)$$

where $\text{Vid}(t + 1)$ and Vid are particle velocities at iterations t and $t + 1$. Pid is particle's best position. Pgd is neighborhood's best position at iteration t . c_1 and c_2 are acceleration coefficients that reflect the weight of stochastic acceleration terms pulling each particle to pbest and gbest positions, respectively. r_1 and r_2 denote two uniformly distributed

random numbers (0, 1). ω is the weight of inertia used in global and local search balance. A large weight of inertia generally makes global exploration easier while a small weight of inertia facilitates local use. And xid is the position of the t -iterative particle. PSO starts by generating particles randomly in the search space [11].

The disadvantage of PSO is that the convergence of the swarm is slow. This problem is based on the fact that particles converge for the global best, because all particles move to a single point between the best global position and the best personal position. This is not ensured for to be the global optimum value [3]. The rapid flow of information between particles is another reason for this problem, resulting in the creation of similar particles with a loss of diversity which increases the risk of trapping them in local optima. However, PSO is likely to pre-converge over multimodal fitness areas and PSO is relatively weak in local search capacities. Moreover, numerous studies have demonstrated that PSO can easily be stored in an optimal solution region in the local market and that its complex multimodal functions are slowly converging. It is important to balance exploration and exploitation to overcome those disadvantages, and we thus use the PSO hybrid strategy with SCA and NMS to improve the performance of PSO [12].

2.2 Sine–Cosine Algorithm (SCA)

In 2016, Mirjalili proposed sine–cosine algorithm (SCA) to solve problems related to optimization. SCA produces a variety of random solutions and then approaches them with the optimum solution using equations such as sine and cosine functions [9]. Of course, apart from the characteristics of these functions, the distance between the best particle of the population and each of the alternatives affects their motion. The SCA algorithm offers fewer operators than other algorithms in terms of operation and exploration. This algorithm applies sine and cosine functions to update solutions for exploration and exploitation using equations (Eqs. 3 and 4) [9]:

$$X_i(t + 1) = X_i(t) + r_1 \sin(r_2) |r_3 P^t - X_i(t)| \tag{3}$$

$$X_i(t + 1) = X_i(t) + r_1 \cos(r_2) |r_3 P^t - X_i(t)| \tag{4}$$

where $X_i(t)$ is the position of the current solution in i th dimension at t th iteration, r_1 , r_2 and r_3 are random numbers, P^t is position of the destination point in i th dimension and $| \cdot |$ symbol indicates the absolute value.

Equations (3) and (4) indicate the space defined by the proposed equations by the two search solution solutions. The cyclical synthetics and cosine make it possible to rearrange a solution around another one. There can be guaranteed space use between two solutions. In order to explore the

search area, solutions must also be able to look outside the space between their destinations. Yet changes in the range of sine and cosine functions require a solution to change the range of sine and cosine functions outside or inside the space between oneself and another alternative. If the random number is defined for r_2 in $[0, 2\beta]$ in Eq. 3, the changed position inside and outside can be achieved [13].

2.3 Nelder–Mead Simplex

A very popular multi-dimensional optimization search method, the Nelder–Mead simplex algorithm (31), was released in 1965. However, Spendley et al. [14] and Nelder and Mead [15] subsequently developed the simplex search method [15, 16]. This algorithm, which only uses the basic operations of reflection, expansion, contract and shrink to find the solution directly, does not need to obtain the first or second derivative for the property of derivative-free. The Nelder–Mead algorithm keeps a simple, optimum point approximation. The vertices are sorted by the value of the objective function. The algorithm tries to replace the worst corner with a new point, depending on the worst and middle of the best corners [17].

A construction called a simplex is used in the Nelder–Mead approach. $N + 1$ solutions, $X(k)$, $k = \{1, 2, n + 1\}$ [6], are a simplex in n dimensions. This is like a two-dimensional plane triangle. Each step assesses the solutions and identifies the worst solution. Simplex center is calculated as described in Eq. 5 [17].

$$C = (\sum X(k))/n \tag{5}$$

A simplex S in R_n is defined as the convex hull of $n + 1$ vertices $x_0, \dots, x_n \in R_n$. For example, a simplex in R_2 is a triangle as shown in Fig. 1. However, a simplex-based direct search method begins with a set of $n + 1$ points $x_0, \dots, x_n \in R_n$ that are considered as the vertices of a working simplex S , and the corresponding set of function values at the vertices $f_j = f(x_j)$, for $j = 0, \dots, n$. The initial working simplex S has to be *non-degenerate*, i.e., the points x_0, \dots, x_n must not lie in the same hyperplane. The method then performs a sequence of transformations of the working simplex S , aimed at decreasing

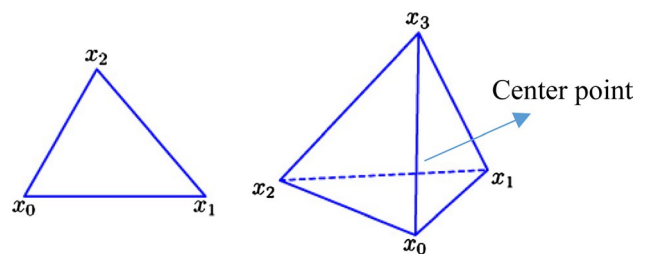


Fig. 1 Triangle simplex center

the function values at its vertices. At each step, the transformation is determined by computing one or more test points, together with their function values, and by comparison of these function values with those at the vertices. This process is terminated when the working simplex S becomes sufficiently small in some sense, or when the function values f_j is close enough in some sense (provided f is continuous) [15, 17].

2.4 Swarm Intelligence (SI) and Meta-heuristics

Swarm intelligence (SI) inspired algorithms. SI describes the collective behavior of a decentralized self-organizing system. It is considered in the theory of artificial intelligence as an optimization method [18]. The term was introduced by Beni and Jin [18], in the context of a system of cellular robots. SI has very wide inspired meta-heuristics that increase every day taking into consideration the hybrid algorithms that result from combining any two meta-heuristics. The first SI meta-heuristic was particle swarm optimization (PSO), a meta-heuristic proposed by Eberhart and Kennedy [10], and they took inspiration from flying birds in nature to propose the PSO meta-heuristic. PSO is based on the collaboration of individuals with each other; each particle moves and at each iteration the closest to the optimum and then communicates with other particles to send its position to change their position. After PSO, many SI meta-heuristics have been proposed and recent ones are: ant colony optimizer (ACO) [19]; fast evolutionary programming (FEP) [20]; cuckoo search (CS) algorithm [21]; bees algorithm (BA) [13]; artificial bee colony (ABC) algorithm [22]; gravitational search algorithm [23]; firefly algorithm (FA) [24]; grey wolf optimizer (GWO) [25]; dolphin echolocation (DE) [26]; fruit fly optimization algorithm (FFOA) [27]; glow-worm swarm optimization (GSO) [28]; tree-seed algorithm (TSA) [29]; cuckoo optimization algorithm (COA) [30]; hunting search (HS) [31]; atom search optimization (ASO) [32]; butterfly optimization algorithm: a novel approach for global optimization (BOA) [33]; augmented grey wolf optimizer (AGWO) [34]; enhanced grey wolf optimizer (EGWO) [33]; bat algorithm (BA) [35]; moth swarm algorithm (MSA) [36]; hybrid PSO–GWO [36]; and supernova optimizer [3].

3 The Proposed Hybrid Particle Swarm Optimization with Sine–Cosine Algorithm and Nelder–Mead Simple (PSOSCANMS) Algorithm

Although PSO algorithm has the excellence of high-speed computation with a small number of parameters and simplicity in implementation; it has two major deficiencies

including slow convergence and falling into the local minima. However, in recent studies such as Singh et al. [37], the PSO velocity and position updating equations have been enhanced by the behavior of sine and cosine [37]. Further, Chin et al. (2018) used hybridization with SCA, where new solutions are generated and the ability to examine the search space is increased [2]. In addition, the Nelder–Mead simplex (NMS) technique was used to speed up the convergence rate [1]; the hybridization with Nelder–Mead simplex, as discussed in Sect. 3, has been a popular means of accelerating convergence, for example hybridization of NMS with evolutionary algorithms. This technique has been reported by several papers [2].

For hybridization of PSO and other optimization algorithms, one of the two (I) tandem and (II) cascading methods has been widely used. If the approach is tandem, the entire population is divided into two subsets in tandem hybridization strategy. One or more of the algorithms are used for a set of individuals. At the end, they combine the two subsets. However, when the approach is cascaded, the stochastic optimization process is used for everyone in the population and the solutions obtained are further improved by the combined algorithm. We used cascade hybridization in this paper.

In this paper, the equations of position updating on SCA (Eqs. 3 and 4) are combined in Eq. 6 [9] and added to PSO. After that, the update position of Nelder–Mead simplex (Eqs. 7–9) [17] is secondly cascaded after SCA (Eq. 6). These two enhancements will modify the PSO particles movement strategy to be able for better search in the search space for better detecting the global optima and to speed up the converge rate.

$$x_i(t+1) = \begin{cases} X_i(t) + r_1 \sin(r_2) |r_3 p_t - X_i(t)|, & r_4 < .05 \\ X_i(t) + r_1 \cos(r_2) |r_3 p_t - X_i(t)|, & r_4 \geq .05 \end{cases} \quad (6)$$

where parameter r_1 determines the region (or direction of motion) in the next position that might be either in or outside the space between the solution and the destination. The r_2 parameter defines the distance to and from the destination of the movement. Parameter r_3 gives random weight to the destination, so that the effect of the destination in defining the distance is stochastically emphasized ($r_3 > 1$) or decanted ($r_3 < 1$). In conclusion, the r_4 parameter switches equally between the sine and cosine components of Eq. 6 [9].

In Eq. 7, the summation is excluded and the worst point is reflected in the center; however, the worst point W shall be replaced with point r reflected in the simplexes, but the simplex will be extended further if the point reflected is better than any simplex solution, as described in Eq. 8.

$$R = C + (C - W) \quad (7)$$



$$Re = C + \eta(C - W) \quad (8)$$

where W is the worst solution. C is the simplex center calculated in Eq. 5, and the expansion coefficient is called η . R is the reflected solution.

If the reflected R solution was worse than W , however, the simplexes would be contracted and the reflected solution would be positioned on the same center. When solution R is not worse than solution W , but worse than any other solution, simplex still contracts but reflection is permitted to remain on the other side of the centroid. Reflection is done accordingly as described in Eq. 9 [10].

$$Rc = C \pm \kappa(C - W) \quad (9)$$

where κ is the contraction coefficient in the above equation. In the next step, the new [17] R , Re or Rc will replace solution W . It is possible to run the simplex algorithm several steps prior to completion. Every step of the simplex shall be referred to as a reverse.

In the new proposed hybrid algorithm, at each iteration the particles position is first updated according to PSO equations. The particles at this point will move toward the global optimum point (gbest). Then its motion will be affected by SCA and NMS equations, respectively. However, if the solution found by any of the two algorithms gives better optimum solution than PSO then the particles will move toward the better solution. However, if the particle position does not improve at the end of every iteration of the optimization process, then the combination of SCA and Nelder–Mead simplex will generate a new position for every particle in the search space.

3.1 PSOSCANMS Algorithm

The PSOSCANMS steps are described in Algorithm 1. The populations are represented in the same set as the actual parameter vectors $x_i = (x_1, \dots, x_D)$, $I = 1, \dots, N$, in which D is a problem dimension and N the population size. At the beginning of the search, the single vectors x_i are randomly initialized. $MaxIt$ is the maximum iteration, the lower and upper bounds in search space are X_{min} and X_{max} . At the

start of the proposed algorithm are those parameters set. The position and velocity of particles are initially generated randomly.

However, by calculating the objective function value, the best local (XpBesti) for each particular and the global (Xgbest) of the whole population is determined. During each iteration, the value of the constants c_1 , c_2 and w is determined. Before updating the vector for the velocity of each particle, the limit value of the particle is checked. Firstly, particle velocity and position are updated by the simple PSO Eqs. 1 and 2. Secondly, Eq. 6 which is the SCA algorithm shall determine the next position of the particle. The fitness value of applying the equations is better than the value obtained from Eqs. 1 and 2. However, in this case, this particle has a random value. When $rand(0,5)$, in Eq. 6 the particle position is calculated.

Thirdly, the algorithm then applies to Eqs. 7–9, respectively, which constitutes to Nelder–Mead simplex. Then, the global value is chosen among the three equations if the founded solution is better than the previous founded solution. The algorithm then saves the best solutions, assigns them to the destination and updates further solutions. In the meantime, the upper and lower search space range is updated so that search space is used to achieve the global optimal point at each iteration.

New particles are produced in the design space by upgrading the speed and position of all particles. After that, all new particles will have an objective function determined. Then, each new particle's objective value is compared with its best personal experience (Xpbesti). If the location is improved, Xpbesti will be updated and its index resets to zero and the index will increase by one piece, otherwise. Finally, for all particles Xgbest is determined. The process above is repeated up to the $MaxIt$ iteration number.

PSOSCANMS has increased the search capacity to include more particle positions during the motion of particles, which increased the possibility of finding the global minimal and avoiding local minimal problems rather than stuck in a local lowest point.

Algorithm 1 Hybrid Particle Swarm Optimization with Sine Cosine Algorithm and Nelder Mead Simplex**Input:** particles, MaxIt , dimension D, number of population (N), Xmin, Xmax., constant (c1), constant (c2)**Output:** Gbest**initial:** P \leftarrow randMaxIt \leftarrow M , % Where M The maximum iteration $X_i = (X_1, \dots, X_D), i = 1, \dots, N$ $X_{Pi} \leftarrow$ RandP % Where RandP is random positions $V_i \leftarrow$ RandV % Where Randv is random velocity**for** i \leftarrow 1: NFitness (Xi) = fobj (F_{xi}) % Where F is the objective function of the optimization problem**End for**Set Xi \leftarrow XpBestgbest \leftarrow min(Fitness (Xi))**Begin:****for** i \leftarrow 1: N $Vid(t+1) = w * vid(t) + c1 * r1 * (pid(t) - xid(t)) + c2 * r2 * (pgd(t) - xid(t))$ $Xid(t+1) = xid(t) + vid(t+1)$ Fitness (Xi) = fobj (F_{xi})**End for****for** i \leftarrow 1: NXP \leftarrow X_{Pi} % Where X_{Pi} is the updated position of the particle

Fitness (Xi) = fobj (F)

End forif f(X_{pBest}) < f(X_{gBest})X_{gBest} = X_{pBest}**End if****for** i \leftarrow 1: N

if trial(i) < 0.5

 $X_i(t+1) = X_i(t) + r_1 \sin(r_2) |r_3 P^i - X_i(t)|$ **else** $X_i(t+1) = X_i(t) + r_1 \cos(r_2) |r_3 P^i - X_i(t)|$ **end if**Fitness (Xi) = fobj (F_{xi})if f(X_i(t)) < f(X_{pBest})

trial(i) = 0

X_{pBest} = X_i**end if**if f(X_{pBest}) < f(X_{gBest})X_{gBest} = X_{pBest}**end if****for** i \leftarrow 1: N

R = C + (C - W)

Fitness (Xi) = fobj (F_{xi})if f(X_{pBest}) < f(X_{gBest})X_{gBest} = X_{pBest}**end if****End for****for** i \leftarrow 1: NRe = C + η (C - W)Fitness (Xi) = fobj (F_{xi})if f(X_{pBest}) < f(X_{gBest})X_{gBest} = X_{pBest}**end if****End for****for** i \leftarrow 1: NRc = C + κ (C - W)Fitness (Xi) = fobj (F_{xi})**End for**if f(X_{pBest}) < f(X_{gBest})X_{gBest} = X_{pBest}**end if**Xmin < X_{gBest} < (Xmax)**End for****End Begin**

Table 1 Benchmark functions (F1–F23)

No.	Function name	No.	Function name
F1	Shifted sphere	F12	Schwefel's problem 2.13
F2	Shifted Schwefel's problem 1.2	F13	Griewank's plus Rosenbrock's function
F2	Shifted rotated high conditioned elliptic function	F14	Shifted rotated expanded Scaffer's F6
F3	Shifted rotated high conditioned elliptic function	F15	Hybrid composition
F4	Schwefel's problem 2.6 with global optimum on bounds	F16	Rotated hybrid composition space function
F5	Schwefel's problem 2.6 with global optimum on bounds	F17	Rotated hybrid composition function with noise in fitness
F6	Shifted Rosenbrock's	F18	Rotated hybrid composition function
F7	Shifted rotated Griewank's function without bounds	F19	Rotated hybrid composition function with a narrow basin for the global optimum
F8	Shifted rotated Ackley's function with global optimum on bounds	F20	Rotated hybrid composition function with the global optimum on the bounds
F9	Shifted Rastrigin's	F21	Rotated hybrid composition
F10	Shifted rotated Rastrigin's	F22	Rotated hybrid composition function with high condition number matrix
F11	Shifted rotated Weierstrass	F23	Non-continuous rotated hybrid composition

When the termination condition is archived, the PSOS-CANMS algorithm terminates the optimization process and the iteration number reaches its maximum limit by default. However, every other termination condition, such as the maximum number of function assessments or the accuracy of the global optimal achieved, can be considered. In order to avoid local minimum standards, and for the following reasons, the proposed algorithm can theoretically identify the global optimum for optimization problems first and then improves the PSO algorithm.

The following section examines the efficacy of PSOS-CANMS with a variety of standard functions representing problems known for testing optimization algorithms. The efficiency of PSOSCANMS algorithm is investigated, analyzed and confirmed.

4 Results and Discussion

In order to evaluate the performance of the proposed PSOS-CANMS algorithm, extensive simulations have been carried out. For comparison, several state-of-the-art optimization algorithms (PSO, BOA, GSA, FEP, GWO, ASO, CHPSO) have been applied.

4.1 Benchmark Functions and Parameter Setting

PSOSCANMS has been experimented over 23 benchmark functions utilized by many researchers to test performance for global optimization algorithms [3, 6, 9, 30, 33, 34, 36, 37].

These benchmark functions are listed in Tables 1 [5]. Further function descriptions are available in Arora and Singh [37]. However, the unimodal functions are functions

{F1–F7}. Functions {F8–F13} represent multimodal basic functions, and fixed-dimension multimodal functions are from function {F14–F23}.

The unimodal {F1–F7} functions have only one optimum global point, no local optimum points, and they are therefore suited for benchmarking algorithms exploitation feature. However, the functions {F8–F13} are classified as multimodal to test the exploration feature and to avoid local optima optimization algorithms because they have many local optima points. The algorithms tested should avoid falling at those points to reach the global optimum point. The functions {F14–F23} are functions of a fixed multimodal dimension, where the fixed-dimension functions mathematical formulations do not allow us to adjust the number of design variables, but offer different search spaces in contrast to multimodal test functions. Graphical representation for certain benchmark functions that are considered in this research is shown in Fig. 2.

4.2 Experiment and Comparison

The experiment must be performed up to n number of times in order to produce stable statistic results for the performance of meta-heuristic algorithms. And for testing the stability of the algorithm every run must be carried out to m number of iterations [37]. The same experimental procedure was followed to produce and report and then verify the performance of PSOSCANMS algorithm, including statistical measurement (mean, standard deviation, Min and Max). A population size and a maximum iteration of 51 and 10,000, respectively, were used for all algorithms.

Results of the comparing PSOSCANMS with state-of-the-art meta-heuristic algorithms (BOA, ASO, GWO, CHPSO, SCA, PSOGWO and MSA) as well as PSO over

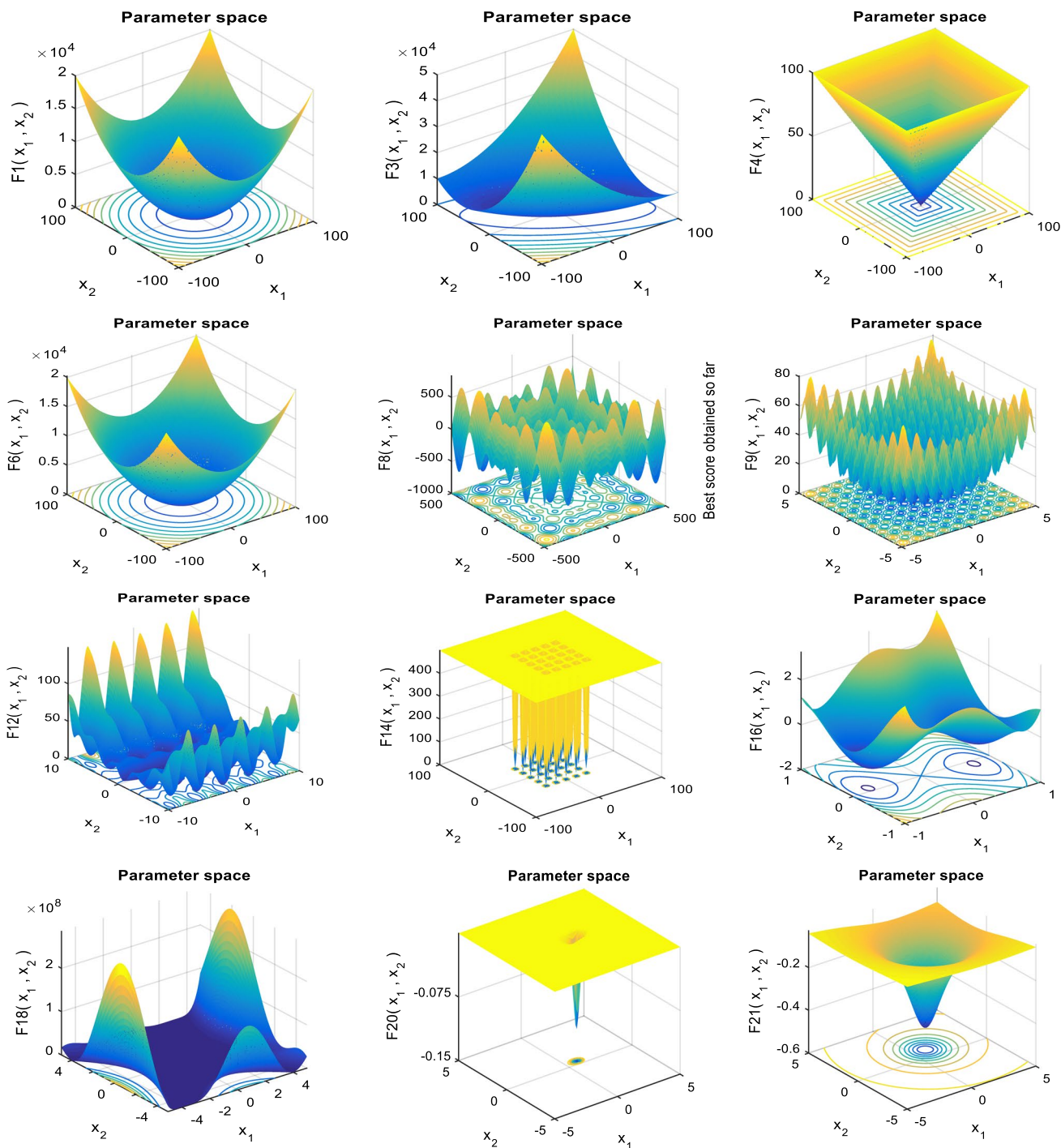


Fig. 2 Representation of benchmark mathematical functions

tested benchmark are described in Table 2. The grouped statistical test results for the 23 functions are also demonstrated. As seen in the table, the results show that the hybrid strategy in PSOSCANMS has clearly enhanced PSO. The results over six of the tested unimodal test

functions (F1–F7) show that the improved method has successfully contributed to PSOSCANMS performance over PSO. However, in traditional PSO all of the particles update their position to move toward one point that may be a local minima and not a global minimum, PSOSCANMS

Table 2 Comparison of optimization results of the algorithms (PSOSCANMS, PSO, BOA) obtained for the unimodal, multimodal and fixed-dimension multimodal benchmark functions

	PSOSCANMS						PSO						BOA					
	MEAN	STD	MIN	MAX	MEAN	STD	MIN	MAX	MEAN	STD	MIN	MAX	MEAN	STD	MIN	MAX		
	F1	1.9514e-130	6.2E-130	5.8E-161	2E-129	3.68868E-07	3.79E-07	2.7E-08	9.36E-07	1.07E-11	6.94E-13	9.82E-12	1.16E-11	1.07E-11	6.94E-13	9.82E-12	1.16E-11	
F2	3.66E-71	1.16E-70	1.07E-87	3.66E-70	0.000377991	0.000243	0.000108	0.000725	3.75E-09	8.91E-10	2.43E-09	4.49E-09	3.75E-09	8.91E-10	2.43E-09	4.49E-09		
F3	4.6E-135	1.4E-134	2.7E-181	4.6E-134	22.30644089	3.721705	18.02812	27.28733	1.1E-11	9.47E-13	1E-11	1.24E-11	1.1E-11	9.47E-13	1E-11	1.24E-11		
F4	1.18E-61	2.48E-61	2.47E-74	6.4E-61	0.523290206	0.138093	0.387735	0.687986	5.5E-09	5.12E-10	4.86E-09	6.06E-09	5.5E-09	5.12E-10	4.86E-09	6.06E-09		
F5	20.7945	0.942194	19.45268	22.47247	122.3570204	112.8315	26.55671	287.264	28.90727	0.022202	28.87978	28.93657	28.90727	0.022202	28.87978	28.93657		
F6	2.32E-10	7.33E-10	3.08E-33	2.32E-09	2.22227E-07	2.41E-07	4.59E-08	5.67E-07	5.315606	0.205174	5.079535	5.561694	5.315606	0.205174	5.079535	5.561694		
F7	3.79E-06	3.93E-06	2.05E-07	1.34E-05	0.066328262	0.017126	0.040264	0.082302	0.000659	0.000241	0.000331	0.000906	0.000659	0.000241	0.000331	0.000906		
F8	-3309.43	372.717	-4024.66	-2800.91	-6370.08483	623.3395	-6843.69	-5303.71	-3979.72	200.4715	-4326.74	-3813.6	-3979.72	200.4715	-4326.74	-3813.6		
F9	0	0	0	0	44.73215272	6.568168	38.28548	52.99159	3E-05	2.54E-14	3E-05	3E-05	3E-05	2.54E-14	3E-05	3E-05		
F10	8.88E-16	0	8.88E-16	8.88E-16	0.000176925	0.000154	6.03E-05	0.000447	5.05E-09	3.69E-10	4.49E-09	5.48E-09	5.05E-09	3.69E-10	4.49E-09	5.48E-09		
F11	0	0	0	0	0.006404683	0.007091	1E-06	0.017227	1E-06	0	1E-06	1E-06	1E-06	0	1E-06	1E-06		
F12	1.74E-32	1.6E-33	1.62E-32	2.18E-32	7.42958E-09	1.2E-08	6.31E-11	2.84E-08	0.394278	0.067629	0.295398	0.465696	0.394278	0.067629	0.295398	0.465696		
F13	0.163939	0.25497	4.31E-32	0.791505	0.002198387	0.004913	1.02E-08	0.010987	2.523125	0.289541	2.290027	2.963217	2.523125	0.289541	2.290027	2.963217		
F14	0.998004	0	0.998004	0.998004	1.246510603	0.497014	0.998004	1.992031	1.157202	0.214295	0.998004	1.391922	1.157202	0.214295	0.998004	1.391922		
F15	0.000307	4.24E-19	0.000307	0.000307	0.000751076	0.000241	0.000387	0.000987	0.000332	1.51E-05	0.000312	0.000348	0.000332	1.51E-05	0.000312	0.000348		
F16	-1.03163	0	-1.03163	-1.03163	-1.03162845	0	-1.03163	-1.03163	-3602.94	5029.103	-10.500.3	-3.31376	-3602.94	5029.103	-10.500.3	-3.31376		
F17	0.397887	0	0.397887	0.397887	0.397887	0	0.397887	0.397887	0.397887	2.3E-08	0.397887	0.397887	0.397887	2.3E-08	0.397887	0.397887		
F18	3	6.28E-16	3	3	3	8.31E-16	3	3	3.022434	0.025092	3.002907	3.064524	3.022434	0.025092	3.002907	3.064524		
F19	-3.86278	0	-3.86278	-3.86278	-3.86278215	0	-3.86278	-3.86278	-4.23757	0.246826	-4.502	-3.97779	-4.23757	0.246826	-4.502	-3.97779		
F20	-3.25066	0.06512	-3.322	-3.2031	-3.29227189	0.059447	-3.322	-3.2031	-4.84266	3.553986	-11.1946	-3.12811	-4.84266	3.553986	-11.1946	-3.12811		
F21	-10.1532	8.88E-16	-10.1532	-10.1532	-9.14271417	2.259514	-10.1532	-5.10077	-4.77318	0.088074	-4.84147	-4.6196	-4.77318	0.088074	-4.84147	-4.6196		
F22	-10.4029	1.26E-15	-10.4029	-10.4029	-10.4029406	8.88E-16	-10.4029	-10.4029	-4.81466	0.11974	-4.96045	-4.71291	-4.81466	0.11974	-4.96045	-4.71291		
F23	-10.5364	1.26E-15	-10.5364	-10.5364	-9.45482401	2.418499	-10.5364	-5.12848	-4.74479	0.119108	-4.87732	-4.58441	-4.74479	0.119108	-4.87732	-4.58441		

Table 3 Comparison of optimization results of the algorithms (ASO, GWO, CHPSO) obtained for the unimodal, multimodal and fixed-dimension multimodal benchmark functions

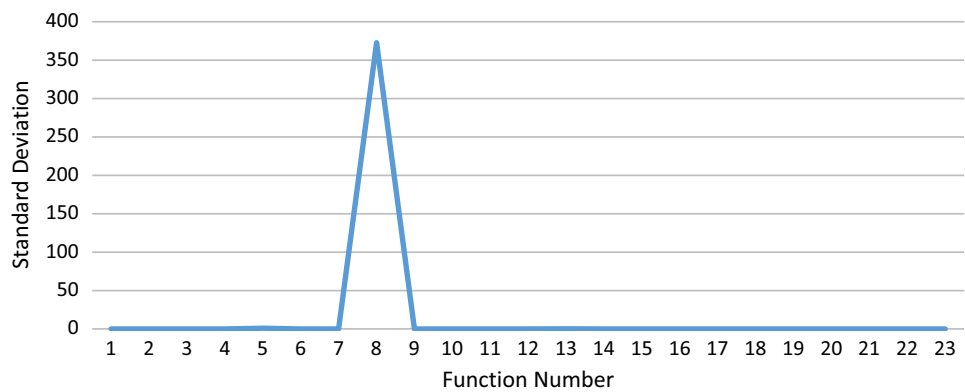
	ASO										GWO										CHPSO																																																																																																																																																																																																																																																																																					
	MEAN	STD	MIN	MAX	MEAN	STD	MIN	MAX	MEAN	STD	MEAN	STD	MIN	MAX	MEAN	STD	MIN	MAX	MEAN	STD	MEAN	STD	MIN	MAX																																																																																																																																																																																																																																																																																		
	F1	215.4122	53.66619	119.4112	239.4125	1.31889E-41	6.07E-42	8.21E-42	2.35E-41	1.3668E-118	3.1E-118	6.6E-131	6.8E-118	F2	52.86905	1.214068	52.32611	55.04084	5.49265E-24	3.78E-24	2.67E-24	1.15E-23	1.7949E-61	3.93E-61	6.52E-67	8.82E-61	F3	22.472.25	795.4003	22,116.54	23,895.1	5.20821E-12	7.38E-12	9.41E-14	1.69E-11	2.5271E-122	5.2E-122	1.1E-132	1.2E-121	F4	44.72397	8.564164	40.89396	60.04401	3.11211E-10	2.24E-10	3.17E-11	6.29E-10	5.45774E-61	1.19E-60	6.45E-64	2.67E-60	F5	8709228	10,7370.8	8517157	8757246	26.56090771	0.469743	26.16251	27.08786	28.88743211	0.032313	28.84023	28.92743	F6	7621.5	1931	6656	10,518	0.201511892	0.112638	2.88E-05	0.253748	0	0	0	0	F7	4.829051	3.817966	3.121604	11.65884	0.000439005	0.000158	0.000243	0.000584	0.000152935	8.03E-05	3.95E-05	0.000235	F8	-4554.23	86.1939	-4592.78	-4400.04	-6339.41806	689.9607	-7253.56	-5403.75	-98,584.7438	43,493.96	-163909	-54,017.8	F9	185.8235	4.291122	178.1473	187.7426	3E-05	5.08E-14	3E-05	3E-05	0	0	0	0	F10	14.78566		14.78566	14.78566					8.88178E-16	N/A	8.88E-16	8.88E-16	F11			0	0	1E-06	0	1E-06	1E-06	0	0	0	0	F12	28.7910.1	13,3496.6	49,104.14	34,7611.6	0.014495975	0.014529	0.006223	0.039943	0.48407984	0.223187	0.140609	0.749628	F13	9,161,900	87,4355.4	7,597,805	9,552,923	0.136306193	0.169501	1.76E-05	0.410966	2.992616362	0.001983	2.989837	2.995025	F14	1.992031	0	1.992031	1.992031	4.141499995	3.800027	0.998004	10.76318	0.998003838	0	0.998004	0.998004	F15	0.00232	0	0.00232	0.00232	0.000307497	9.42E-09	0.000307	0.000308	0.00071298	0.000371	0.000307	0.001034	F16	-1.03163	0	-1.03163	-1.03163	-1.03162845	2.49E-09	-1.03163	-1.03163	-1.03162845	0	-1.03163	-1.03163	F17	0.397887	0	0.397887	0.397887	0.397887	2.2E-07	0.397887	0.397888	0.397887	0	0.397887	0.397887	F18	3	0	3	3	3.000003576	2.14E-06	3	3.000006	3	7.36E-16	3	3	F19	-3.86278	0	-3.86278	-3.86278	-3.86277023	1.49E-05	-3.86278	-3.86275	-3.86278215	0	-3.86278	-3.86278	F20	-3.19328	0	-3.19328	-3.19328	-3.23978182	0.077502	-3.32199	-3.15337	-3.25065921	0.065121	-3.322	-3.2031	F21	-2.63047	0	-2.63047	-2.63047	-9.13265076	2.279367	-10.1524	-5.05519	-10.1531997	1.26E-15	-10.1532	-10.1532	F22	-2.75193	0	-2.75193	-2.75193	-9.34710747	2.35876	-10.4024	-5.12763	-10.4029406	8.88E-16	-10.4029	-10.4029	F23	-2.42734		-2.42734	-2.42734	-10.5354778	0.000274	-10.5357	-10.535	-10.5364098	1.78E-15	-10.5364

Table 4 Comparison of optimization results of the algorithms (SCA, PSO, GWO, MSA) obtained for the unimodal, multimodal and fixed-dimension multimodal benchmark functions

	SCA						PSOGWO						MSA					
	Mean	Std.	Min	Max	Mean	Std.	Min	Max	Mean	Std.	Min	Max	Mean	Std.	Min	Max		
	F1	4.0634E-06	8.65E-08	223.3377	1.881502	0.140658	0.283507	2.85E-13	0.646998	1.6015E-240	0	7.6E-253	8E-240	2.9608E-123	6.3E-123	1.1E-126	1.4E-122	
F2	8.83042E-06	2.67E-07	248.8871	1.913471	4074.131	5739.171	1.743268	13,139.18	1.7398E-194	0	5.5E-199	7.1E-194	1.7398E-194	0	5.5E-199	7.1E-194		
F3	3.95422E-11	1.42E-11	0.506488	0.273875	1.864517	3.382281	1.01E-07	7.799158	9.5029E-115	2.1E-114	2.5E-120	4.7E-114	9.5029E-115	2.1E-114	2.5E-120	4.7E-114		
F4	3.49693E-05	1.46E-06	865.3114	8.817002	1587862	3550511	24.86008	7939208	26.72828569	0.77949	26.04339	27.66449	26.72828569	0.77949	26.04339	27.66449		
F5	4.0634E-06	8.65E-08	223.3377	1.881502	450.6096	1005.543	5.03E-05	2249.377	0.011758024	0.003345	0.006228	0.014613	0.011758024	0.003345	0.006228	0.014613		
F6	8.83042E-06	2.67E-07	248.8871	1.913471	8.711013	19.46754	0.002117	43.5356	0.00011888	3.26E-05	8.84E-05	0.000171	0.00011888	3.26E-05	8.84E-05	0.000171		
F7	3.95422E-11	1.42E-11	0.506488	0.273875	-7604.23	640.2243	-8453.62	-6853.37	-12.306.05978	418.3869	-12.569.5	-11.611.1	-12.306.05978	418.3869	-12.569.5	-11.611.1		
F8	3.49693E-05	1.46E-06	865.3114	8.817002	30.44332	20.07431	7.009196	56.22347	3E-05	0	3E-05	3E-05	3E-05	0	3E-05	3E-05		
F9	4.0634E-06	8.65E-08	223.3377	1.881502	0.397929	6.27E-05	0.397929	0.397929	2.66454E-15	2.51E-15	8.88E-16	4.44E-15	2.66454E-15	2.51E-15	8.88E-16	4.44E-15		
F10	8.83042E-06	2.67E-07	248.8871	1.913471	0.011146	0.020279	1E-06	0.046749	1E-06	0	1E-06	1E-06	1E-06	0	1E-06	1E-06		
F11	3.95422E-11	1.42E-11	0.506488	0.273875	5.425319	9.295735	6.66E-07	21.46578	0.020759754	0.046376	8.73E-06	0.103719	0.020759754	0.046376	8.73E-06	0.103719		
F12	3.49693E-05	1.46E-06	865.3114	8.817002	10049409	22298262	1.98E-05	49937029	0.029766503	0.024627	0.011425	0.057023	0.029766503	0.024627	0.011425	0.057023		
F13	4.0634E-06	8.65E-08	223.3377	1.881502	0.998022	3.94E-05	0.998004	0.998092	0.998003838	1.57E-16	0.998004	0.998004	0.998003838	1.57E-16	0.998004	0.998004		
F14	8.83042E-06	2.67E-07	248.8871	1.913471	0.004645	0.008805	0.000307	0.020363	0.000307523	6.5E-08	0.000307	0.000308	0.000307523	6.5E-08	0.000307	0.000308		
F15	3.95422E-11	1.42E-11	0.506488	0.273875	-1.03163	2.43E-09	-1.03163	-1.03163	-0.8684	0.365	-1.0316	-0.2154	-0.8684	0.365	-1.0316	-0.2154		
F16	3.49693E-05	1.46E-06	865.3114	8.817002	3.000045	9.8E-05	3	3.00022	-3.86278	0	3	3	-3.86278	0	3	3		
F17	4.0634E-06	8.65E-08	223.3377	1.881502	-3.8627	0.000145	-3.86278	-3.86244	-3.86278	4.44E-16	-3.86278	-3.86278	-3.86278	4.44E-16	-3.86278	-3.86278		
F18	8.83042E-06	2.67E-07	248.8871	1.913471	-3.18912	0.120723	-3.322	-3.0861	-3.29227	0.059447	-3.322	-3.2031	-3.29227	0.059447	-3.322	-3.2031		
F19	3.95422E-11	1.42E-11	0.506488	0.273875	-6.56997	3.398006	-10.153	-2.68286	-10.1532	8.88E-16	-10.1532	-10.1532	-10.1532	8.88E-16	-10.1532	-10.1532		
F20	3.49693E-05	1.46E-06	865.3114	8.817002	-8.60287	2.759987	-10.4029	-4.09405	-7.73148	3.658042	-10.4029	-3.7243	-7.73148	3.658042	-10.4029	-3.7243		
F21	4.0634E-06	8.65E-08	223.3377	1.881502	-9.72245	1.082621	-10.5359	-8.39044	-10.5364	3.55E-15	-10.5364	-10.5364	-10.5364	3.55E-15	-10.5364	-10.5364		
F22	8.83042E-06	2.67E-07	248.8871	1.913471	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875		
F23	3.95422E-11	1.42E-11	0.506488	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875	0.273875		

Table 5 Summary of the results of comparing the mean of PSOSCANMS over the benchmark functions

	VS PSO	VS BOA	VS ASO	VS GWO	VS CHPSO	VS SCA	VS PSOGWO	VS MSA
F1	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>
F2	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>W</i>
F3	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>
F4	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>
F5	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>
F6	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>	<i>B</i>
F7	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>
F8	<i>W</i>	<i>W</i>	<i>W</i>	<i>W</i>	<i>W</i>	<i>B</i>	<i>W</i>	<i>W</i>
F9	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
F10	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>
F11	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
F12	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
F13	<i>W</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>W</i>
F14	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>W</i>	<i>B</i>	<i>W</i>
F15	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>
F16	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
F17	<i>W</i>	<i>W</i>	<i>W</i>	<i>W</i>	<i>W</i>	<i>W</i>	<i>W</i>	<i>W</i>
F18	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>
F19	<i>W</i>	<i>W</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>	<i>B</i>
F20	<i>W</i>	<i>W</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>
F21	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>
F22	<i>W</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>	<i>B</i>
F23	<i>B</i>	<i>B</i>	<i>B</i>	<i>B</i>	<i>W</i>	<i>B</i>	<i>B</i>	<i>B</i>

Fig. 3 Plot of the standard deviation vales for all functions: F1–F23

update the particle positions simultaneously at each iteration to new position according to the mathematical equations in SCA and NMS which overcome the problem of local minima.

Table 5 presents a summary of the results. The aggregated results of statistical testing over the functions are shown in Table 5. The symbols (*B*) and (*W*) indicate that a given algorithm performed better (*B*), worse (*W*), in comparison with PSOSCANMS after the compared algorithm had reached the termination criteria. The maximum number of objective function evaluations is 10,000. All results are based on 51 runs.

4.3 Discussion of the Results Presented in Tables 2, 3 and 4 and Summary Table 5

This section is divided into two parts, and the first part discusses the evaluation of exploitation capability, which is presented in Tables 2, 3 and 4 for the functions (F1–F7). However, the second part presents evaluation of exploration capability and the balance between exploration and exploitation for functions (F8–F23). In addition, it explains the summary of mean results presented in Table 5.



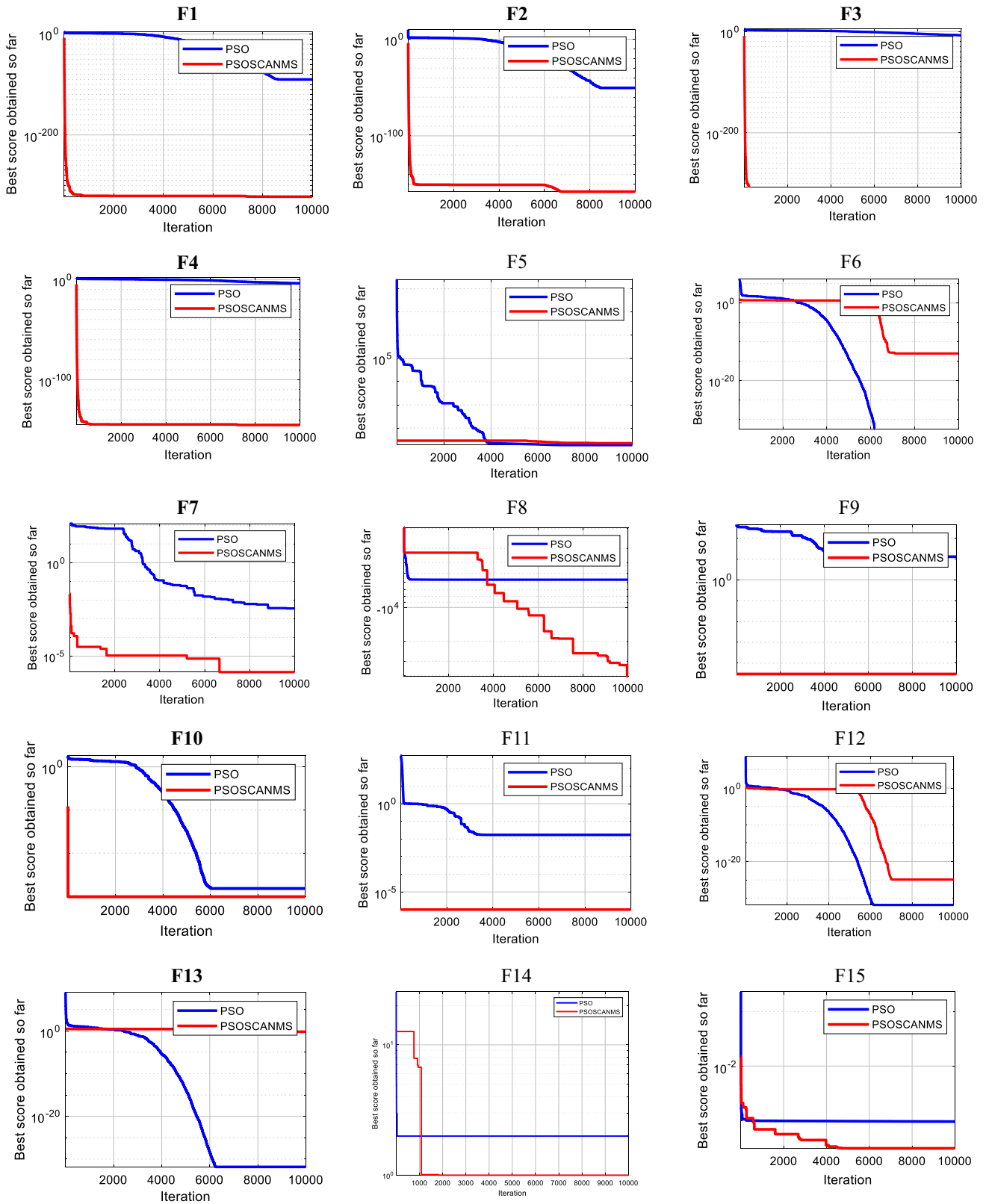


Fig. 4 Convergence of PSOSCANMS algorithm (red color) compared to PSO (blue color)

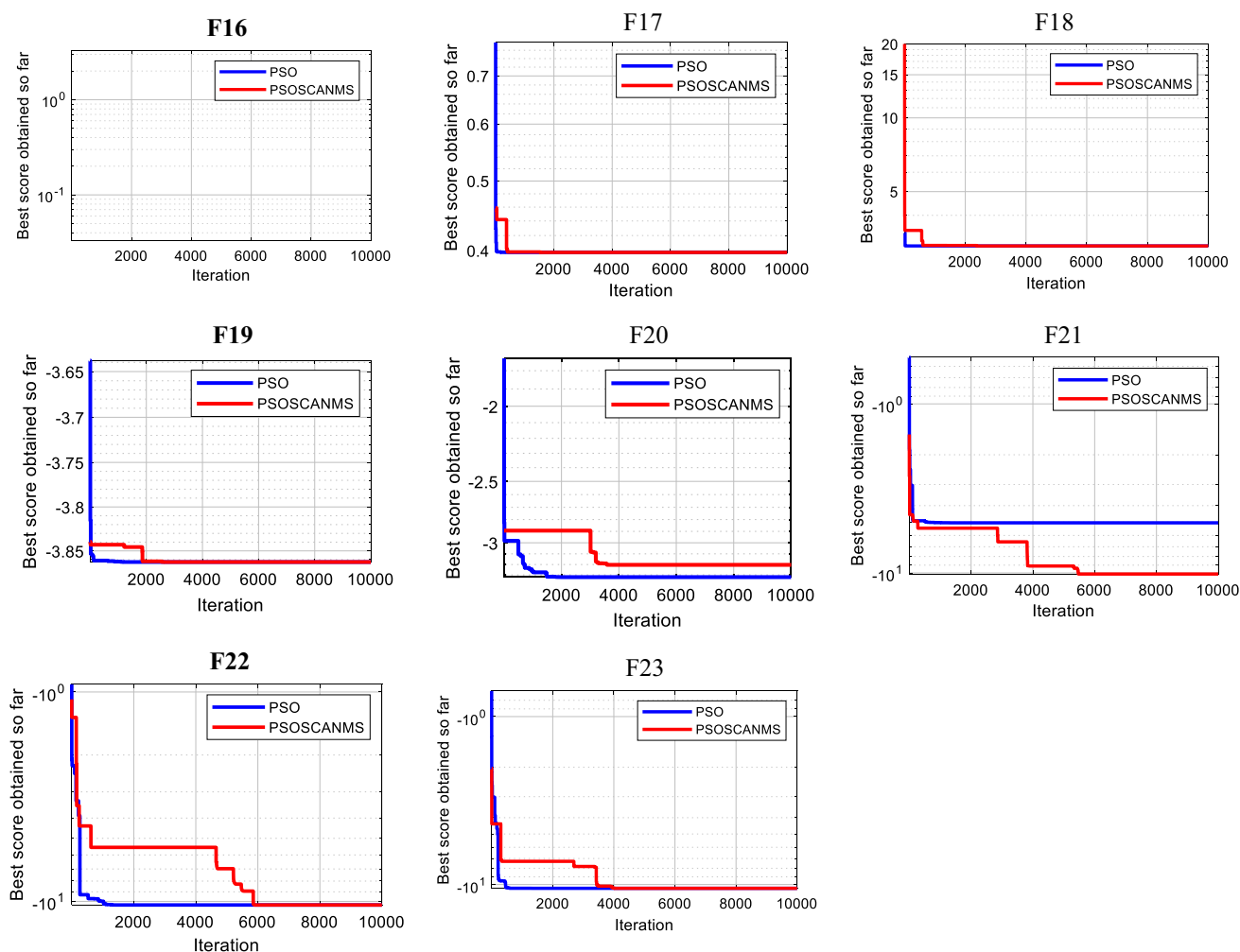


Fig. 4 (continued)

4.3.1 Evaluation of Exploitation Capability (Functions F1–F7)

The results of Tables 2, 3 and 4 and the summary presented in Table 5 show that the proposed algorithm is able to provide very competitive results on the unimodal test functions. These functions do not have a local solution but only one global optimum. However, PSOSCANMS outperformed previous state-of-the-art optimization algorithms (PSO, BOA, ASO and GWO) on all of the seven functions {F1–F7}. Moreover, the compared algorithms have shown the superior results of PSOSCANMS over CHPSO and PSOGWO algorithm, with six functions. The reason for these results is that hybridization between the three algorithms allowed the function to reach the optimum global point. Since all of the seven unimodal testing functions have the best results with the PSOSCANMS algorithm, it proves that it has strong exploitation behavior which is important for numerous issues of optimization that need to be solved. As already mentioned,

unimodal functions are appropriate for benchmarking algorithms. The results therefore show that the PSOSCANMS algorithm is highly operational. Further, the superior of the results over both original PSO algorithm and SCA algorithm proves that the hybridization is very successful and very beneficial for solving the optimization problems with single objective point and in reaching the global optimum.

4.3.2 Evaluation of Exploration Capability and the Balance Between Exploration and Exploitation (Functions F8–F23)

Tested benchmark functions {F8–F13} represent multimodal functions that include many local optima points whose number increases exponentially with the problem size. To evaluate the exploration capacity of optimization algorithms, multimodal functions are used. As shown in Tables 2, 3 and 4 and the summary in Table 5, the PSOSCANMS algorithm highly outperforms (PSO, BOA, ASO, GWO, CHPSO, SCA,

Table 6 Overall confusion matrix of PSOSCANMS using different dimensions over functions (F1)

Dimension	Number of particles	Training accuracy (%)	Testing accuracy (%)	Overall accuracy (%)
10	100	95.5	94.2	94.85
30	100	94.7	93.1	93.9
50	100	88.8	83.7	86.25
100	100	82.9	79.3	81.1

Table 7 Overall confusion matrix of PSOSCANMS using different dimensions over functions (F3)

Dimension	Number of particles	Training accuracy (%)	Testing accuracy (%)	Overall accuracy (%)
10	100	96.2	96.7	96.45
30	100	95.2	94.1	94.65
50	100	91.0	90.4	90.7
100	100	83.8	80.9	82.35

Table 8 Overall confusion matrix of PSOSCANMS using different dimensions over functions (F6)

Dimension	Number of particles	Training accuracy (%)	Testing accuracy (%)	Overall accuracy (%)
10	100	93.5	92.8	93.15
30	100	94.6	90.3	92.45
50	100	87.5	86.1	86.8
100	100	80.2	77.3	78.75

Table 9 Overall confusion matrix of PSOSCANMS using different dimensions over functions (F9)

Dimension	Number of particles	Training accuracy (%)	Testing accuracy (%)	Overall accuracy (%)
10	100	91.7	89.4	90.55
30	100	89.4	87.8	88.6
50	100	82.2	79.1	80.65
100	100	77.5	77.2	77.35

PSOGWO and MSA) over at least eleven functions over the tested multimodal functions {F8–F13}. This proves that PSOSCANMS can well balance exploration and exploitation phases. This may be due to the fact that PSOSCANMS provides more exploring points in the search space for the reason of using the two equations of SCA algorithm and the three equations of NMS algorithm. It generates new positions for each particle at each iteration and updates the current particle position to the best position in the different dimensions that

guide PSOSCANMS algorithm to reach the global optimum point in the search space. However, PSOSCANMS outperforms PSOGWO over fifteen functions and ASO over fourteen tested functions indicating that PSOSCANMS has also a very good exploration capability. In fact, the present algorithm always appeared the most efficient or the second best algorithm in the majority of test problems. Functions F8 to F23 are divided into two sections, the first is unimodal functions and the second is fixed unimodal functions. However, Tables 2, 3 and 4 show that PSOSCANMS performance over the tested fixed-dimension functions {F14–F23} is better than the compared algorithms. PSOSCANMS outperforms (ASO, GWO and PSOGWO) over nine functions out of ten and outperforms (PSO, BOA, SCA and MSA) in six functions.

These results show that PSOSCANMS can perform good at real-world problems which may not have uniform functions. Further it can be seen that in the fixed-dimension functions {F14–F23}, PSOSCANMS performs better than PSO. The multimodal fixed-dimensional functions have very difficult search spaces and high exploration requirements for the accurate approximation of their global optimal. The results show that the PSOSCANMS algorithm has an excellent scan capability which leads to global optimization. Moreover, high local optimum avoidance of this algorithm is from which these results can be deduced. In addition to multimodal functionality, the algorithm can further explore the search space and find promising search space areas, improving the number of local solutions.

4.4 Stability of PSOSCANMS Algorithm

The plot of the standard deviation values for all functions from F1 to F23 is shown in Fig. 3. Except for F8, all other functions are near zero and this means that POLARPSO is stable over the 51 experiments that were performed, the reason for the high standard deviation value of F8 is due to the very large values in the search space.

4.5 Convergence Curves Analysis and Discussion

The convergence curves of the tested benchmark functions of PSOSCANMS in comparison with PSO are shown in Fig. 4. Please note that the average best solution for every iteration indicates the average best solution obtained up to now in every iteration.

It can be seen in Fig. 4 that the convergence behaviors of PSOSCANMS fall into different categories: First, it tends to be accelerated as iteration increases; second it finds the optimum value at the later stages of the iterations. However, according to Van Den Berg et al. [38], this behavior can guarantee that a population-based algorithm eventually converges to a point and searches locally in a search space. This is an evidence that the proposed hybrid algorithm can successfully

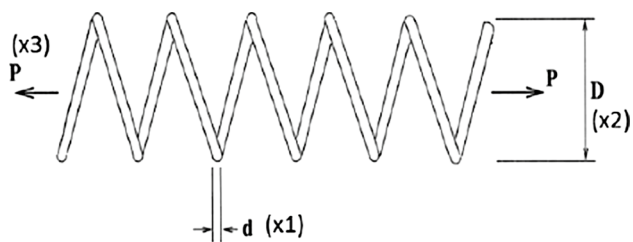


Fig. 5 Schematic of tension/compression spring design problem

improve the fitness of all particles and guarantee finding better solutions as iteration increases. This can be discussed and reasoned according to the hybridization of the proposed algorithm. Since the particles of the swarm tend to move from a high value to global minimum value, similarly to the evolution concepts in PSO, the whole particles and their fitness average tend to be improved over the course of iterations. In addition, we save the best value and move it to the next iteration, so the best value formed so far is always available to improve the fitness of other particles. It can be also observed that when comparing the behavior of PSOSCANMS with PSO, in most of the functions, PSOSCANMS has a higher success rate in finding the optimum global value than PSO and it is faster to reach the global value as it can be seen in {F1, F2, F3, F4 F5, F7, F9, F10, F11, F14, F15, F17, F18, F19, F21}.

The first behavior of the acceleration as iteration increases is due to the increasing number of search points resulted from the hybridization between the three algorithms, which help the particles in the initial stages of the iteration to search for new regions in the search space that speed up finding the optimal values with a minimal number of iterations. This performance of the algorithm can be seen in all of the unimodal functions as well as in most of the multimodal functions {F1, F2, F3, F4 F5, F7, F9, F10, F11, F14, F15, F17, F18, F19, F21}. The second behavior of PSOSCANMS convergence curve is to find the optimum value at the later stages of the iterations.

This behavior can be seen in functions {F6, F8, F12, F15, F20, F21, F22}. The explanation of this is probably because PSOSCANMS exploits the solutions, and it has been found so far before exploring new search regions; it benefits from both exploration and exploitation features that are improved in PSOSCANMS, which helps this algorithm find the global optimum. It shows that PSOSCANMS is able to escape local minimum and find the global optimum in the search space.

4.6 Confusion Matrix

The overall confusion matrix of PSOSCANMS using different dimensions over four selected functions (F1, F3, F6, F9) has been calculated since these functions dimension is flexible and not fixed. The results of applying the algorithm over (10, 30, 50) dimension are provided in Table 5. It is known in the optimization field that as the dimension increases the problem will be harder and the accuracy of reaching the optimal point will be lower; however, performance was good at high dimension as illustrated in the confusion matrix (Tables 6, 7, 8 and 9). The confusion matrix has been guided by the method used by [39, 40].

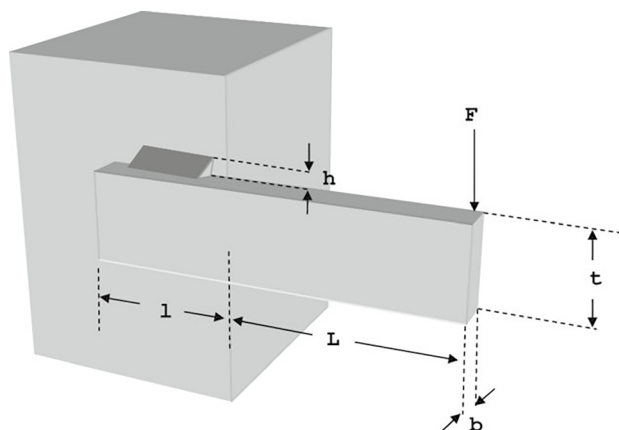


Fig. 6 Welded beam design

Table 10 Comparison of results for tension/compression spring design problem

Algorithm	Optimum variables			Optimum
	Wire diameter	Mean coil diameter	Number of active coils	
PSOSCANMS	0.05072	0.334801	10.79431	0.012475
GWO	0.05169	0.356737	11.28885	0.012666
GSA	0.050276	0.323680	13.525410	0.0127022
PSO (Ha and Wang)	0.051728	0.357644	11.244543	0.0126747
ES (Coello and Montes)	0.051989	0.363965	10.890522	0.0126810
GA (Coello)	0.051480	0.351661	11.632201	0.0127048
HS (Mahdavi et al.)	0.051154	0.349871	12.076432	0.0126706
DE (Huang et al.)	0.051609	0.354714	11.410831	0.0126702
Mathematical optimization (Belegundu)	0.053396	0.399180	9.1854000	0.0127303
Constraint correction (Arora)	0.050000	0.315900	14.250000	0.0128334

Table 11 Comparison results of the welded beam design problem

Algorithm	Optimum variables				Optimum
	<i>h</i>	<i>l</i>	<i>t</i>	<i>b</i>	
PSOSCANMS	0.167953	3.279103	6.17342	0.117935	1.34960
GWO	0.205676	3.478377	9.03681	0.205778	1.72624
GSA	0.182129	3.856979	10.00000	0.202376	1.879952
GA (Coello)	N/A	N/A	N/A	N/A	1.8245
GA (Deb)	N/A	N/A	N/A	N/A	2.3800
GA (Deb)	0.2489	6.1730	8.1789	0.2533	2.4331
HS (Lee and Geem)	0.2442	6.2231	8.2915	0.2443	2.3807
Random	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex	0.2792	5.6256	7.7512	0.2796	2.5307
David	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815

5 PSOSCANMS for Solving Problems in Engineering Design

In this section, a traditional engineering design problem such as spring compression, welded beam is also considered. The PSOSCANMS should be fitted with a limiting technique to optimize restricted issues. These issues are restricted in equal and in inequality measure. In general, limit handling is very difficult when the fitness function directly impacts the search agent’s position update. However, any kind of restrictive constraint can be used for fitness algorithms without changing the algorithm system. While the PSOSCANMS algorithm searches, agents update their positions regarding alpha, beta and delta places. Between the search agents and fitness function, there is no immediate relationship. For example, the simplest limit handling method may be used to effectively handle restrictions of PSOSCANMS when search agents are given high objective function values if they break one of the restrictions. If the alpha, beta or delta are restricted, then the next iteration will automatically replace them with a new search agent. Any kind of feature can be used to penalize search agents on the basis of the rate of infringement. If the penalty is lower than any other, the alpha, beta or delta are substituted automatically in the next iteration by a fresh search agent [41].

5.1 Tension/Compression Spring Design

This problem is aimed at reducing the voltage/pressure source weight as shown in Fig. 4. The minimization method is subject to certain restrictions, including shear stress, floating frequency and minimum floating deflections. Both mathematical and heuristic methods have addressed this problem. A constantly limited issue lays in the tension/compression spring design (TCSD) issue as shown in Fig. 4. The issue is that a coil spring volume *V* is kept below a steady voltage

compression load. Three design factors are as follows: number of spring coils that is active as shown in mathematical Eq. 10, winding diameter as shown in Eq. 11, wire diameter as shown in Eq. 12 [42].

$$P = x_1 \in [2, 15] \tag{10}$$

$$D = x_2 \in [0.25, 1.3] \tag{11}$$

$$d = x_3 \in [0.05, 2] \tag{12}$$

The tension/compression spring design problem mathematically describes as the following (Fig. 5).

$$\text{MIN } F(X) = (X_2 + 2)X_2X_1^2 \tag{13}$$

minimize subject to:

$$q_2(x) = (4X_2^2 - X_1X_2/12566(X_2X_1^3 - X_1^4)) + (1/5108X_1^2) - 1 \leq 0$$

$$q_1(x) = 1 - (X_2^3X_3/71785X_1^4) \leq 0$$

$$q_3(x) = 1 - 140.45X_1 \leq 0$$

$$q_4(x) = (X_2+X_1/1.5) \leq 0$$

The upper and lower limit variables are design problems: $2 \leq X_1 \leq 15, 0.25 \leq X_2 \leq 1.3, 0.05 \leq X_3 \leq 2$.

This is a convex optimization problem, and the closed-form optimum solution of the problem is $f(X) = 0.0126652327883$ for $X = [x_1, x_2, x_3] = [0.051689156131, 0.356720026419, 11.288831695483]$ [6].

The numerical optimization (continuous constraints) and mathematical optimization technique are the mathematical methods used to fix tension/compression spring design problem. The results are compared to PSOSCANMS in Table 10. It should be noted that PSOSCANMS uses the same penalty feature to make a fair comparison. It indicates that PSOSCANMS finds a design for this issue with the minimum weight.

Experimental results showed that PSOSCANMS is more successful than PSO and outperforms the other state-of-the-art compared algorithms over the tested optimization problems [6].

5.2 Welded Beam Design

This issue is aimed at reducing the price of manufacturing a welded beam, as shown in Fig. 6. The figure shows a rigid component which is welded on a beam. At the end of the member, a load is applied. The complete manufacturing price corresponds to the labor costs (depending on the size of the sod) plus the price for the soldering and beams. By changing weld and member dimensions x_1 , x_2 , x_3 and x_4 , the beam is to be optimized for minimal costs. Limits to shear stress, bending stress, hinge load and end deflection include restrictions. The x_1 and x_2 variables are generally 0.0625-inch integer, but are presumed to be constant for this implementation [43].

The following restrictions apply: shear stress (s), beam bending stress (h), bar bending load (P_c), beam deflection end (d) and side constraint. This problem has four variables such as thickness of weld (h), length of attached part of bar (l), the height of the bar (t) and thickness of the bar (b). The mathematical formulation is as follows [44].

Consider minimize subject to [6]:

$$\begin{aligned} \dot{X} &= [X_1 X_2 X_3 X_4] = [hltb], \\ (f\dot{X}) &= 1.10471 X_1^2 X_2 + 0.04811 X_3 X_4 (14.0 + X_2) \\ g_1(\dot{X}) &= T(\dot{X}) - T_{\max} \leq 0, \\ g_4(\dot{X}) &= X_1 - X_4 \leq 0 \\ g_2(\dot{X}) &= \sigma(\dot{X}) - \sigma_{\max} \leq 0, \\ g_5(\dot{X}) &= P - P_c(\dot{X}) \leq 0 \\ g_6(\dot{X}) &= 0.125 - X_1 \leq 0 \\ g_3(\dot{X}) &= \sigma(\dot{X}) - \sigma_{\max} \leq 0, \\ g_7(\dot{X}) &= 1.10471 X_1^2 + 0.04811 X_3 X_4 (14.0 + X_2) - 5.0 \leq 0 \end{aligned} \quad (14)$$

variable ranges: $0.1 \leq X_1 \leq 2$, $0.1 \leq X_2 \leq 2$, $0.1 \leq X_3 \leq 10$, $0.1 \leq X_4 \leq 2$.

where

$$\begin{aligned} T(\dot{X}) &= \sqrt{((T')^2 + 2T'T''(X_2/2R) + (T'')^2)} \\ T' &= (p/(2x_1 * x_2)), \quad T'' = MR/J, M = P(L + X_2/2) \\ R &= \sqrt{((x_2^2/4) + (x_1 + x_3/2))} \\ J &= 2\{\sqrt{(2x_1 x_2)}[(x_2^2/4) + ((x_1 + x_2)/2)^2]\} \\ \sigma(\dot{X}) &= (6PL/X_4 X_3^2), \quad \sigma(\dot{X}) = (6PL^3/EX_3^2 X_4) \\ P_c(\dot{X}) &= (4.013E\sqrt{(x_3^2 x_4^6)})/L^2(1 - ((x_3/21)\sqrt{(E/4G)}) \end{aligned} \quad (15)$$

The results of the comparison are shown in Table 11. It indicates that PSOSCANMS over the welded beam design

problem finds a design whose costs are minima when comparing it with other designs algorithms.

6 Conclusion

In this paper, a novel hybrid evolutionary algorithm that combines particle swarm optimization (PSO) with sine–cosine optimizer and Nelder–Mead simplex optimization technique has been proposed in order to solve the drawbacks in PSO algorithm in locating local minima rather than global minima, low converge rate and low balance between exploration and exploitation. The proposed algorithm is called PSOSCANMS. The solutions are updated by the proposed algorithm with the best position found by the three cascaded hybrid algorithms. The range of the search points has increased due to using the mathematical equations of SCA and NMS. These equations allow particles to continuously update their position and reassign the solutions that they already have found. Further, they enhance PSO to explore new areas and dimensions in the search space. The proposed algorithm is tested on 23 benchmark functions in order to investigate its performance on the different scale functions. The results showed that PSOSCANMS has high rate in finding the optimum global value and high exploration feature over the original PSO and the other compared algorithms. However, this is due to the fact that cascading the three hybrid algorithms allows particles to examine every possible solution from the initial iteration to the end of the optimization procedure in the different dimensions. PSOSCANMS has demonstrated fast behavior, which simultaneously reaches the optimal global point and a high local optimum avoidance speed. Experimental results of comparison PSOSCANMS showed that PSOSCANMS has outperformed the other state-of-the-art compared algorithms over the compared algorithms and when applying to solve engineering design problems such as tension/compression spring design and welded beam design problems.

References

1. Krawiec, K., Simons, C., Swan, J., Woodward, J.: Metaheuristic design patterns: new perspectives for larger-scale search architectures. In: Vasant, P., Alparslan-Gok, S.Z., Weber, G. (eds.) Handbook of Research on Emergent Applications of Optimization Algorithms, pp. 1–36. IGI Global, Pennsylvania (2018)
2. Ong, P.; Chin, D.D.V.S.; Ho, C.S.; Ng, C.H.: Metaheuristic approaches for extrusion manufacturing process: utilization of flower pollination algorithm and particle swarm optimization. In: Handbook of Research on Applied Optimization Methodologies in Manufacturing Systems, pp. 43–56. IGI Global, Pennsylvania (2018)



3. Hudaib, A.A.; Fakhouri, H.N.: Supernova optimizer: a novel natural inspired meta-heuristic. *Mod. Appl. Sci.* **12**(1), 32 (2017)
4. Mendes, R.; Kennedy, J.; Neves, J.: The fully informed particle swarm: Simpler, mabe better. *IEEE Trans. Evol. Comput.* **8**, 204–210. (2004). <https://doi.org/10.1109/TEVC.2004.826074>
5. Kennedy, J.: Particle swarm optimization. *Encyclopedia of machine learning*, pp. 760–766. Springer, US (2011)
6. Al-Sayyed, R.M.; Fakhouri, H.N.; Rodan, A.; Pattinson, C.: Polar particle swarm algorithm for solving cloud data migration optimization problem. *Mod. Appl. Sci.* **11**(8), 98 (2017)
7. Altay, E.V.; Alatas, B.: Performance comparisons of socially inspired metaheuristic algorithms on unconstrained global optimization. In *Advances in Computer Communication and Computational Sciences*, pp. 163–175. Springer, Singapore (2019)
8. Chegini, S.N.; Bagheri, A.; Najafi, F.: PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems. *Appl. Soft Comput.* **73**, 697–726 (2018)
9. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **96**, 120–133 (2016)
10. Eberhart, R.; Kennedy, J.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948 (1995)
11. Benítez-Hidalgo, A.; Nebro, A.J.; Durillo, J.J.; García-Nieto, J.; López-Camacho, E.; Barba-González, C.; Aldana-Montes, J.F.: About designing an observer pattern-based architecture for a multi-objective metaheuristic optimization framework. In: *International Symposium on Intelligent and Distributed Computing*, pp. 50–60. Springer, Cham (2018)
12. Li, Y.G.; Gui, W.H.; Yang, C.H.; Li, J.: Improved PSO algorithm and its application. *J. Central South Univ. Technol.* **12**(1), 222–226 (2005)
13. Pham, D.T.; Ghanbarzadeh, A.; Koç, E.; Otri, S.; Rahim, S.; Zaidi, M.: The bee's algorithm—a novel tool for complex optimization problems. In: *Intelligent Production Machines and Systems*, pp. 454–459. Elsevier Science Ltd, Amsterdam (2006)
14. Spendley, W. G. R. F. R.; Hext, G. R.; Himsforth, F. R.: Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, **4**(4), 441–461 (1962)
15. Nelder, J.A.; Mead, R.: A simplex method for function minimization. *Comput. J.* **7**(4), 308–313 (1965)
16. Wright, M.H.: Nelder, Mead, and the other simplex method. *Doc. Math.* **7**, 271–276 (2010)
17. Sörensen, K.; Sevaux, M.; Glover, F.: A history of metaheuristics. In: *Handbook of Heuristics*, pp. 1–18 (2018)
18. Beni, G.; Wang, J.: Swarm intelligence in cellular robotic systems. In: *Proceedings of NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, Italy, June 26–30 (1989). https://doi.org/10.1007/978-3-642-58069-7_38
19. Dorigo, M.; Di Caro, G.: Ant colony optimization: a new metaheuristic. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406), Vol. 2, pp. 1470–1477. IEEE, Washington (1999)
20. Yao, X.; Liu, Y.; Lin, G.: Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **3**(2), 82–102 (1999)
21. Yang, X.S.; Deb, S.: Cuckoo search via Lévy flights. In: *2009 World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pp. 210–214. IEEE, Washington (2009)
22. Karaboga, D.; Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Global Optim.* **39**(3), 459–471 (2007)
23. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci.* **179**(13), 2232–2248 (2009)
24. Yang, X.S.: Firefly algorithm. In: *Engineering Optimization*, pp. 221–223 (2010)
25. Mirjalili, S.; Mirjalili, S.M.; Lewis, A.: Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61 (2014)
26. Kaveh, A.; Farhoudi, N.: A new optimization method: Dolphin echolocation. *Adv. Eng. Softw.* **59**, 53–70 (2013)
27. Pan, W.T.: A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowl. Based Syst.* **26**, 69–74 (2012)
28. Krishnanand, K.N.; Ghose, D.: Glowworm swarm optimization: a new method for optimising multi-modal functions. *Int. J. Comput. Intell. Stud.* **1**(1), 93–119 (2009)
29. Kiran, M.S.: TSA: tree-seed algorithm for continuous optimization. *Expert Syst. Appl.* **42**(19), 6686–6698 (2015)
30. Rajabioun, R.: Cuckoo optimization algorithm. *Appl. Soft Comput.* **11**(8), 5508–5518 (2011)
31. Oftadeh, R.; Mahjoob, M.J.; Shariatpanahi, M.: A novel metaheuristic optimization algorithm inspired by group hunting of animals: hunting search. *Comput. Math Appl.* **60**(7), 2087–2098 (2010)
32. Zhao, W.; Wang, L.; Zhang, Z.: Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl. Based Syst.* **163**, 283–304 (2019)
33. Joshi, H.; Arora, S.: Enhanced grey wolf optimization algorithm for global optimization. *Fundam. Inf.* **153**(3), 235–264 (2017)
34. Qais, M.H.; Hasanien, H.M.; Alghuwainem, S.: Augmented grey wolf optimizer for grid-connected PMSG-based wind energy conversion systems. *Appl. Soft Comput.* **69**, 504–515 (2018)
35. Fakhouri, S.N., Hudaib, A., Fakhouri, H.N.: Enhanced optimizer algorithm and its application to software testing. *J. Exp. Theor. Artif. Intell.* (2019). <https://doi.org/10.1080/0952813X.2019.1694591>
36. Mohamed, A.A.A.; Mohamed, Y.S.; El-Gaafary, A.A.; Hemeida, A.M.: Optimal power flow using moth swarm algorithm. *Electr. Power Syst. Res.* **142**, 190–206 (2017)
37. Arora, S.; Singh, S.: Butterfly optimization algorithm: a novel approach for global optimization. *Soft. Comput.* **23**, 715–734 (2018)
38. Van Den Berg, R. A.; Pogromsky, A. Y.; Leonov, G. A.; Rooda, J. E.: Design of convergent switched systems. In *Pettersen K.Y., Gravdahl J.T., Nijmeijer H. (eds.) Group coordination and cooperative control* (pp. 291–311). Springer, Berlin, Heidelberg (2006)
39. Semwal, V.B.; Singha, J.; Sharma, P.K.; Chauhan, A.; Behera, B.: An optimized feature selection technique based on incremental feature analysis for bio-metric gait data classification. *Multimed. Tools Appl.* **76**(22), 24457–24475 (2017)
40. Semwal, V.B., Gaud, N., Nandi, G.C.: Human gait state prediction using cellular automata and classification using ELM. In: *Tanveer, M., Pachori, R. (eds.) Machine Intelligence and Signal Analysis*, pp. 135–145. Springer, Singapore (2019)
41. Kumar, S.; Aaron, J.; Sokolov, K.: Directional conjugation of antibodies to nanoparticles for synthesis of multiplexed optical contrast agents with both delivery and targeting moieties. *Nat. Protoc.* **3**(2), 314 (2008)
42. Valsange, P.S.: Design of helical coil compression spring: a review. *Int. J. Eng. Res. Appl.* **2**(6), 513–522 (2012)
43. Deb, K.: Optimal design of a welded beam via genetic algorithms. *AIAA J.* **29**(11), 2013–2015 (1991)
44. Azqand, M.S., Delavar, M., Arjmand, M.: An enhanced time evolutionary optimization for solving engineering design problems. *Eng. Comput.* (2019). <https://doi.org/10.1007/s00366-019-00729-w>

