A COMPREHENSIVE SURVEY ON PARTICLE SWARM OPTIMIZATION (PSO) AND ITS VARIANTS TO CLASSICAL BENCHMARK FUNCTIONS

[1]Rajapandian Poongavanam, [2]Vengattaraman Thirumal, [3] Kalaipriyan Thirugnanasambandam
[1]Research Scholar, [2]Assistant Professor, [3] Assistant Professor
[1]Department of Computer Science, Pondicherry University, Puducherry - 605014, India
[2]Department of Computer Science, Pondicherry University, Puducherry - 605014, India
[3] Department of Computer Science, Koneru Lakshmaiah Education Foundation, Guntur, AP, India

Abstract : Particle Swarm Optimization (PSO) is one of the most popular population-based swarm intelligent algorithms inspired by the behaviours of birds flocking and fish schooling. This algorithm holds few tuning parameters to obtain the optimum solution; however, PSO has maximum possibility to stagnate in local optima due to its simplicity and flexibility. Several modified PSO versions are developed to maintain the diversity of the particles and to avoid the local optima struck. In this paper, Generic PSO and a novel collection of eight different PSO variants are presented. The collected PSO variants have successful balancing capabilities between the local search and global search of optimum solutions. This survey offers good understanding on generic particle swarm optimization and its distinct variants. The particles in each algorithm have unique characteristics on converging towards best optimum solutions. The experimentation results of the collected PSO variants are processed using MATLAB (version 2013 a). The performance of distinct collection of PSO variants are computed using three different groups of benchmark test function viz., unimodal, multimodal and fixed dimension multimodal functions. The performance of each algorithm is discussed with the experimentation results.

***IndexTerms* -** Particle Swarm Optimization (PSO); Swarm Intelligent; unimodal; multimodal; fixed dimension multimodal function.

I. INTRODUCTION

From last two decades, Meta-heuristic algorithms have become more and more popular in solving engineering problems in different fields. Further, this algorithm is classified into two categories namely single solution based and population based algorithms. In single solution based algorithms, the search process begins with single solution and improves its solution in a certain number of iterations. Whereas, population based meta-heuristic starts with a set of solutions and updates its solutions in a course of iterations. Population based meta-heuristic have few pros compared to single solution based algorithms: i) Set of solutions convey information about search space which aids solution to jump toward the promising part of search space, ii) it guides each other to avoid local optimal solutions, iii) provides better diversification than single solution based algorithms (Seyedali Mirjalili, et al. 2014). Swarm Intelligence (SI) is one of the interesting subdivision of the population based meta-heuristic algorithms. (Beni, et al. 1993) introduced these SI concepts in cellular robotic systems in 1993. Swarm intelligent algorithms (bonabeau, et al. 1999) are inspired from natural colonies, herds, flock and schools. Thus, many Swarm intelligent algorithms such as Particle swarm optimization (PSO) (Kennedy, Eberhart, 1995), Ant colony optimization (ACO) (Dorigo, et al. 2006), Artificial Bee Colony Algorithm (ABC) (Basturk, Karaboga, 2006) and etc have emerged.

Particle Swarm Optimization is the best performing algorithm in Swarm Intelligent. This algorithm is inspired by the behavior of birds flocking and fish schooling (Kennedy, et al. 1999,). Few tuning parameters are incorporated to identify the optimum solutions which makes algorithm very simple and flexible. This feature attracts more researchers to apply PSO in different scientific and real-world problems (M. Clerc and J. Kennedy, 2002; R. Mendes et al., 2004). However, PSO algorithm suffers from two major problems of trapping in local optima and premature convergence and decreased convergence rate in later course of evolution. To overcome those issues, several variants of PSO algorithms were developed in the literature. Shi and Eberhart developed an inertia weight with time varying parameter into original PSO algorithm. This algorithm proved that modified PSO provides global optimum solution by using divergence capability (Y. Shi and R. Eberhart, 2002). Zhan et al. proposed orthogonal learning strategy to guide the particles to move in better search space based on best particles experience and neighborhood experience (Zhan et al., 2011).

In (Wang et al., 2003) used special velocity update mechanism to prevent premature convergence problems. Haiping yu et al proposed Elite grouped Adaptive PSO (EGAPSO) to differentiate the elite particles and worst particles, later the worst particles are replaced stochastically to maintain the swarm diversity (Haiping yu et al., 2013). Author (Suganthan et al., 2012) proposed a niching particle swarm optimizer with local search technique to identify local optimum solutions within the local basin to premature convergence. (Swagatam et al., 2013) presented distance-based locally Informed particle swarm (LIPS) optimizer to improve the global best solution (gbest) by distance measurement and also improves the local search capability (Swagatam et al., 2013). The author (Tsoulos et al.,2010) introduced modified velocity update procedure together with stopping rule, relationship check and some exploitation methods.

In (Mendes et al., 2004) presented shared information of the entire neighborhood to assist the particle to move in better directions. Particles neighborhood determined using topological structure in (Yue-jiao et al., 2013). Liang et al. proposed a new comprehensive learning method to adjust velocity of particles by taking into account of all best particles and as a result exploration of particles guides to overcome the premature convergence (Liang et al., 2004) and (Ran Chenga et al., 2015) proposed new social learning to update the position of the particles based on their historical information. In (Chun-Feng Wang et al., 2016) was introduced three different improvement strategy to avoid local optima and premature convergence. Zahra Beheshti et al., a new PSO algorithm with fusion global-local-topology (FGLT-PSO) were introduced to overcome the premature convergence and neighboring optimum solutions (Zahra Beheshti et al., 2014).

Many of researchers contributed to improve the inertia weight in PSO; however this operator helps to balance the exploration and exploitation but fails in achieving local optimum solutions and premature convergence. Natural exponential (G. Chen et al., 2006), Sigmoid (R.F. Malik., 2007), Chaotic (Y. Feng et al., 2008), Logarithm Decreasing (Y. Gao et al., 2008), linearly decreasing (J. Xin et al., 2009), Oscillating (K. Kentzoglanakis and M. Poole, 2009), Exponent decreasing (H.R. Li and Y.L. Gao, 2009) are some of the inertia weight parameters integrated with generic PSO. Other variants of PSO are as Cellular PSO (Yang Shi et al., 2011), Example-based learning PSO (Han Huang et al., 2012), accurate sub-swarm PSO (Jianxin Liao et al., 2014), Enhanced Leader PSO (A. RezaeeJordehi, 2015) and so on.

Conventional PSO improved through hybridization of various meta-heuristic algorithms in that some of them are as Genetic algorithm (Boudjelaba and Chikouche, 2014; Harish Garg, 2016), Gravitational search (GSA) (Seyedali and Siti Zaiton, 2010; Shanhe Jiang et al., 2014), bacterial foraging algorithm (BFA) (S.M. Abd-Elazim and E.S. Ali, 2013; Padmavathi Kora and Sri Ramakrishna Kalva, 2015), Imperialistic completive algorithm (ICA) (Lhassane Idoumghar et al., 2013), Swallow Swarm Optimization algorithm (SSO) (A. Kaveh, 2014) and so on incorporated with PSO to improve the generic PSO and to apply in different domain problems. In our survey, we collected the PSO variants in which exploration and exploitation is considered as the main part to improve the algorithmic nature.

The rest of the paper is organized as follows. In section 2 generic PSO algorithms is presented and in section 3 PSO variants are explained. The conducted experiments and results are discussed in section 4. Finally, the paper concluded with summary in section 5.

II. GENERIC PARTICLE SWARM OPTIMIZATION

Most of the problems such as non-linear combinatorial problems are mainly solve through the various evolutionary algorithms. In this evolutionary algorithm, Particle swarm optimization is a major role for solving the problems to identify the optimal solutions. PSO is a population-based search algorithm inspired by the flocking of birds foraging for the food, which is first introduced by James Kennedy and Russell Eberhart in 1995 (Kennedy J and Eberhart R, 1995). Compared to other evolutionary algorithms such as genetic algorithm, Memetic algorithm, PSO is not incorporated with any genetic operators to solve the problems. Each and every solution in this algorithm is termed as 'Particle', collection of these particles are 'swarm'. PSO works with their own tactics that is, each individual particle adjust their positions based on their surrounding best positions and their previous historic best (J. Kennedy, 1999; J. Kennedy, 2002).

In PSO, velocity update and particle position update are the two equations are used. In PSO, Velocity update equations are classified into three major parts such as (M. Clerc and J. Kennedy, 2002; R. Mendes et al., 2004)

a) The first part termed as Momentum, where the outcome of new velocity obtained from the previous velocity of the particle.
b) The second part termed as Cognitive part, where the particles learn themselves by their previous best position.
c) The last part is cognitive part; here the particles are collaborating with each other by the best particle in their surroundings.

$$V_{id}^{t+1} = \omega_t * V_{id}^t + C_1 * r_1 * (p_{id}^t - X_{id}^t) + C_2 * r_2 * (p_{gd} - X_{id}^t) \quad \dots (1)$$

Where C_1 and C_2 are the random learning factors for the adjustment of cognitive and social parameters ω_t represents as the inertia weight for slow changes in the velocity for t iterations, r_2, r_2 are the random numbers between (0,1), V_{id}^t Determines that the velocity of each particles 'i' at the current iteration 't', P_{id}^t represent as the personal best position of particle found so far, p_{gd} represent as the global best particles position.

After calculating the velocity, the particle's position is updated using the equation as follows:

$$X_{id}^{t+1} = X_{id}^t + V_{id}^{t+1} \quad \dots (2)$$

whereX_{id}^t represent as current position of the particles of current iteration t and X_{id}^{t+1}represents the new position of the particles in next iteration t+1. Further, PSO algorithms are modified to improve its exploration and exploitation parameters by mathematical computation and hybridization of other meta-heuristics algorithms.

III. VARIANTS ON PARTICLE SWARM OPTIMIZATION (PSO)

In conventional PSO, each particles are learns from its own local experience and neighbor best solutions. The particle positions are not updated in a certain number generations, whenever the particles select the local optimum solution as the global best solution (A. RezaeeJordehi, 2015; F. Yano et al., 2007). To overcome these issues and to provide balance between exploitation and exploration several distinct variants of PSO are introduced in the literature. This section deals with the variants of PSO and their modifications with conventional PSO.

3.1. Directionally Driven Self-Regulating Particle Swarm Optimization (DDSRPSO)

M.R. Tanweer et al., proposed a novel algorithm named Directionally Driven Self-Regulating Particle Swarm Optimization (DDSRPSO) in 2015 (M.R. Tanweer et al., 2016). This algorithm contributes two new learning approaches namely, a directional update strategy and a rotational invariant strategy. However, it is an improved variant of Self-Regulating Particle Swarm Optimization (SRPSO) (M.R. Tanweer et al., 2015). DDSRPSO split the size of population particles into three groups based on its fitness value such as elite group (best particles); poorly perform particles (worst particle) and remaining particles. Elite group particles are follows the same strategy of SRPSO and the improvisation is done by using a group of elite particles (best particles) to achieve directional updates. Here, authors consider that all the worst particles follow the global search direction and the SPSO velocity is modified as follows.

$$V_{kd}^{t+1} = \omega_k * V_{kd}^t + c_1 * r_1 * (lp_{kd}^t - X_{kd}^t) + c_2 * r_2 * (p_{gd}^t - X_{kd}^t) \dots (3)$$

where, ω_k is the SRPSO inertia weight to achieve slow motion in velocity changes, lp_{kd}^t represents the median of selected three best performing particles from elite group, p_{gd}^t represents the global best particles so for achieved in t generations, k is the particles from worst particles group.

All other remaining particles are performed either SRPSO strategy of self-perception of global search direction or rotational invariant strategy to search the rotation variance property of the search space. Rotational invariant strategy is defined by the center of gravity calculated by three points such as current position of the particles, point beyond the personal best particles and point beyond the global best positions. In SRPSO, the centre of gravity is aid to compute a new position of particles which intern lead to unaware of global best positions. In DD-SRPSO, the self-perception of particles for global search directions p_i^{so} plays a major role to particles to aware of global search directions.

$$\overrightarrow{p_i^t} = \overrightarrow{X_i^t} + c_1 * r_1 \otimes (\overrightarrow{p_i^t} - \overrightarrow{X_i^t}) \quad \dots (4)$$

$$\overrightarrow{p_g^t} = \overrightarrow{X_i^t} + c_2 * r_2\, \overrightarrow{p_i^{so}} (\overrightarrow{p_g^t} - \overrightarrow{X_i^t}) \quad \dots (5)$$

$$\overrightarrow{G_i^t} = \begin{cases} \frac{\overrightarrow{p_i^t} + \overrightarrow{p_g^t} + \overrightarrow{X_i^t}}{3} & if\ \overrightarrow{p_i^{so}} = 1 \\ \frac{\overrightarrow{p_i^t} + \overrightarrow{X_i^t}}{2} & if\ \overrightarrow{p_i^{so}} = 0 \end{cases} \quad \dots (6) \qquad (3.4)$$

Where, $\overrightarrow{G_i^t}$ represents the centreof gravity, which is used to select the stochastic point of $\overrightarrow{X_i^t}$ particles in the hypersphere $H_i\left(\overrightarrow{G_i^t}, \left\|\overrightarrow{G_i^t} - \overrightarrow{X_i^t}\right\|\right)$, $\overrightarrow{p_i^t}$ represents the personal best position of particles i in the t iterations, $\overrightarrow{p_g^t}$ defines the global best position of particles i, $\overrightarrow{p_i^{so}}$ represents the self-perception of particles with social cognition (ie., $\overrightarrow{p_i^{so}} = 1$) and without social cognition (ie., $\overrightarrow{p_i^{so}} = 0$) , r_1, r_2 are random numbers between (0,1), symbol $\otimes$ represents a certain mathematical operation between the two opted attributes. Thus the rotational invariant characterictics modified the velocity equationas follows:

$$\overrightarrow{V_i^{t+1}} = \omega * \overrightarrow{V_i^t} + H_i\left(\overrightarrow{G_i^t}, \left\|\overrightarrow{G_i^t} - \overrightarrow{X_i^t}\right\|\right) - \overrightarrow{X_i^t} \quad \dots (7)$$

where, $\overrightarrow{V_i^{t+1}}$ represents the updation of new position velocity of particles in next iteration (t+1), represents the current velocity of the particles.

3.2. Bisection PSO (BPSO)

Bin Jiang et al, introduced Bisection PSO (BPSO) to optimize photonic crystal band gap. BPSO divides the search space for each dimension with the optimal particle at the center. After initialization, evaluation of fitness and optimal search point are processed based on the division of the search space (Bin Jiang et al., 2011). If the search space is halved, then particles are divided into multiple generations based on search space division. Otherwise the conventional PSO method will be followed without any search space division.

$$V_{id}^{t+1} = W_{LDW} * V_{id}^t + C_1 * r_1 * (p_{id}^t - X_{id}^t) + C_2 * r_2 * (p_{gd}^t - X_{id}^t) \dots (8)$$

$$x_{id}^{new} = x_{id}^{old} + V_{id}^{new} \quad \dots (9)$$

where, ω_{LDW}represents the linearly decreasing inertia weight (LDW), p_{id}^t represents the personal best of particle i of d dimension in t iterations, p_{gd}^t defines the global best position obtained so far in the t iterations, X_{id}^t represents the current particle i position. Other features include improvising cognitive and social parameter. Enhancing cognitive parameter to emphasize search space exploration and for improvising convergence rate at end of iterations social parameter values were been set linearly.

$$C_1 = 1 + e^{\frac{-i}{nmax}} \quad \dots (10)$$

$$C_2 = 1 - e^{\frac{-i}{nmax}} \quad \dots (11)$$

where, i is current iteration number and $nmax$ is maximum number of iterations. C_1 cognitive parameter gives larger value when compared with social parameter at initial stage of iteration. In order to explore search space and over time of iterations cognitive parameter decreases its value periodically and concentration of social parameter plays high role in order to exploit the solutions.

3.3. Dispersed PSO (DPSO)

XingjuanCai proposed Dispersed PSO (Xingjuan Cai et al., 2008), to improve the convergence speed in PSO by deploying dispersed co-efficient setting in social parameter instead of being centralized in conventional PSO. Conventional PSO is said to be centralized since all particles moves towards a Gbest value and because of this the individual best which holds historical best become unnoticed.

To improve the conventional PSO, Dispersed PSO introduced with an index which holds performance difference between fitness value of the best, worst and current values of particles. The Grade of computed fitness values is represented as

$$Grade_u(t) = \frac{f_{worst}(t) - f(x_u(t))}{f_{worst}(t) - f_{best}(t)} \qquad \ldots (12)$$

where, $Grade_u(t)$determined as the ratio of fitness difference between the numerator of worst $f_{worst}(t)$ and current particle $f(x_u(t))$ and denominator of worst particle with the global best particle $f_{best}(t)$ in t iterations.

With the help of $Grade_u(t)$ social parameter is redefined as

$$C_2(t) = C_{low} + (C_{up} - C_{low})Grade_u(t) \qquad \ldots (13) \qquad (3.10)$$

where, C_{low} and C_{up} are two constant numbers. This redefined social parameter introduces the concept of exploitation in around the neighborhood of best individuals. It also describes about the high rate of exploration, at time of worst individuals comes into attack. Then the solutions go under the process of mutation, in order to avoid premature convergence. Individual chosen for mutation will be of the uniformly random manner.

Mutation strategy is incorporated with the dispersed PSO to improve the local search ability among the particles with the conditional parameter value of random number which satisfies the threshold (0.5). Mutation strategy equation is as described as follows

$$V_{jk}(t) = \begin{cases} 0.5\ X\ x_{max}\ X\ r_1 if\ r_2 < 0.5 \\ 0.5\ X\ x_{max}\ X\ r_1 \quad otherwise \end{cases} \qquad \ldots (14)$$

where, r1 and r2 are random numbers between ~ (0, 1). At each step of iterations only one variable participates under mutation process.

3.4. Compact PSO (CPSO)

Ferrante Neri et al., proposed Compact PSO (cPSO) (Ferrante Neri et al., 2013) which has the tendency to work on memory spaces. Minimal memory spaces which expressed as minimal in nature like embedded environments. In cPSO, a Perturbation Vector (PV) has been initialized instead of conventional PSO swarm population. It has binary vectors in its population.

$$PV^t = \mu^t \sigma^t \qquad \ldots (15)$$

where μ, σ are design variable mean and standard deviation which will be built at the iterations of cPSO and t is current iteration number. After the initialization of PV, cPSO initializes the μ[i], $\sigma[i]$ where i is the variable number. Then the personal best and global best positions are taken from PV vector. Random generations of particles positions and velocities were generated and updated as follows.

$$V_{id}^{t+1} = \emptyset_1 V^t + \emptyset_2 U_1 (P_{id}^t - x^t) + \emptyset_3 U_2 (X_{gd}^t - X_{id}^t) \qquad \ldots (16)$$

$$X_{id}^{t+1} = \gamma_1 X_{id}^t + \gamma_2 V_{id}^{t+1} \qquad \ldots (18)$$

where, $\emptyset_1, \emptyset_2, \emptyset_3$ are the fine-tuning parameters of meta-optimization of number of particles, U_1 and U_2 are constants. X_{gd}^t represents the global best solution found so far in t iterations, d dimensions, X_{id}^t represents the current particle position for d dimension in t iterations. After these steps, winner and loser (best, worst) particles have been calculated from generated X_{id}^{t+1}, whereas P_{id}^t are updated using Perturbation Vector (PV). Updating the design variables of the next generation as follows

$$\mu^{t+1}[i] = \mu^t[i] + \frac{1}{N_p}(winner[i] - loser[i]) \qquad \ldots (19)$$

$$\sigma^{t+1}[i] = \sqrt{\begin{matrix}(e^t[i])^2 + (\mu^t[i])^2 - (\mu^{t+1}[i])^2 + \\ \frac{1}{N_p}(winner^2[i] - loser^2[i])\end{matrix}} \qquad \ldots (20)$$

where, N_p represents the virtual size population, e^t determines the exponential value at t iterations, $winner[i], loser[i$ represents the winner and loser particles in each iteration helps to current particles to update with the winner of current iteration if it holds best value than previous best.

3.5. Hybrid Imperialist competitive algorithm and Particle Swarm Optimization (ICAPSO)

Lhassane Idoumghar et. al., proposed a new hybrid algorithm known as hybrid Imperialist competitive algorithm and Particle swarm optimization (ICAPSO) (Lhassane Idoumghar et. Al., 2013). This algorithm aims to enhance the diversity of the solutions viz., crowding distances of a particles and crossover operator (Nitasha Soni and Dr .Tapas Kumar, 2014) also used to improve the memory of each individuals of the population. In ICA algorithm all individuals are grouped as an empire and the best fitness solutions of the population are grouped as strong empire whereas the worst fitness solutions of the population are grouped as weak empire. This algorithm strengthen the strongest empire and suppress the weaken empire. The initial grouping in imperialist mechanism is based on the normalized power of individuals. The imperialistic competition has been taken place to pick the weakest solution from the weaken empire. ICA algorithm (Bo Xing and Wen-Jing Gao, 2013) hybridized with the classical PSO to improvise the exploration and the diversity of the solution in a search space.

Initially, all the individuals positions are settled as random and first N_{emp} individuals are selected as imperialists and N_{col} are the colonies divided for the each empires with the help of $\frac{N-N_{emp}}{N_{emp}}$ colonies. Computations of each individual's fitness are termed as crowding distance. Based on the crowding distance the worst solutions are removed from the population and the best solution among all individuals is stored in global archive and then the individual best are stored in the local archive. The velocity of the each individual is termed as colony speed which is modified as the follows:

$$V_i^{t+1} = \omega . v_i^t + r_1 c_1 (P_i^t - X_i^t) + r_2 c_2 (E_i^t - X_i^t) \text{With } -V_{Max} \leq V_i^t \leq V_{max} \quad \text{... (21)}$$

where ω is the inertia factor, E_i^t is the position of the colonies imperialist, X_i^t represents position of the colony, P_i^t is the best colony's position, r_1 and r_2 are the random numbers between 0 and 1. Local archive is improved with the aid of crossover operator by mixing the local archive of the colony and local achieve of the imperialist.

3.6. Diversity enhanced particle swarm optimization with neighborhood search (DNSPSO)

In (Hui Wang et. al., 2013) proposed a hybrid mechanism known as diversity enhanced particle swarm optimization (DNSPSO), which seek to maintain the intensification and diversification search abilities. Diversity enhanced operator is used to overcome the premature convergence and to maintain the diversity in all the individuals.

The diversity of the individuals is computed using equation as follows:

$$Diversity = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\sum_{j=1}^{D} (x_{ij}(t) - \bar{x}_j(t))^2} \quad \text{... (22)}$$

$$\bar{x}_j(t) = \frac{\sum_{i=1}^{N} x_{ij}(t)}{N} \quad \text{... (23)}$$

Neighborhood strategy helps to outfit the premature convergence by identifying the fittest solution within the local search space. The neighborhood strategy is classified into categories such as local neighborhood strategy and global neighborhood strategy. In the local neighborhood strategy, improves the exploitation with the help of the particle best solution (pbest).

$$LX_i = r_1 . x_i + r_2 . pbest_i + r_3 (x_c - x_d) \quad \text{... (24)}$$

$$LV_i = V_i \quad \text{... (25)}$$

where, x_i is the current position vector of the ith particle, $pbest_i$ is the previous best particle of x_i, x_c, x_d are the position vectors of two random particles in the k-neighborhood radius of x_i, r_1, r_2, r_3 are three uniform random numbers within (0,1), and $r_1 + r_2 + r_3 = 1$.

Global neighborhood strategy helps to improve the exploration in a global search space. The trial solution is given by G_i = (GX_i, GV_i)

$$GX_i = r_4 . x_i + r_5 . gbest + r_6 . (x_e - x_f) \quad \text{... (26)}$$

$$GV_i = V_i \quad \text{... (27)}$$

where, GX_i represents the global neighbor position for i particles, r_4, r_5, r_6 are uniform random numbers between (0,1), $gbest$ is the global best position found so far, x_e, x_f are selected neighborhood particles x_i, GV_i is the global neighborhood velocity, this strategy helps to solve the multimodal problems.

3.7. Hybrid Particle Swarm and Swallow Swarm Optimization (HPSSO)

A. Kavehet. al., proposed a novel mechanism known as hybrid particle swarm and swallow swarm optimization (HPSSO) (A. Kaveh et. al., 2014). The main aim of the algorithm is to consider promising particles other than global best particles. Here, they consider the swallow swarm optimization to control the high-speed flying which aids to overcome the convergence speed affect. Swallow swarm splits the total population into sub colonies such as leader, explorer and aimless particles. Leader particles split up into two leaders such as a local leader and head leader, the global best particle are termed as head leader, whereas the particle with local best position is notified as local leader. All other particles termed as explorer particles which needs to change their positions. The changes in explorer particles is processed by the velocity noted identified by head leader and corresponding local leader. The formulation of this algorithm is given by mathematical representation as follows:

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad \text{... (28)}$$

$$V_i^{t+1} = VHL_i^{t+1} + VLL_i^{t+1} \quad \text{... (29)}$$

$$VHL_i^{t+1} = VHL_i^t + \alpha_{HL} rand() (Xbest_i^t - X_i^t) + \beta_{HL} rand() (HL^t - X_i^t) \quad \text{... (30)}$$

$$VLL_i^{t+1} = VLL_i^t + \alpha_{LL} rand() (Xbest_i^t - X_i^t) + \beta_{LL} rand() (LL_i^t - X_i^t) \quad \text{... (31)}$$

where, VHL_i^t represents the current velocity of the head leader particles, VLL_i^t represents the current velocity of the local leader particles, α_{HL} and β_{HL} are the random numbers of the head leader particles, α_{LL} and $\beta_{LL}$are the random numbers of the local leader particles, HL^t is the current local head leader and LL_i^t is the local leader of the particles i at the current iteration t.

3.8. Hybrid PSO with Gravitational Search Algorithm (HPSOGSA)

In (ShanheJiang, et al., 2014), proposed a new variant which is a collaboration between conventional PSO and Gravitational search algorithm (HPSOGSA). GSA algorithm (Esmat Rashedi et al., 2009) is inspired from the law of gravity and cooperation between masses. They have tested the performance of this algorithm with benchmark functions and designed it especially for economic emission load dispatch problems. This algorithm introduced to maintain the balance between the local search and global search. Generally in PSO, the particle best and global best are measured to adjust the position of the particles while GSA uses fitness value to adjust the position of the particle.

The hybridization of PSO and GSA modified the velocity based on the adjusting both the velocities of the PSO and GSA. The combination of these algorithms new velocity value is computed and these value aids to update the position of the particles.

$$V_i^d(t+1)_{HPSO-GSA} = c_1 * r_1 + v_i^d(t+1)_{PSO} + c_2 * (1-r_1) + v_i^d(t+1)_{GSA} \quad \dots (32)$$

where, $V_i^d(t+1)_{HPSO-GSA}$ represents the new velocity obtained from hybrid HPSO-GSA in iteration t for each dimension d, c_1, c_2 denotes the some constant value to adjust the position of the particles, these constant value linearly decreases from 2 to 0 whereas r_1, r_2 determine the random values (0,1). $v_i^d(t+1)_{PSO}$ mention the velocity obtained from the generic PSO in each iteration t. The PSO velocity update equation is represented in equation (33), $v_i^d(t+1)_{GSA}$ denotes the velocity update of gravitational search algorithm is given below:

$$v_i^d(t+1)_{GSA} = rand_i * V_i^d(t) + a_i^d(t) \quad \dots (33)$$

where, $a_i^d(t)$ presents the acceleration of agent i in the dth dimension at each generation t, $rand_i$ is the uniform stochastic variable in the interval (0, 1).

Table 1 Description of Unimodal function ($f_1 - f_7$)

Function	*Dim*	*Range*	f_{m}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100, 100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[-100,100]	0
$f_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30	[-100, 100]	0
$f_5(x) = \sum_{i=1}^{n} \lvert 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \rvert$	30	[-30, 30]	0
$f_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	[-100, 100]	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random(0,1)$	30	[-1.28, 1.28]	0

IV. EXPERIMENTAL RESULTS

The numerical efficiency of Generic PSO and its variants are tested in twenty three benchmark test functions (K. Tang et al., 2007; Momin Jamil and Xin-She Yang, 2013). The collected benchmark functions are classical functions used by many researchers. Despite the simplicity, these test functions are chosen to compare our simulated results to those of the current meta-heuristics.

The three classical benchmark classified into three different groups: Group A holds seven different functions $f_1 - f_7$ which are all unimodal, Group B holds six different functions $f_8 - f_{13}$ which are all multimodal and Group C holds remaining set of functions $f_{14} - f_{23}$ which are all fixed dimension functions (Seyedali Mirjalili and Andrew Lewis, 2016). Unimodal test functions hold one optimum (either minimum or maximum), so they can benchmark the intensification and convergence of an algorithm. Multi-modal test functions hold more than one optimum solution, which formulates these functions more challenging than unimodal functions. These functions hold one global optimum solution and many local optima solutions. The main motive of all algorithms is to approximate the global optimum and to avoid all the local optima.

The multi-modal functions examine to analyze the efficiency of the algorithms in contributing exploration and local optima avoidance. The last group is fixed dimension multi-modal test functions are consists only few local optimum solutions. This function does not permit to tune the number of design variables, but they offer peculiar search space. Three group's benchmark functions are listed in Tables **1-3** where *Dim* represents the dimension of the function f_{min} represents the optimum value and *Range* is the boundary of the functions search space. Typical 2D plots of the benchmark functions are illustrated in Figure **1-3.**

Table 2 Description of Multimodal Benchmark functions ($f_8 - f_{13}$)

Function	*Dim*	*Range*	f_{min}
$f_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[-500, 500]	-418.9829 × 5
$f_9(x) = \sum_{i=1}^{n} \lvert x_i^2 - 10\cos(2\pi x_i) + 10\rvert$	30	[-5.12, 5.12]	0
$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[-32, 32]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600, 600]	0
$f_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	30	[-50,50]	0
$y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a & < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$f_{13}(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n} \frac{(x_i - 1)^2\lvert 1 + sin^2(3\pi x_i + 1)\rvert + (x_n - 1)^2}{[1 + sin^2(2\pi x_n)]}\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[-50, 50]	0

Table 3 Description of fixed dimension multimodal functions ($f_{14} - f_{23}$)

Function	Dim	Range	f_{min}
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[-65, 65]	1
$f_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_{3+} x_4}\right]^2$	4	[-5,5]	0.0003
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	2	[-5, 5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	[1,3]	-3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	[0,1]	-3.32
$f_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.1532
$f_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$f_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	-10.5363

All algorithms are implemented in Matlab (Version 2013 a) whereas the population size and maximum generation equal to 30 and 500 have been utilized and executed in 30 independent runs. Table **4-5** reports the statistical results of best, mean and std (Standard Deviation). The Generic PSO with its variants is compared with the convergence rate and computation time required to obtain the best optimum or nearer best optimum solutions.

The figure 1 shows 2D representation of the unimodal benchmark functions $f_1, \ldots, f_7$ where as f_1 is the sphere function whose upper and lower bound in the interval of [-100,100] with 30 dimension set. f_2 is Schwefel problem 2.22 with interval of [-10,10] along with 30 dimension. f_3, f_4 represents the Schwefel problem 1.2 and 2.21 with lower and upper limit in a range of [-100, 100]. f_5 named as Rosenbrock function with interval of [-30,30] and f_6, f_7 determines the step function and quartic with random noise in the interval of [-100,100] and [-1.28,1.28].

Figure 1 Typical 2D representation of Unimodal benchmark mathematical functions

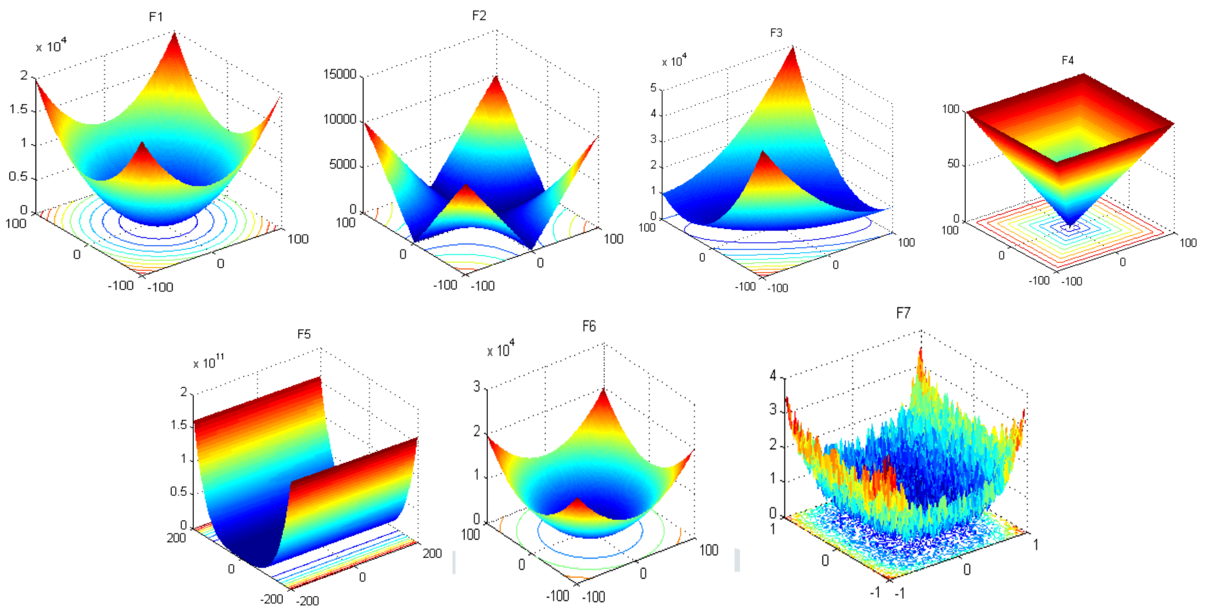


Figure 2 Typical 2D representation of Multimodal benchmark mathematical functions

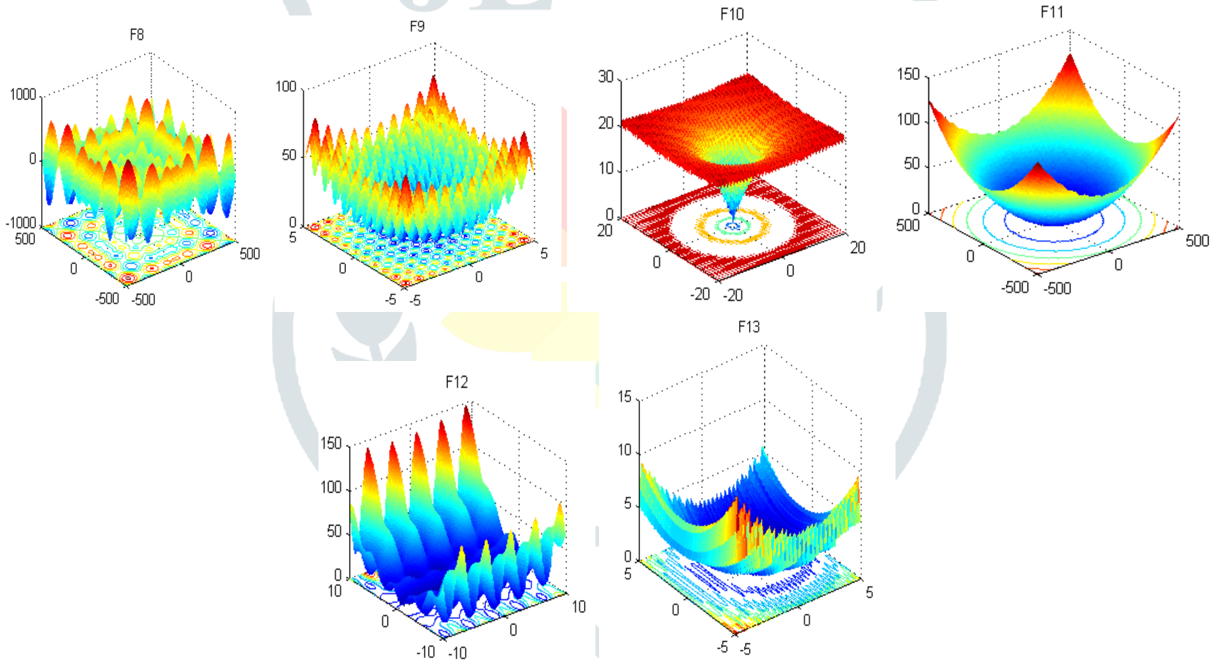
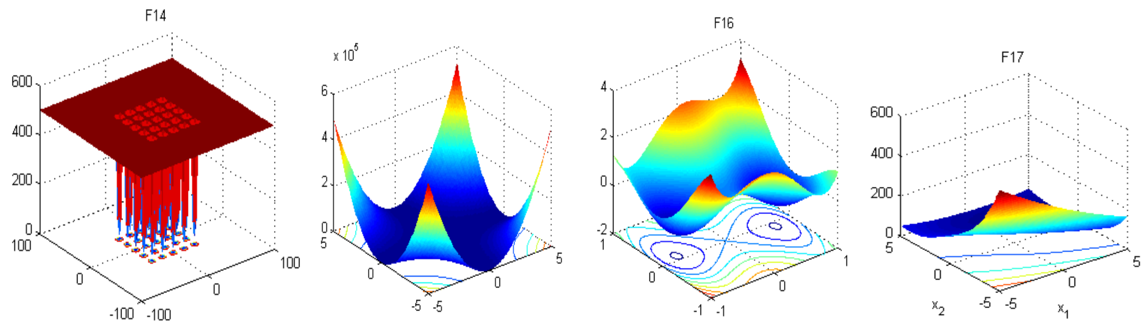
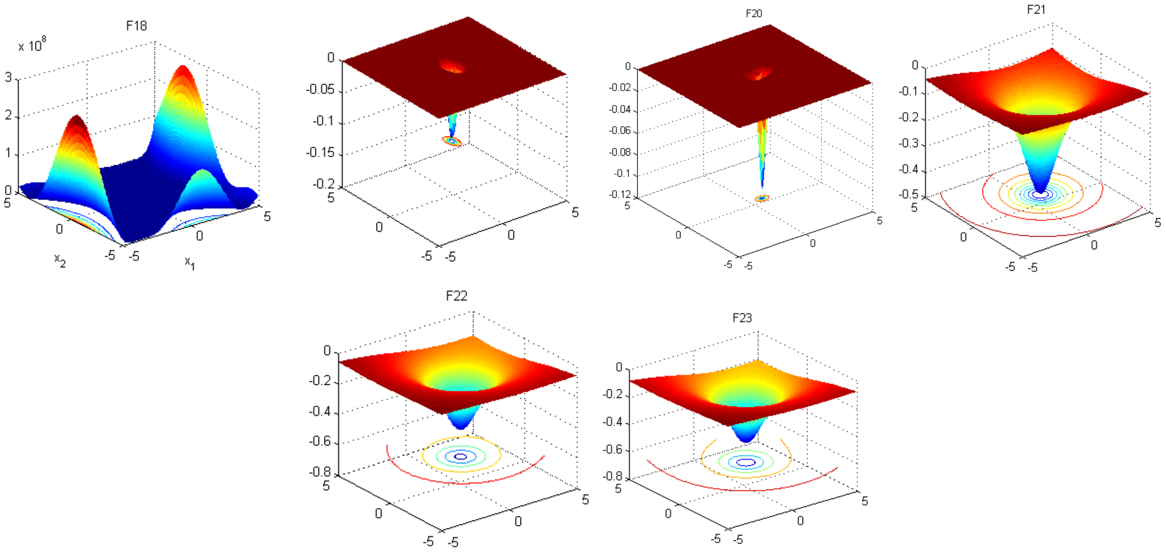


Figure 3 Typical 2D representation of Fixed-Dimension Multimodal benchmark mathematical functions





The figure 2 displays that 2D representation of the multimodal benchmark functions $f_8, \ldots, f_{13}$ where as f_8 is the Schwefel problem 2.26 with upper and lower bound in the interval of [-500,500]. f_9 determines Rastrigin function with interval of [-5.12,5.12] along with 30 dimension. f_{10}, f_{11} represents the Ackley and Griewank function with lower and upper limit in a range of [-30, 30] and [-600,600] and f_{12}, f_{13} determines the penalized function 1 and 2 in the interval of [-50,50].

The 2D representations of fixed dimension multimodal function were shown in figure 3. f_{14} is the fifth De Jong problem (Shekel's Foxholes) with upper and lower bound in the interval of [-65.536, 65.536] in which the dimension of these function is 2 and the global minimum of this function was $(-32, -32) \cong 0.998004$ [58]. f_{15} determines Kowalik function with interval of [-5, 5] along with 4 dimension. f_{16} denotes that the Six-hump Camel function with the range of min and max [-5, 5] in 2 dimension set. f_{17} is the test function of Brainin-Hoo function in the interval of [-5,5] and its global optimum is 0.398. f_{18} is the test problem of Goldstein-Price which have [-2, 2] boundary range with 2 dimensional set. Similarly f_{19}, f_{20} determines the test function of Harmann with 3 and 6 dimensional sets in the interval of [0, 1]. f_{21}, f_{22}, f_{23} were the test problem of Shekel function 5,7 and 10 with 4 dimension set in the interval range of [0,1].

4.1 Evaluation of exploitation/intensification capability for 30-D unimodal functions ($f_1 - f_7$)

Unimodal functions $f_1 - f_7$ contain only one global optimum. These functions grant to compute the intensification capability of the investigated meta-heuristic algorithms. The experiments of unimodal functions for all algorithms are computed and the convergence graph of all these functions are graphically represented in figure 4. Likewise, the results of unimodal function which is applied in all the algorithms are shown in table 1 and 2.

From the observed results, the sphere function (f_1) results says that Generic PSO, DNSPSO, ICAPSO get a solution beyond the global minimum, whereas in initial iterations other algorithm performs better but in course of iterations, convergence rate are reduced and they are not attained the optimum solutions. In Schwefel problem 2.22 (f_2), initially all the algorithms converges towards the optimum solutions and later they sustain in the local solutions which in turn its convergence rate is reduced.

Apart from those algorithms ICAPSO finds the global optimum solution in a less computation time and still it converges beyond the optimum solutions. For Schwefel problems 1.2 (f_3) and 2.21 (f_4) we observed the results that in initial iteration itself the DNSPSO converges nearer to optimum solution and then later it slowly converges to the global optimum solutions whereas all other algorithms provides average convergence rate in initial start of iteration but later it reduces their convergence towards the optimum over 500 iterations. In Rosenbrock function(f_5), the global minimum lies inside a narrow long and parabolic shaped flat valley so identifying the minimum is difficult. From initial iteration onwards, DNSPSO converges faster for f_5, f_7 compare to all other PSO variants and later ICAPSO converges when it reaches near 100 iterations. For step function f_6, BPSO and HPSOGSA obtain rare optimal solutions over 30 independent runs, whereas for the same function Generic PSO and DNSPSO find good solution close to the global optimizer.

Figure 4. Comparison of convergence rate on Generic PSO and eight variant algorithms obtained in Unimodal benchmark problems a) Sphere function, (b) Schwefel problem 2.22, (c) Schwefel problem 1.2, (d) Schwefel problem 2.21, (e) Rosenbrock function, (f) step function, (g) Quartic function

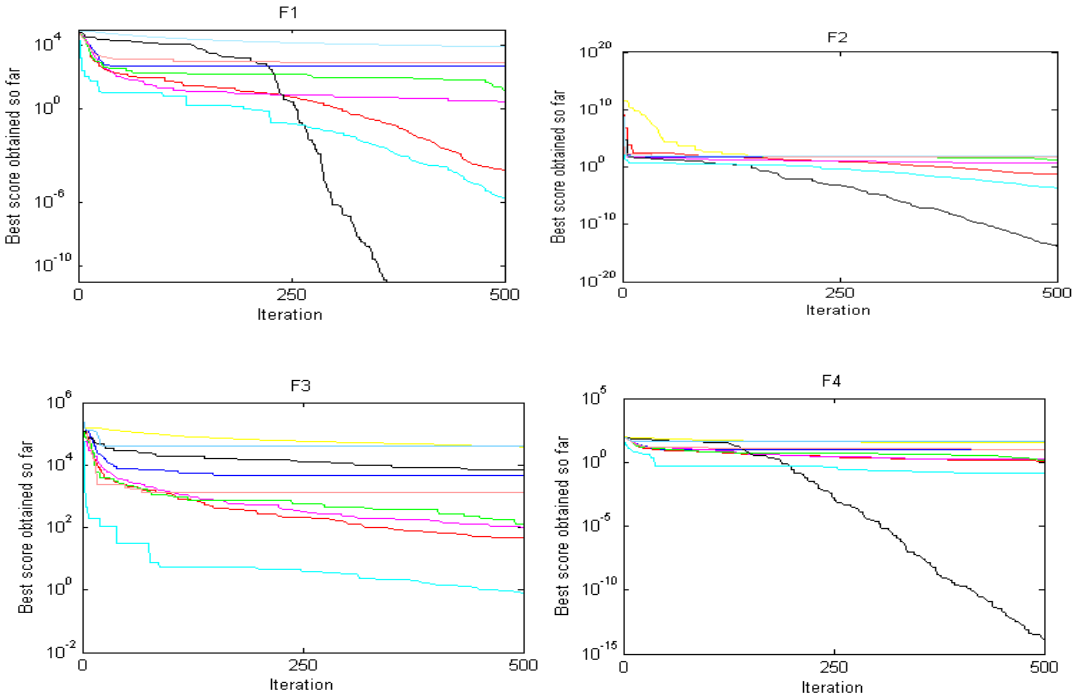


Table 4. Results for the unimodal 30-dimensional functions $(f_1, \ldots, f_7)$. The experiments were repeated 30 independent runs and the best, mean and standard deviation (std)

fun	PSO			DDSRPSO			BPSO			CPSO			DPSO		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
f1	1.36E-05	1.09E-04	1.39E-04	1.88E+02	1.60E+03	6.29E+03	5.83E+03	6.86E+03	7.06E+02	1.78E+00	3.19E+00	1.00E+00	2.79E+00	7.35E+00	5.50E+00
f2	3.68E-03	2.98E-02	4.46E-02	2.02E+01	9.07E+09	2.03E+11	7.18E+01	9.69E+01	1.38E+01	3.94E+00	1.08E+01	7.56E+00	1.05E+01	2.36E+01	6.62E+00
f3	1.83E+01	9.45E+01	4.91E+01	2.45E+03	5.31E+03	1.12E+04	2.02E+04	3.16E+04	7.50E+03	7.93E+01	1.43E+02	4.92E+01	7.63E+01	1.48E+02	3.92E+01
f4	5.42E-01	1.07E+00	1.97E-01	1.05E+01	1.41E+01	7.28E+00	2.79E+01	3.34E+01	3.26E+00	1.46E+00	2.00E+00	2.60E-01	1.09E+00	1.42E+00	1.90E-01
f5	1.76E+01	1.11E+02	9.48E+01	3.62E+04	1.91E+05	1.14E+05	2.09E+06	2.97E+06	6.42E+05	3.40E+02	7.63E+02	3.22E+02	1.15E+03	2.88E+03	1.66E+03
f6	3.31E-06	2.85E-04	5.03E-04	2.42E+02	4.56E+02	9.78E+01	5.70E+03	6.93E+03	7.73E+02	1.51E+00	2.99E+00	7.39E-01	2.75E+00	7.31E+00	3.51E+00
f7	1.15E-01	1.75E-01	4.74E-02	5.82E+00	4.21E+01	3.40E+01	1.41E+01	1.57E+01	1.04E+00	3.72E-01	2.23E+00	4.27E+00	9.06E-01	6.06E+00	5.89E+00

fun	ICAPSO			DNSPSO			HPSSO			HPSOGSA		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
f1	**4.17E-32**	1.83E+04	7.15E+02	**3.74E-08**	7.01E-06	9.28E-06	5.45E+02	6.85E+02	6.59E+01	7.49E+03	2.16E+04	1.03E+04
f2	1.70E-19	2.63E-14	1.32E-13	1.57E-04	1.52E-03	1.66E-03	4.47E+01	5.20E+01	3.24E+00	5.70E+01	1.41E+06	6.77E+06
f3	6.02E-04	1.44E+04	1.14E+04	4.62E-03	1.92E-01	1.89E-01	5.88E+02	1.11E+03	2.14E+02	1.13E+04	4.20E+04	2.13E+04
f4	4.43E-13	3.98E-03	2.18E-02	1.30E-02	7.13E-02	4.42E-02	7.56E+00	9.86E+00	7.14E-01	2.78E+01	6.10E+01	1.44E+01
f5	2.78E+01	2.81E+01	2.37E-01	2.62E+01	2.67E+01	2.73E-01	2.50E+05	5.31E+05	1.52E+05	9.46E+06	4.38E+07	3.41E+07
f6	2.20E+00	6.21E+01	3.24E+02	6.28E-06	1.32E-04	1.81E-04	4.15E+02	6.94E+02	8.98E+01	1.14E+04	2.03E+04	5.87E+03
f7	1.04E-05	4.57E-04	4.00E-04	9.84E-04	7.43E-03	4.84E-03	6.14E+01	1.21E+02	2.50E+01	4.61E+01	1.06E+02	2.48E+01

Table 5. Results for the Multimodal 30-dimensional functions $(f_8, \ldots, f_{13})$. The experiments were repeated 30 independent runs and the best, mean and standard deviation (std)

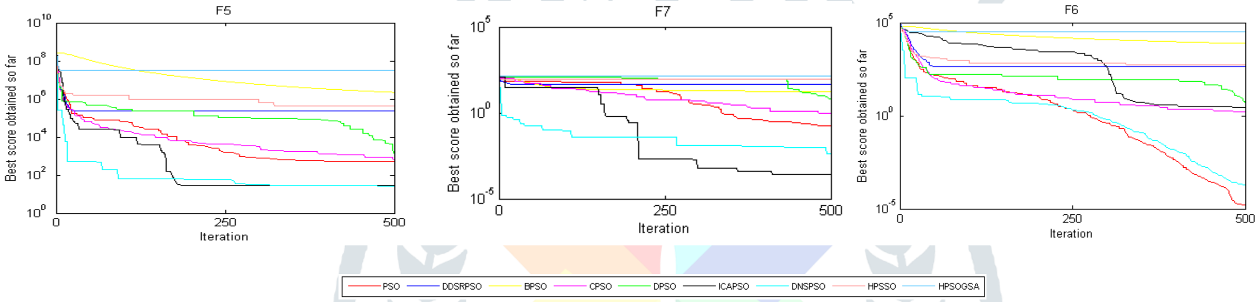
f	PSO			DDSRPSO			BPSO			CPSO			DPSO		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
f8	-7.80E+03	-4.79E+03	1.61E+03	-6.79E+03	-3.73E+03	1.45E+03	-3.84E+03	-3.05E+03	3.16E+02	-8.94E+03	-6.00E+03	9.95E+02	-8.57E+03	-4.73E+03	1.68E+03
f9	3.25E+01	5.62E+01	1.06E+01	1.05E+02	1.87E+02	4.95E+01	2.92E+02	3.37E+02	1.93E+01	9.63E+01	1.69E+02	4.36E+01	2.18E+02	2.73E+02	2.39E+01
f10	2.16E-03	1.36E-01	4.64E-01	6.98E+00	8.83E+00	6.83E-01	2.00E+01	2.02E+01	9.63E-02	1.58E+00	2.49E+00	3.53E-01	2.35E+00	3.26E+00	5.00E-01
f11	8.94E-07	8.06E-03	7.77E-03	2.24E+00	3.22E+00	5.32E-01	5.63E+01	6.48E+01	5.03E+00	1.14E-01	2.04E-01	5.51E-02	7.47E-02	2.44E-01	1.68E-01
f12	5.31E-08	1.38E-02	4.50E-02	6.52E+00	2.04E+01	1.10E+01	2.27E+04	1.90E+05	1.67E+05	2.55E-02	2.36E-01	2.54E-01	1.22E-01	9.10E-01	1.61E+00
f13	2.53E-06	5.56E-03	8.99E-03	1.62E+02	3.66E+03	4.57E+03	1.51E+06	3.49E+06	1.28E+06	3.04E-01	5.65E-01	1.39E-01	1.29E+00	3.54E+00	1.12E+00

f	ICAPSO			DNSPSO			HPSSO			HPSOGSA		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
f8	-3.10E+03	-2.10E+03	5.11E+02	-1.34E+04	-8.98E+03	1.65E+03	-4.21E+03	-2.76E+03	5.70E+02	-4.12E+03	-3.44E+03	3.97E+02
f9	0.00E+00	1.43E-10	7.82E-10	8.66E-07	4.35E-05	5.95E-05	2.78E+02	3.27E+02	1.65E+01	3.54E+02	3.98E+02	2.25E+01
f10	4.44E-15	1.61E-13	4.61E-13	5.99E-05	5.41E-04	3.20E-04	9.95E+00	1.08E+01	3.30E-01	1.56E+01	1.78E+01	9.52E-01
f11	5.06E+02	5.85E+02	4.42E+01	1.73E-08	1.72E-03	6.23E-03	1.79E+00	2.01E+00	1.16E-01	8.30E+01	2.16E+02	7.25E+01
f12	1.27E-01	2.10E-01	6.01E-02	5.08E-08	8.88E-07	9.87E-07	9.73E+00	1.29E+01	1.75E+00	1.10E+05	9.47E+07	1.13E+08
f13	1.16E+00	1.48E+00	1.82E-01	1.27E-06	3.72E-03	5.28E-03	2.07E+02	6.73E+03	5.70E+03	1.28E+07	3.15E+08	2.11E+08

Table 6. Results for the fixed dimensional multimodal functions $(f_{14}, \ldots, f_{23})$. The experiments were repeated 30 independent runs and the best, mean and standard deviation (std)

F	PSO			DDSRPSO			BPSO			CPSO			DPSO		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
f_{14}	9.98E-01	2.31E+00	2.07E+00	9.98E-01	1.10E+00	3.66E-01	9.98E-01	4.04E+00	2.48E+00	9.98E-01	3.46E+00	2.99E+00	9.98E-01	4.15E+00	2.61E+00
f_{15}	3.27E-04	8.90E-04	1.56E-04	5.55E-04	1.03E-02	1.53E-02	8.22E-03	6.98E-02	5.03E-02	3.92E-04	1.04E-02	1.54E-02	6.91E-04	1.12E-03	6.86E-04
f_{16}	-1.03E+00	-1.03E+00	6.52E-16	-1.03E+00	-1.03E+00	6.61E-05	-1.03E+00	-8.77E-01	1.46E-01	-1.03E+00	-1.03E+00	5.38E-16	-1.03E+00	-1.03E+00	1.96E-05
f_{17}	3.98E-01	3.98E-01	0.00E+00	3.98E-01	3.98E-01	3.88E-06	3.98E-01	6.47E-01	3.75E-01	3.98E-01	3.98E-01	0.00E+00	3.98E-01	3.98E-01	7.59E-04
f_{18}	3.00E+00	3.00E+00	1.23E-15	3.00E+00	3.68E+00	3.72E+00	3.31E+00	9.29E+00	4.19E+00	3.00E+00	3.00E+00	3.19E-15	3.00E+00	3.00E+00	1.33E-09
f_{19}	-3.86E+00	-3.86E+00	2.64E-15	-3.86E+00	-3.86E+00	1.44E-03	-3.83E+00	-3.70E+00	8.29E-02	-3.86E+00	-3.86E+00	2.32E-15	-3.86E+00	-3.86E+00	2.07E-08
f_{20}	-3.32E+00	-3.27E+00	6.03E-02	-3.32E+00	-3.28E+00	6.53E-02	-3.03E+00	-2.31E+00	5.06E-01	-3.32E+00	-3.27E+00	5.99E-02	-3.31E+00	-3.19E+00	3.66E-02
f_{21}	-1.02E+01	-6.32E+00	3.51E+00	-1.02E+01	-8.17E+00	3.17E+00	-1.96E+00	-1.23E+00	3.21E-01	-1.02E+01	-5.71E+00	3.34E+00	-1.02E+01	-1.01E+01	3.95E-01
f_{22}	-1.04E+01	-8.81E+00	2.98E+00	-1.04E+01	-7.08E+00	3.75E+00	-3.13E+00	-1.74E+00	5.63E-01	-1.04E+01	-6.63E+00	3.67E+00	-1.04E+01	-1.01E+01	1.09E+00
f_{23}	-1.05E+01	-9.61E+00	2.41E+00	-1.05E+01	-8.11E+00	3.54E+00	-3.09E+00	-1.75E+00	5.24E-01	-1.05E+01	-5.10E+00	3.41E+00	-1.05E+01	-1.03E+01	6.19E-01

F	ICAPSO			DNSPSO			HPSSO			HPSOGSA		
	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
f_{14}	2.00E+00	1.11E+01	6.06E+00	9.98E-01	9.98E-01	7.14E-17	9.98E-01	1.65E+00	9.44E-01	2.01E+00	4.58E+00	1.96E+00
f_{15}	9.43E-04	7.19E-03	8.78E-03	3.08E-04	6.94E-04	2.93E-04	9.38E-04	1.82E-03	6.12E-04	6.10E-04	1.37E-03	1.41E-03
f_{16}	-1.03E+00	-1.03E+00	1.62E-03	-1.03E+00	-1.03E+00	6.32E-16	-1.03E+00	-1.03E+00	3.29E-03	-1.03E+00	-1.03E+00	1.10E-03
f_{17}	4.01E-01	4.56E-01	3.83E-02	3.98E-01	3.98E-01	0.00E+00	3.98E-01	4.01E-01	3.16E-03	3.98E-01	3.99E-01	2.48E-03
f_{18}	3.01E+00	3.14E+00	7.69E-02	3.00E+00	3.00E+00	1.44E-15	3.00E+00	3.24E+00	2.18E-01	3.00E+00	3.00E+00	1.12E-03
f_{19}	-3.86E+00	-3.77E+00	7.27E-02	-3.86E+00	-3.86E+00	2.70E-15	-3.86E+00	-3.80E+00	5.06E-02	-3.86E+00	-3.86E+00	5.78E-04
f_{20}	-3.17E+00	-2.93E+00	1.41E-01	-3.32E+00	-3.29E+00	5.11E-02	-3.05E+00	-2.47E+00	3.50E-01	-3.13E+00	-2.80E+00	2.06E-01
f_{21}	-9.43E+00	-4.97E+00	1.65E+00	-1.02E+01	-6.93E+00	2.48E+00	-8.02E+00	-3.76E+00	1.56E+00	-1.02E+01	-6.85E+00	3.34E+00
f_{22}	-9.61E+00	-5.10E+00	1.57E+00	-1.04E+01	-8.81E+00	2.48E+00	-8.21E+00	-3.83E+00	1.41E+00	-1.04E+01	-9.35E+00	2.33E+00
f_{23}	-9.89E+00	-5.38E+00	2.20E+00	-1.05E+01	-7.70E+00	2.71E+00	-8.46E+00	-4.66E+00	1.47E+00	-1.05E+01	-9.19E+00	2.67E+00

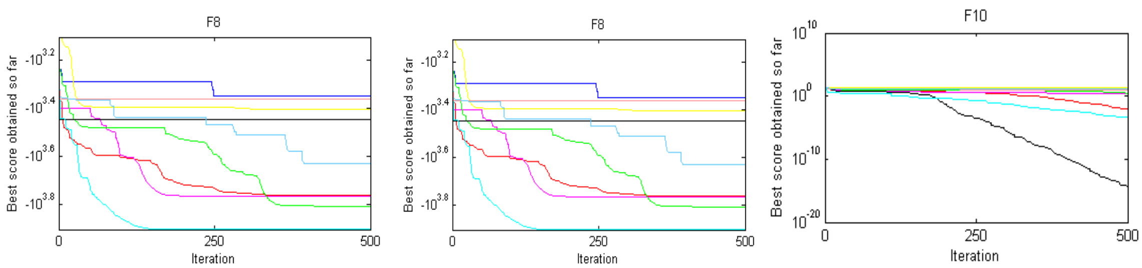


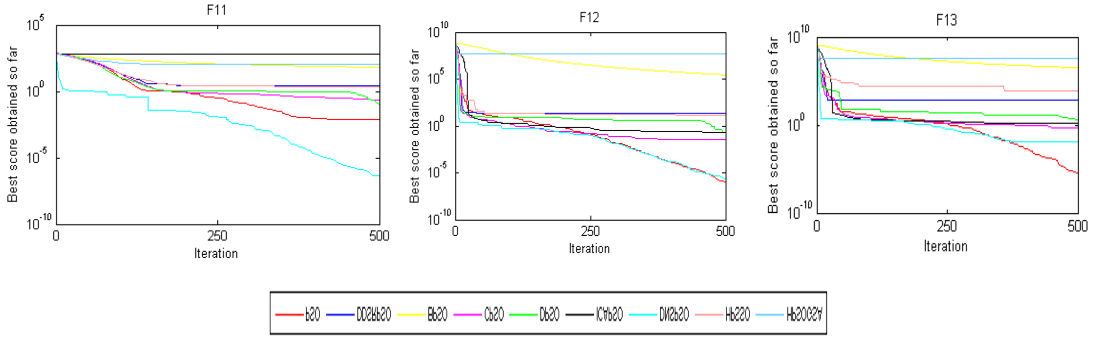
4.2 *Evaluation of diversification/exploration capability for 30-D multimodal functions* $(f_8 - f_{14})$

Apart from unimodal functions, multimodal functions have increase in number of design variables; this may lead to massive number of local optimum solutions. These kinds of test problems are very useful to evaluate the exploration capability of an optimization algorithm. Function $f_8 - f_{13}$ (i.e. multi-modal functions) results are stated in table 5 and the convergence rates of the same were presented in figure 5.

For function f_8 all the algorithms were performs well and in function f_9 ICAPSO and DNSPSO performs better when it compares with other variants. ICAPSO perform well in function f_{10} whereas other algorithms are gradually finding the minimum solution which is shown in figure 5.c. In function f_{11}, all the variants of PSO perform better after a course of iterations; apart from that DNSPSO perform well in convergence rate at the initial state of iterations. Beside function f_{12} and f_{13}, all the algorithms better well interms of convergence rate but generic PSO, BPSO, CPSO and HPSSO performs better interms of computation time in figure 7. Overall results of multimodal functions say that every algorithm performs well in finding good optimum solutions over 30 independent runs but when compare to convergence rate ICAPSO, DNSPSO, Generic PSO, CPSO performs better.

Figure 5. Convergence rate comparison of Generic PSO and its eight variants obtained in Multiimodal benchmark problems (a) Schwefel problem 2.26, (b) Rastrigin function, (c) Ackley function 1.2, (d) Griewank function, (e) penalized function 1, (f) penalized function 2

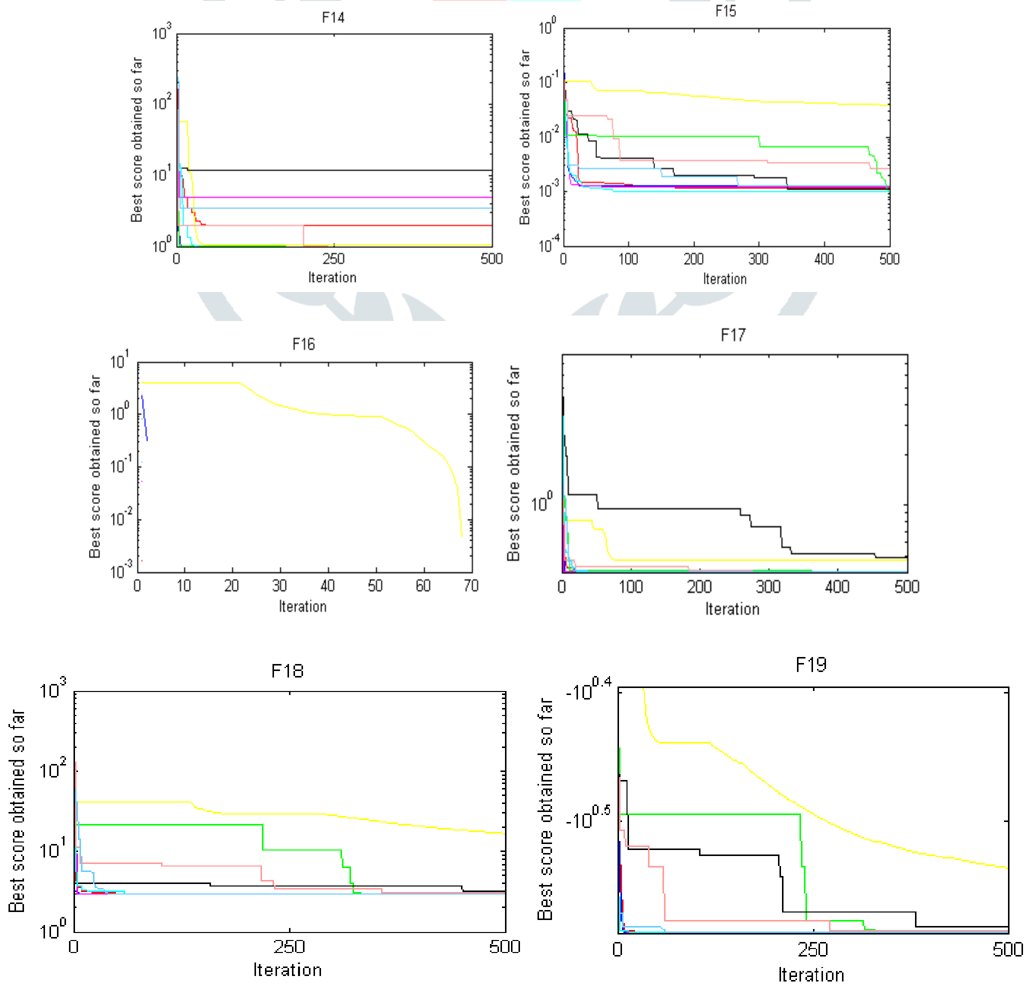


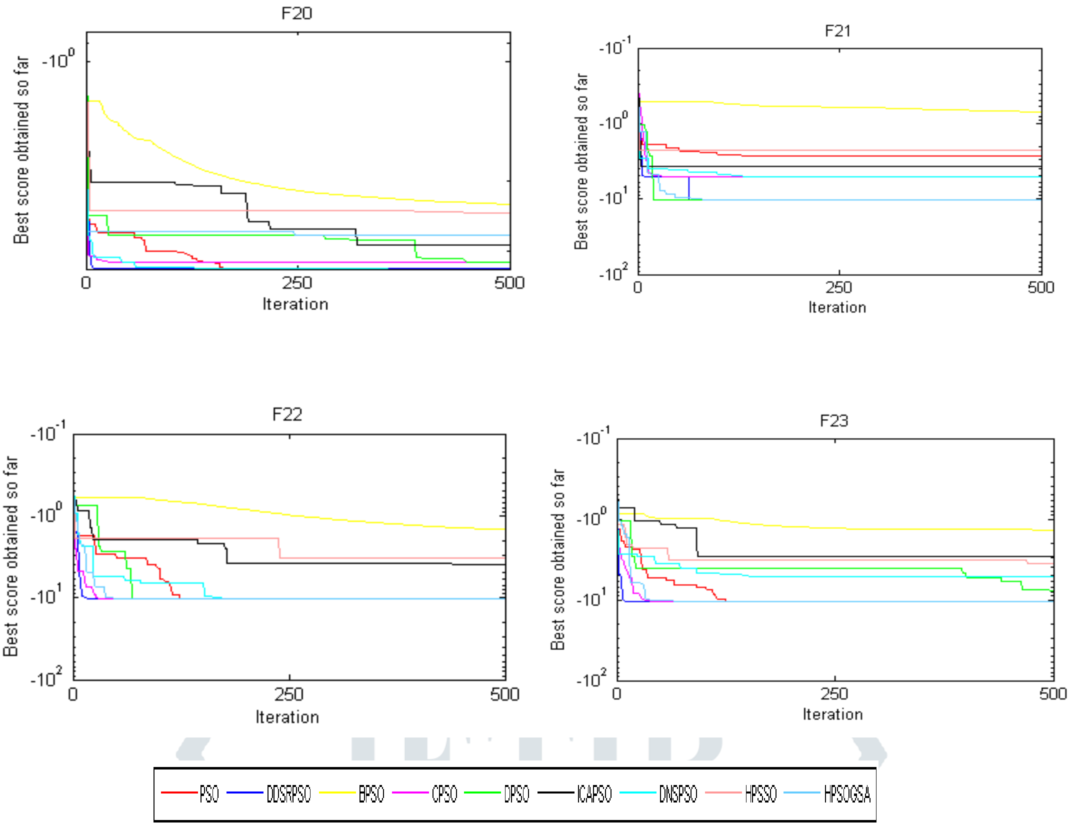


4.3 *Evaluation of diversification/exploration capability for fixed dimension multimodal functions* ($f_{14} - f_{23}$)

According to (Berg et al., 2008), movement of search agents perform abrupt changes over the initial stage of optimization. This assists the meta-heuristic to explore broadly within the search space. At the final stage of optimization these changes should be minimized to highlight exploitation. It was noticed that variants of PSO algorithms search particles incline to broad search promising areas of objective space and achieved the best one. Search particles adopt sharply towards the best optimal point in the early stages of the optimization process and then leisurely converge. Population based algorithm can guarantee eventually convergences to a best optimal point in a search space. Fixed dimension multimodal functions $f_{14} - f_{23}$ have low dimensions (2, 4, or 6), in that some of them are difficult to optimize. Based on the convergence graph and results ICAPSO not perform well when it compares to other algorithms Function f_{14} whereas the same algorithm has moderate convergence in a course of iterations for functionf_{15}. Similarly BPSO converge slowly in function f_{15}over the entire iterations. We identifies that function $f_{15}$is difficult to converge near to optimum solutions, but expect BPSO all other algorithms perform better and achieves solutions near to optimum.

Figure 6. Convergence rate comparison of Generic PSO and its eight variants obtained in fixed dimensional Multimodal benchmark problems (a) Shekel's Foxholes, (b) Kowalik function, (c) Six - hump Camel function, (d) Brainin-Hoo function, (e) Goldstein-Price function, (f) 3-dimensional Harmann function, (g) 6-dimensional Harmann function, (h) Shekel function 5, (i) Shekel function 7, (j) Shekel function 10

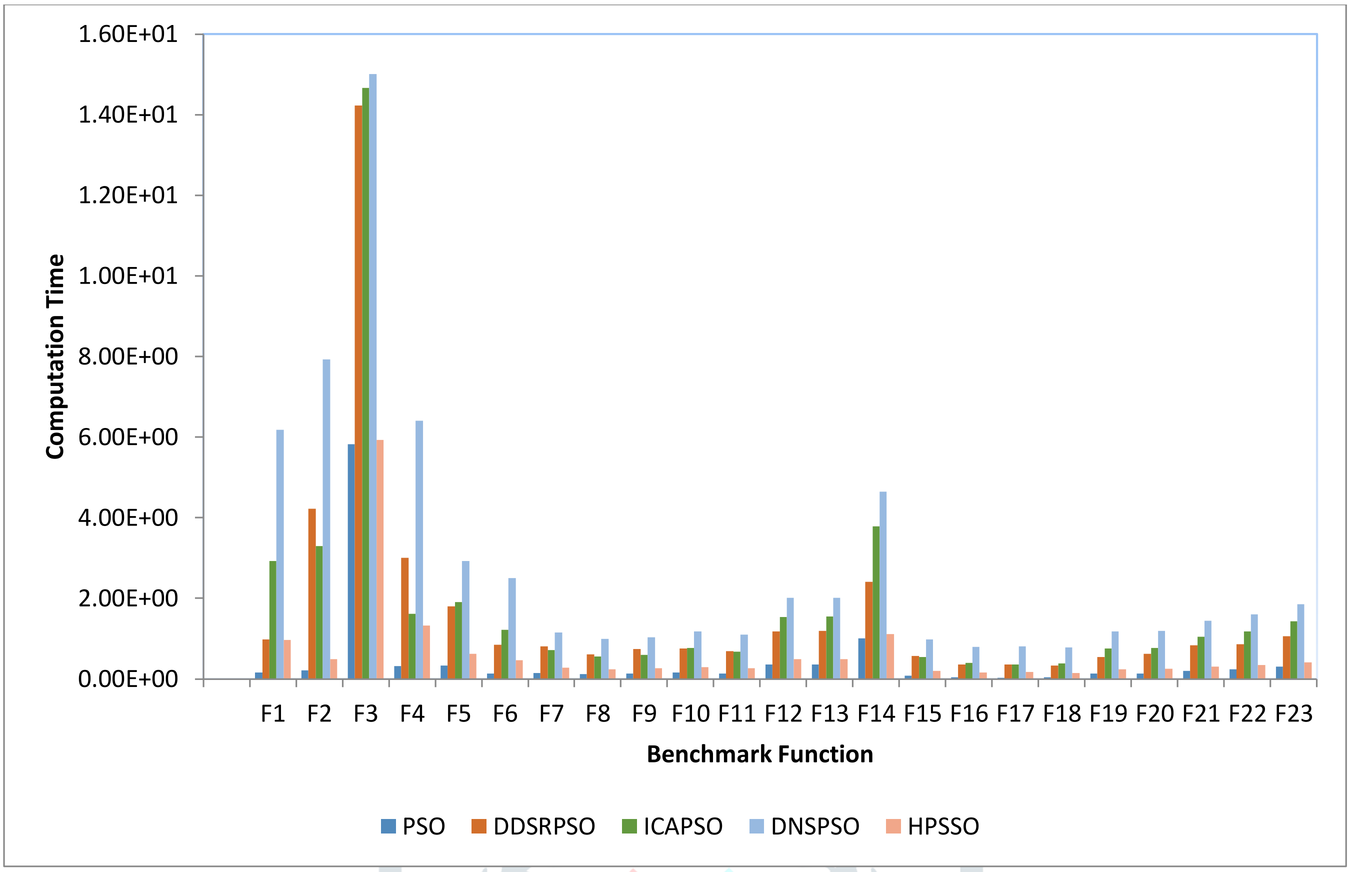




In function f_{16} expect BPSO, all other algorithms identifies the optimum solution when it reaches 20th iterations, but the BPSO converges slowly to optimum which exploit depth into the local solutions. For f_{17} compare to BPSO, ICAPSO converge very slowly near to the optimum solutions and remaining all other algorithms perform well to find good optimum solutions. The experimentation of function f_{18} provides that BPSO fails to identify the optimum within the specified iterations.

In function f19, we note that DPSO, ICAPSO, HPSSO particles starts its convergence in iteration 20 and withstands in the same positions for course iterations and later it moves on converging, Whereas BPSO converges moderately to the best positions. Similarly, function f20, f21, all the variants performs well interms of convergence, whereas BSPSO has slower convergence rate. Based on our experiments, we identified that the BPSO suits for photo crystal based problems, in some cases it fails to attain the convergence rate. However, tuning BPSO parameter may increase its convergence and cope up together as other algorithms have been performed. Finally, we state the function f22, f23 DDSRPSO begins its finding and attain the optimum in initial state iterations, whereas other algorithms starts it's finding process moves towards the optimum in course of iterations.

Figure 7. Computation time Analysis for PSO variants with benchmark test functions



Analysis of Computational time

The performance of discrete variants of PSO algorithms is validated using the statistical analysis of computation time shown in figure 7. The CPU clock time is used as the computation time requirements for all the PSO variants. Clearly, it shows that the performance of Generic PSO takes minimum computation time than all PSO variants. But, this algorithm may trap in local optima and provides similar solution in a certain number of iterations. DD-SRPSO achieves minimum computation time than all other discrete variants of PSO in some set of functions and also converges nearer to the optimum solutions. BPSO repeatedly provides a better result for both some set of multi-modal and fixed dimension multimodal function in minimum computation time. CPSO and HPSSO algorithm provide better results in gradient set of function; whereas for some set of benchmark functions it takes more computation time to converge. DPSO and HPSOGSA takes more computation time to get converge from local optima to global optimum solutions. Imperialist Competitive Particle Swarm Optimization (ICA-PSO) grant better results within less computation time in all set of benchmark functions. In addition with crowding distance is incorporated in order to sustain exploration within the archive. From the overall observations, Imperialist Competitive Particle swarm Optimization and Diversity Enhanced Particle Swarm optimization provides better results based on the computation time for unimodal, multimodal and multidimensional benchmark functions.

V. CONCLUSIONS

In this paper, the generic Particle Swarm Optimization (PSO) with its novel variants presented in the literature is discussed. The state-of-the-art eight different PSO variants are considered here, in which exploration and exploitation plays a vital role to improve the generic PSO. The performance of these algorithms are analyzed using three different group of benchmark test functions viz., unimodal, multimodal and fixed dimension multimodal functions. The statistical analyses (average, standard deviation and computation time) are measured and observed results shows that each algorithm performs better efficiency in different set of benchmark functions. From this survey, we identified that PSO has some disadvantages such as minimum convergence, premature, high computational complexity and so forth. Mainly two reasons behind the disadvantages of PSO they are: Initially, PSO does not hold any novel operators (like crossover or mutation) as applied in GA or DE; hence the sharing of good information between the particles is not at an expected level. Another disadvantage may fall within the fact that PSO does not handle the balance between the exploration and exploitation, so it quickly converges to a local minimum. The major variants, including Bisection PSO, Chaotic PSO, DDSRPSO, Dispersed PSO, Compact PSO, ICAPSO, DNSPSO, HPSSO and HPSOGSA are good in balancing exploration and exploitation capabilities as well as overcome the disadvantages of generic PSO. From these variants we evaluated the performance with convergence rate and computation time analysis. Based on our experimentation, we conclude that ICAPSO and DNSPSO performed well in identifying optimum solution within a reasonable time and has better convergence rate.

REFERENCES

[1] A. Kaveh, T. Bakhshpoori, E. Afshari, An efficient hybrid Particle Swarm and Swallow Swarm Optimization algorithm, Computers & Structures, 2014; 143: 40-59.

[2] A. RezaeeJordehi, "Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimization problems", Applied Soft Computing, 2015.

[3] B. Y. Qu, Ponnuthurai Nagaratnam Suganthan, Swagatam Das, A Distance-Based Locally Informed Particle Swarm Model for Multimodal Optimization, IEEE Transactions On Evolutionary Computation, 3,2013.

[4] Basturk B, Karaboga D. An artificial bee colony (ABC) algorithm for numeric function optimization. In: IEEE swarm intelligence symposium; 2006. p. 12–4.

[5] Beni G, Wang J. Swarm intelligence in cellular robotic systems. In: Robots and biological systems: towards a new bionics?, ed. Springer; 1993. p. 703–12.

[6] Bin Jiang, Anjin Liu, Wei Chen, Mingxin Xing, wenjun Zhou, wanhuaZheng, "The optimization of large gap–midgap ratio photonic crystal with improved Bisection-Particle Swarm Optimization", Journal of Optics Communication, 2011; 284:226-230.

[7] Bo Xing , Wen-Jing Gao, Imperialist Competitive Algorithm, Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms, 2013; 62: 203-209.

[8] Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: from natural to artificial systems: OUP USA; 1999.

[9] Chun-Feng Wang, Kui Liu, A Novel Particle Swarm Optimization Algorithm for Global Optimization, Computational Intelligence and Neuroscience, 2016

[10] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. Computation Intelligence Magazine, IEEE, Vol.1 p. 28–39, 2006.

[11] Esmat Rashedi, Hossein Nezamabadi-pour, Saeid Saryazdi, GSA: A Gravitational Search Algorithm, Information Sciences, 2009; 179:2232-2248.

[12] F. Yano, T. Shohdohji, Y. Toyoda, An improvement of particle swarm optimization with a neighborhood search algorithm, Industrial Engineering and Management Systems, 2007

[13] FerranteNeri, Ernesto Mininno, Giovanni Lacca, " Compact Particle Swarm Optimization", Journal of Information Sciences, 2013; pp. 96-121.

[14] G. Chen, X. Huang, J. Jia, and Z. Min., "Natural exponential Inertia Weight strategy in particle swarm optimization", In Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, volume 1, pages 3672–3675. IEEE, 2006.

[15] H.R. Li and Y.L. Gao, "Particle Swarm Optimization Algorithm with Exponent Decreasing Inertia Weight and Stochastic Mutation", In ―2009 Second International Conference on Information and Computing Science, pages 66–69. IEEE, 2009.

[16] Haiping Yu, Xueyan Li, High Dimensional Problem Based on Elite-Grouped Adaptive Particle Swarm Optimization, In. proceedings of Springer, 2013.

[17] Han Huang, Hu Qin, ZhifengHao, Andrew Lim, "Example-based learning particle swarm optimization for continuous optimization", Information Sciences, 2012.

[18] Harish Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Applied Mathematics and computation, 2016

[19] Hui Wang, Hui Sun, Changhe Li, Shahryar Rahnamayan, Jeng-shyang Pan, Diversity enhanced particle swarm optimization with neighborhood search, Information Sciences, 2013; 223: 119-135.

[20] I.G. Tsoulos and A. Stavrakoudis (2010), Enhancing PSO methods for global optimization, Applied Mathematics and Computation, 2010.

[21] J. J. Liang, A. K. Qin, Ponnuthurai Nagaratnam Suganthan, S. Baskar, Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions, IEEE Transactions on Evolutionary Computation, p.281-295, 2006.

[22] J. Kennedy, R. , Population structure and particle swarm performance, in: Proceedings of IEEE Congress on Evolutionary Computation, Vol. 02, pages. 1671-1676, May 2002

[23] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, in: Proceedings of IEEE Congress on Evolutionary Computation, 1999.

[24] J. van den Berg, et al., "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation", IEEE International Conference on Robotics and Automation, 2008, pp. 1928-1935.

[25] J. Xin, G. Chen, and Y. Hai., "A Particle Swarm Optimizer with Multistage Linearly-Decreasing Inertia Weight", In Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on, volume 1, pages 505–508. IEEE, 2009.

[26] Jianxin Liao, Yang Liu, Xiaomin Zhu, Jingyu Wang, "Accurate sub-swarms particle swarm optimization algorithm for service composition Accurate sub-swarms particle swarm optimization algorithm for service composition", The Journal of Systems and Software, 2014

[27] K. Kentzoglanakis and M. Poole., "Particle swarm optimization with an oscillating Inertia Weight", In Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pages 1749– 1750. ACM, 2009.

[28] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y. P. Chen, C. M. Chen, and Z. Yang, "Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization," Technical Report, Nature Inspired Computation and Applications Laboratory, USTC, China, http://nical.ustc.edu.cn/cec08ss.php, November 2007.

[29] Kamal Boudjelaba, Frédéric RosDjamel Chikouche, Potential of Particle Swarm Optimization and Genetic Algorithms for FIR Filter Design, Circuits, Systems, and Signal Processing, 2014; 33:3195-3222.

[30] Kennedy J, Eberhart R. Particle swarm optimization, in Neural Networks, In: Proceedings, IEEE international conference on; 1995. p. 1942–1948

[31] Lhassane Idoumghar, Nicolas Chérin, Patrick Siarry, Robin Roche, Abdellatif Miraoui, Hybrid ICA–PSO algorithm for continuous optimization, Applied Mathematics and Computation, 2013

[32] Lhassane Idoumghar, Nicolas Chérin, Patrick Siarry, Robin Roche, Abdellatif Miraoui, Hybrid ICA–PSO algorithm for continuous optimization, Applied Mathematics and Computation, 2013; 219: 11149-11170.

[33] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation, 2002

[34] M.R. Tanweer, R. Auditya, , S. Suresh, N. Sundararajan, , N. Srikanth, Directionally Driven Self-Regulating Particle Swarm Optimization algorithm, Swarm and Evolutionary Computation, 2016; 28:98-116.

[35] M.R. Tanweer, S. Suresh , N. Sundararajan, Self regulating particle swarm optimization algorithm, Information Sciences, 2015; 294:182-202.

[36] Momin Jamil and Xin-She Yang, "A literature survey of benchmark functions for global optimization problems", Int. Journal of Mathematical Modelling and Numerical Optimisation, Vol. 4, No. 2, pp. 150–194 (2013).

[37] Nitasha Soni, Dr .Tapas Kumar, Study of Various Crossover Operators in Genetic Algorithms, Int. Journal of Computer Science and Information Technologies, 2014;5:7235-7238.

[38] Padmavathi Kora, Sri Ramakrishna Kalva, Hybrid Bacterial Foraging and Particle Swarm Optimization for detecting Bundle Branch Block, Springer Plus, 2015

[39] Qu, B., Liang, J., & Suganthan, Niching particle swarm optimization with local search for multi-modal optimization. Information Sciences, 197, 131–143, 2012

[40] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Transactions on Evolutionary Computation, 2004

[41] R.F. Malik, T.A. Rahman, S.Z.M. Hashim, and R. Ngah, "New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight", International Journal of Computer Science and Security (IJCSS), 1(2):35, 2007.

[42] Ran Chenga, Yaochu Jin, A social learning particle swarm optimization algorithm for scalable optimization, Information Sciences, 2015.

[43] Rui Mendes,James Kennedy, José Neves, The Fully Informed Particle Swarm: Simpler, Maybe Better, IEEE Transactions on Evolutionary Computation, p.204-210, 2004.

[44] S.M. Abd-Elazim, E.S. Ali, A hybrid Particle Swarm Optimization and Bacterial Foraging for optimal Power System Stabilizers design, International Journal of Electrical Power & Energy Systems, 2013; 46:334-341.

[45] Seyedali Mirjalili, Andrew Lewis, The Whale Optimization Algorithm, Advances in Engineering Software, 2016; 95:51-67.

[46] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis, Grey Wolf Optimizer, Advances in Engineering Software, 2014; p. 46-61.

[47] Seyedali Mirjalili, Siti Zaiton Mohd Hashim, A New Hybrid PSOGSA Algorithm for Function Optimization, In. Proceedings on Computer and Information Application, 2010

[48] Shanhe Jiang, Zhicheng Ji, Yanxia Shen, A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints, International Journal of Electrical Power & Energy Systems, 2014; 55:628-644.

[49] Shanhe Jiang, Zhicheng Ji, Yanxia Shen, A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints, International Journal of Electrical Power & Energy Systems, 2014; 55: 628-644.

[50] T. Peram, K. Veeramachaneni, C.K. Mchan, Fitness-distance-ratio based particle swarm optimization, in: Proceedings of the IEEE Swarm Intelligence Symposium, 2003

[51] Xingjuan Cai, Zhihua Cui, Jianchao Zeng, Ying Tan, " Dispersed Particle Swarm Optimization", In Proceedings of Information Processing Letters, 2008; 105: 231-235.

[52] Y. Feng, G.F. Teng, A.X. Wang, and Y.M. Yao., "Chaotic Inertia Weight in Particle Swarm Optimization", In Innovative Computing, Information and Control, 2007.ICICIC'07. Second International Conference on, page 475. IEEE, 2008.

[53] Y. Gao, X. An, and J. Liu., "A Particle Swarm Optimization Algorithm with Logarithm Decreasing Inertia Weight and Chaos Mutation", In Computational Intelligence and Security, 2008.CIS'08. International Conference on, volume 1, pages 61–65. IEEE, 2008

[54] Y. Shi and R. Eberhart., "A modified particle swarm optimizer", In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, pages 69–73. IEEE, 2002.

[55] Yang Shi, Hongcheng Liu, Liang Gao, Guohui Zhang, "Cellular particle swarm optimization", Information Sciences, 2011.

[56] Yue-jiao Gong, Jun Zhang, Small-World Particle Swarm Optimization with Topology Adaptation, In. Proceedins GECCO, 2013

[57] Zahra Beheshti, Siti Mariyam Shamsuddin, Sarina Sulaiman, Fusion Global-Local-Topology Particle Swarm Optimization for Global Optimization Problems, Mathematical problems in Engineering, 2014

[58] Zhi-Hui Zhan, Jun Zhang, Yun Li, Yu-Hui Shi, Orthogonal Learning Particle Swarm Optimization, IEEE Transactions on Evolutionary Computation, 2011