Neural Computing and Applications (2019) 31:7665–7683
https://doi.org/10.1007/s00521-018-3592-0

ORIGINAL ARTICLE

On the exploration and exploitation in popular swarm-based metaheuristic algorithms

Kashif Hussain[1] · Mohd Najib Mohd Salleh[1] · Shi Cheng[2] · Yuhui Shi[3]

Received: 19 September 2017 / Accepted: 23 June 2018 / Published online: 3 July 2018

Abstract

It is obvious from wider spectrum of successful applications that metaheuristic algorithms are potential solutions to hard optimization problems. Among such algorithms are swarm-based methods like particle swarm optimization and ant colony optimization which are increasingly attracting new researchers. Despite popularity, the core questions on performance issues are still partially answered due to limited insightful analyses. Mere investigation and comparison of end results may not reveal the reasons behind poor or better performance. This study, therefore, performed in-depth empirical analysis by quantitatively analyzing exploration and exploitation of five swarm-based metaheuristic algorithms. The analysis unearthed explanations the way algorithms performed on numerical problems as well as on real-world application of classification using adaptive neuro-fuzzy inference system (ANFIS) trained by selected metaheuristics. The outcome of empirical study suggested that coherence and consistency in the swarm individuals throughout iterations is the key to success in swarm-based metaheuristic algorithms. The analytical approach adopted in this study may be employed to perform component-wise diversity analysis so that the contribution of each component on performance may be determined for devising efficient search strategies.

Keywords Swarm intelligence · Metaheuristic · Population diversity · Exploration and exploitation · Optimization

1 Introduction

Despite complexity, nonlinearity and high-level dimensions of real-world optimization problems, contrary to deterministic and statistical methods, metaheuristic algorithms reach near optimal solutions with reasonable time and resources. Additionally, ease of implementation and efficiency has led to wider spectrum applications in science and engineering optimization problems (see for sample [1–4], and [5]). Such successful interventions of metaheuristic algorithms in solving hard optimization problems motivated researchers to develop more and more of such algorithms, inspired from natural as well as man-made processes. Moreover, metaheuristic algorithms based on swarm intelligence are gaining more popularity among researchers as compared to other population-based counterparts [1, 6]. Thanks to landmark particle swarm optimization (PSO) [7] and ant colony optimization (ACO) [8] which derived the addition of adequately increasing number of swarm-based metaheuristic algorithms—not necessarily are all efficient methods hence not achieved generous acceptance in metaheuristic community. According to a limited survey of publications related to swarm-based metaheuristics between 1995 and 2016, there exist more than fifty swarm-based metaheuristic algorithms, out of which top ten are listed in Fig. 1.

As it is depicted from the bar chart in Fig. 1, the landmark PSO beats rest of the algorithms in number of publications due to simplicity and ease of implementation. The rest of the algorithms are artificial bee colony (ABC) [9], ant colony optimization (ACO) [8], cuckoo search (CS) [10], firefly algorithm (FA) [11], fireworks algorithm (FWA) [12], bat algorithm (BA) [13], teaching-learning

✉ Mohd Najib Mohd Salleh
najib@uthm.edu.my

[1] Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, Malaysia

[2] School of Computer Science, Shaanxi Normal University, Xi'an, China

[3] Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

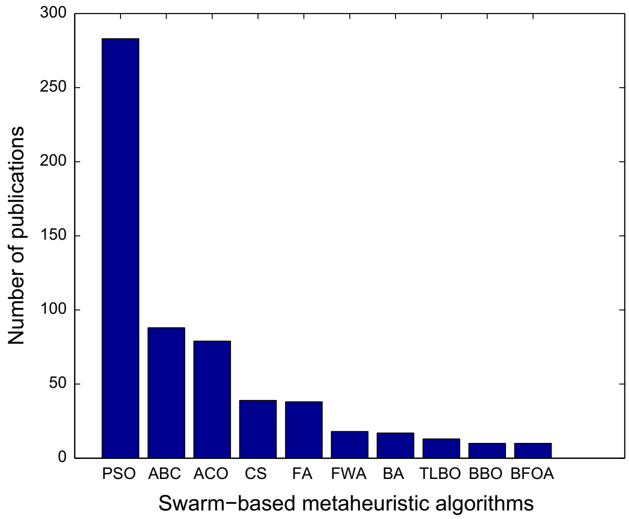


Fig. 1 Popular swarm-based metaheuristic algorithms based on number of publications

based optimization (TLBO) [14], biogeography-based optimization (BBO) [15], bacterial foraging algorithm (BFOA) [16]. Yudong et al. in a survey [17] also found these algorithms popular in literature. Moreover, according to comprehensive surveys [17–22] found in recent literature, the top five swarm-based algorithms (PSO, ABC, ACO, CS, and FA) have successfully solved hard optimization problems due to efficient search ability, ease of implementation, and robustness of results. The applications of these algorithms cover wider range of domains including science, engineering, medical, business, data science, etc.

Apart from wider acceptance due to efficient results, it is still relatively unknown how and why these algorithms perform better than one or the other algorithm on one or the other optimization problem. This leads to a remarkable research to be performed on open questions raised by critics, such as [23–25]. These concerns are beyond the justification often provided based on "no-free lunch" theorem; even though it is true due to stochastic nature of algorithms [26], more theoretical and practical explanations are required. However, the questions on convergence and performance analyses are repeatedly answered, in the literature, with the help of convergence graphs and statistics (mean, best, worst, and standard deviation, etc.) obtained over a certain number of runs. This may reveal '*what happened*,' but '*how and why it happened*' is related to more in-depth analyses of how efficiently the swarm individuals explore a search space. Hence, this study took top five (Fig. 1) swarm-based metaheuristic algorithm to examine the behavior of swarm individuals in terms of diversity. Through diversity measurement, we gauged explorative and exploitative abilities of the algorithms. Moreover, this paper provides extensive in-depth analysis and discussion on components effecting exploration and exploitation of the swarm-based metaheuristics. For examining the efficiency, five commonly used benchmark numerical problems were utilized. To further investigate performances on real-world applications, the algorithms were employed on training the parameters of adaptive neuro-fuzzy inference system (ANFIS) for solving the problem of classifying Small Medium Enterprises (SMEs) based on strength.

Overall, the contribution of this study is to propose an approach to measure and quantitatively analyze the level of exploration and exploitation in a metaheuristic algorithm while solving certain optimization problem. The approach may help maintain trade-off balance between exploration and exploitation ratios in a metaheuristic algorithm. The measurement of exploration and exploitation also helps understand why certain metaheuristic algorithm performed poor or better on an optimization problem.

The remainder of the paper is structured as follows. The subsequent section (Sect. 2) gives brief introduction of swarm-based metaheuristic algorithms of this study. Section 3 explains the method to measure exploration and exploitation based on diversity in swarm. This section also briefs about numerical optimization problems and SME classification problem. Section 4 reports experimental results, followed by discussion and in-depth analyses in Sect. 5. Lastly, conclusions and future works are provided toward the end in Sect. 6.

2 Swarm intelligence

The exceptional features of collective intelligence of various swarm behaviors in nature have been adopted to design a range of optimization algorithms. Such features are mainly related to how swarm individuals communicate in order to reach the best food source with collective decision. These decentralized individuals perform search based on their own personal cognition or experience, as well as, information available globally among all the individuals. The source of information exchange is pheromone in case of ants, sound waves in bats, waggle dance in bees, etc. Apart from essential communication behaviors in nature, researchers also implanted other intelligence mechanism to develop better and better optimization algorithms with explorative and exploitative capabilities. This study considered top five swarm-based metaheuristics according to Fig. 1. Following is given a short introduction of each of these algorithms, while the reader is encouraged to refer to the cited literature for extended details, as the focus of this study is purely on performance analyses.

2.1 Particle swarm optimization (PSO)

PSO [7] uses particles, representing a flock of birds or school of fish, to observe the search environment based on cognitive and social intelligence for searching the best food location. Particles in PSO have velocity and position. The next move is decided based on current position and new velocity calculated with respect to personal best position and the globally best particle's position, see (1) and (2).

$$v_i^{t+1} = \omega v_i^t + c_1 R_1 (p_i^{best} - x_i^t) + c_2 R_2 (x^{gbest} - x_i^t) \tag{1}$$

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

In Eq. (1), v_i^{t+1} is velocity vector for next iteration $t+1, \omega$ is inertia weight which controls velocity and allows swarm to converge in later iterations. v_i^t and p_i^{best} are the current velocity and position of ith particle. x^{gbest} is the best position the whole swarm found so far. c_1 and c_2 are cognitive and social factors to control the added randomness to the velocity for next move at position x_i^{t+1}, whereas R_1 and R_2 are the two different random vectors. For the balanced exploration and exploitation, inertia weight is crucial among other parameters of PSO algorithm. In (2), the next position x_i^{t+1} of ith particle is computed using current position x_i^t and the velocity vector v_i^{t+1} generated in (1). Here, vector x_i represents a solution and vector v_i represents momentum of a particle.

2.2 Artificial bee colony (ABC)

ABC [9] is inspired from the swarm behavior of honey bees that fly in search of the location, with best flower patch, where they can maximize the collection of nectar from. The swarm in this algorithm is divided into three types of individuals: employed bees, onlooker bees, and scout bees. Employed bees are the first to sightsee and discover food sources, followed by onlooker bees which pursue the potential locations shared by employed bees. This is based on probability (ρ_i) of selection of employed bees calculated as (3):

$$\rho_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{3}$$

where fit_i is nectar amount (objective function value) collected from ith food source and SN is the total number of food sources. Roulette wheel selection method is used on employed bees' probability values. The new location of onlooker bee is calculated using (4):

$$x_i^{\text{new}} = x_i + R_i(x_i - x_j) \tag{4}$$

where x_i is the employed bee's current location, x_j is randomly chosen bee j other than i, and R_i is the randomness added to the new location x_i^{new}. After certain number of attempts (defined by parameter *Limit*), when some of the bees are unable to find any improved food source, scout bees invoke and replace them to try random places using (5):

$$x_k = lb + R_k(ub - lb) \tag{5}$$

where x_k is the kth scout bee, ub and ub are the upper and lower bounds of the problem domain, and R_k is random $(-1, 1)$ number generated for kth bee. After each iteration, employed bees search neighborhoods of the previously found potential locations using (4), but in this case, x_i is the previous food source, x_j is randomly selected food source other than x_i from previous iteration.

2.3 Ant colony optimization (ACO)

ACO [8] metaphorizes the foraging behavior of social ants that use pheromone as a tool of communication. When returning from food source, ants deposit certain amount of pheromone along the path indicating the suitability of the food source just visited. The most suitable path for other ants to follow is the shortest one with maximum pheromone representing optimum food source. The concentration of the pheromone is time dependent, as it evaporates gradually. Initially, m ants search food source randomly (using same equation as (5)) and while returning deposit pheromone (objective function value) along the path which is, later on, gauged and reinforced by other ants through further dumping pheromone, see (6) below:

$$\tau_{ij}(t) = \rho\tau_{ij}(t-1) + \Delta\tau_{ij}; it = 1, 2, \ldots, MaxItr \tag{6}$$

where $\tau_{ij}(itr), \rho, MaxItr$, and $\Delta\tau_{ij}$ are revised pheromone concentration, pheromone evaporation rate, maximum number of iterations, and change in pheromone concentration, respectively. The change in pheromone is calculated using (7):

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \begin{cases} R/fit_k & \text{if } l_{ij} \text{ is chosen by ant } k \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where R and fit_k are pheromone reward factor and objective function value of kth ant. As the iterations proceed, the pheromone deposited along the path evaporates, which allows ants to avoid premature convergence. Once the pheromone value is updated with each path, next iteration changes path of the ants moving in succeeding iteration using (8):

$$P_{ij}(k,t) = \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta}{\sum_{l_{ij}} [\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta} \tag{8}$$

where $P_{ij}(k,t)$ is the path chosen by kth ant for iteration $t, \tau_{ij}(t)$ denotes pheromone concentration level at the chosen path, η_{ij} is heuristic value assigned solution indicating feasibility for kth ant to select. Parameters α and β influence pheromone concentration and heuristic information. Since ACO is mainly designed for combinatorial optimization problems, we chose the suitable variant $\text{ACO}_{\mathbb{R}}$ [27] for solving continuous optimization problems in this work.

2.4 Cuckoo search (CS)

CS [10] algorithm follows the way Cuckoo birds manipulate hosts by resembling eggs with the ones in the host's nest. The eggs with maximum survival rate hatch successfully and carry to next generation, whereas the poor eggs are destroyed by the host bird. The algorithm starts with initial random solutions in terms host nests where cuckoos lay eggs. Each habitat has fitness value representing suitability of eggs to survive. CS defines eggs laying radius (ELR) by (9):

$$\text{ELR} = \alpha \times \frac{C_{\text{eggs}}}{N_{\text{eggs}}} \times (ub - lb) \tag{9}$$

where $\alpha, C_{\text{eggs}}, N_{\text{eggs}}, ub$, and lb are constant that controls radius, number of cuckoo's eggs, total eggs, upper bound, and lower bound respectively. After laying new eggs in randomly chosen host nests in the predefined radius, certain percentage of eggs with worst fitness value are destroyed. CS uses lévy flight random walk to decide for the next move, using (10):

$$x_i^{(t+1)} = x_i^t + \alpha \oplus Lévy(\lambda) \tag{10}$$

where α is step size, $1 < \lambda < = 3$, and $\oplus$ is entry-wise multiplication. There is only one parameter which is discovery rate ρ of poor eggs to be destroyed and replaced with new ones.

2.5 Firefly algorithm (FA)

FA [11] mimics the flashing pattern of fireflies to communicate and attract other fireflies. The brighter the better is the firefly to attract others, as light intensity represents fitness value. The light intensity increases and decreases with respect to distance from other fireflies. The algorithm starts with initial random population generated by (5) and light intensity calculated using (11):

$$I = I_0 e^{-\gamma r^2}, \beta = \beta_0 e^{-\gamma r^2} \tag{11}$$

where I_0, r, and γ are the original light intensity, distance, rand absorption parameter. With light intensity calculated, FA calculates attractiveness feature using (11), where β_0 is the initial attractiveness. The new location x_i^{new} generated by movement of firefly x_i to firefly x_j is calculated using (12), where R_i is random number and α is step size.

$$x_i^{\text{new}} = xi + \beta_0 e^{-\gamma r^2}(x_j - x_i) + \alpha R_i \tag{12}$$

Since the fundamental understanding has been established about the algorithms, the subsequent section explains the methodology adopted in this work for measuring exploration and exploitation of the swarm-based metaheuristic algorithms.

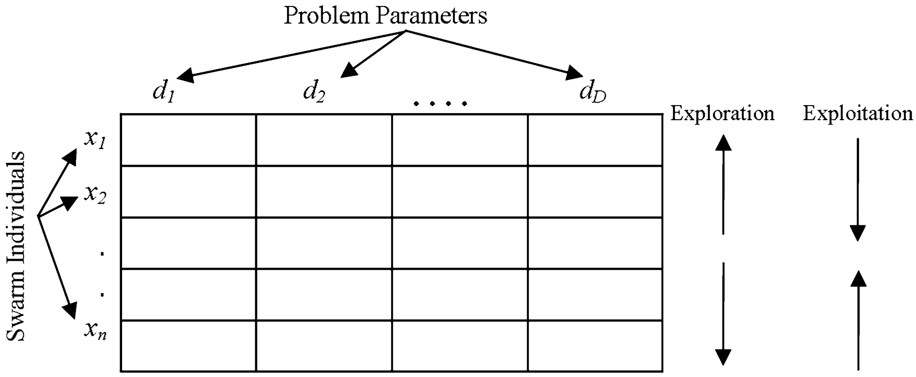
3 Methodology

A hard optimization problem poses substantial amount of available solutions. Finding the optimum (nearly best) solution demands a swarm-based metaheuristic algorithm to drive swarm individuals so efficiently to draw effective search of the environment. This requires diversified and dynamic moves to the promising regions without wastefully consuming time and resources. To determine effectiveness of the selected algorithms, this study measured the two performance cornerstones, exploration and exploitation, on numerical optimization problems and a real-world application. For numerical problems, commonly used benchmark test functions with different modalities were employed, and for the later, we solved SME classification problem using adaptive neuro-fuzzy inference system (ANFIS) trained by selected metaheuristic algorithms. This section explains, in detail, the three empirical components of this study: exploration and exploitation measurement, simulations on test functions, and application on SME classification problem.

3.1 Exploration and exploitation measurement

A swarm individual, say $x_i, i \in \{1, 2, 3, \ldots, n\}, n =$ swarm size, is a D dimensional vector that represents parameter values to be optimized for the optimization problem in hand (for example Sphere, Ackley, etc.). As depicted via Fig. 2, the difference between dimensions of individuals infers if the swarm is diverging or clustering in a concentrated space. When the algorithm is diverging, the difference between the values of dimension d within swarm individuals enlarges, meaning that swarm individuals are scattered in the search environment. This is referred to as exploration or diversification in metaheuristic research. On the other hand, when the swarm is converging, the difference minimizes and swarm individuals gather to a condensed area. This is called exploitation or intensification. During the course of iterations, different metaheuristic algorithms employ different strategies to enforce diversification and intensification within the swarm individuals.

Fig. 2 $n \times D$ dimensional representation of swarm



These two concepts are omnipresent in any metaheuristic algorithm. Through exploration, an algorithm is able to visit unseen neighborhoods in the search environment in order to maximize efficiency of finding the globally optimal location. Contrarily, exploitation allows swarm individuals to successfully converge to a potential neighborhood with highly likelihood of global best solution. The balance between the two abilities is a trade-off problem. Algorithms with poor in both abilities fail to produce effective results. Hence, the search philosophy of any swarm-based algorithm is crucial to its performance. Therefore, it is imperative to measure exploration and exploitation of an algorithm so that the search strategies influencing these two factors may be analyzed practically.

As mentioned earlier, studying convergence graph and mean, best, worst, standard deviation of the solutions found over certain number of runs does not help understand the insights of search behavior; it is therefore, such end results still leave open questions about the performance efficiency of a metaheuristic algorithm. That said, for swarm-based metaheuristic algorithms, it is significantly important to analyze the behavior of each individual in a swarm, as well as, swarm as a whole. This motivated the current research to adopt dimension-wise diversity measurement proposed by [28] with modification where mean is replaced with median in (13); as it reflects the center of the population more effectively.

$$Div_j = \frac{1}{n}\sum_{i=1}^{n} \text{median}(x^j) - x_i^j; \qquad Div = \frac{1}{D}\sum_{j=1}^{D} Div_j \tag{13}$$

where $median(x^j)$ is median of dimension j in whole swarm, whereas x_i^j is the dimension j of swam individual i, and n is the size of swarm. After taking dimension-wise distance of each swarm-individual i from the median of dimension j, we take average Div_j for all the individuals. Later on, the average of diversity of all dimensions is computed in Div.

Once diversity of swarm has been captured for each iteration, it is now possible to determine the percentage of exploration and exploitation in an algorithm for each iteration using (14):

$$\text{Xpl}\% = \frac{Div}{Div_{\max}} \times 100; \qquad \text{Xpt}\% = \frac{|Div - Div_{\max}|}{Div_{\max}} \times 100 \tag{14}$$

In (14), Div is the diversity of swarm in an iteration and $Div_{\max}$ is the maximum diversity in all iterations. Xpl% and Xpt% are exploration and exploitation percentages for an iteration, respectively.

3.2 Numerical optimization

In numerical optimization, a mathematically expressed problem is either minimized or maximized, with the help of solution vector representing problem variables. This study focused on minimization problems using commonly used numerical optimization problems in the form of benchmark test functions. In literature, such test functions with diverse properties are vastly used in order to test and validate metaheuristic performances [29]. This study used a set of five test functions including unimodals (Sphere and Schwefel 2.22) and multimodals (Ackley, Rastrigin, Generalized Penalized 1) in nature, Table 1 lists the details. In this table, first column represents the name of the problem, mathematical expression of the problem is given in second column, range specifies the domain of the search environment, whereas the last column shows the objective function value of the optimum solution; the metaheuristic algorithm that generates solution closer to this value is considered an efficient algorithm.

To better understand how the selected metaheuristic algorithms solve these problem, consider a D-dimensional solution vector that represents number of parameters to be tuned to achieve best solution. Each swarm individual in the selected metaheuristic algorithms represents the solution vector and n swarm individuals represent n number of

Table 1 Numerical optimization problems

Test function	Equation	Range	Optimum
Sphere	$f_1(x)=\sum_{i=1}^{D}x_i^2,$	$[-100,100]^D$	0
Schwefel 2.22	$f_2(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	$[-10,10]^D$	0
Ackley	$f_3(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e$	$[-32.768,32.768]^D$	0
Rastrigin	$f_4(x)=10D+\sum_{i=1}^{D}(x_i^2-10cos(2\pi x_i)),$	$[-5.12,5.12]^D$	0
Generalized Penalized 1	$f_5(x)=\frac{\pi}{D}\times\left\{10\sin^2(\pi y_1)+\sum_{i=1}^{D-1}(y_i-1)^2\times[1+\sin^2(\pi y_{i+1})]+(y_D-1)^2\right\}+\sum_{i=1}^{D}u(x_i,10,100,4)$ where$y_i=1+0.25(x_i+1)$ and $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, & x_i>a\\ 0, & -a\leq x_i\leq a\\ k(-x_i-a)^m, & x_i<-a\end{cases}$	$[-50,50]^D$	0

solutions. These solutions are generated by the metaheuristic algorithm during each iteration and the best solution is presented in the end of search iterations.

3.3 SME classification problem

Apart from test functions, the measurements of efficiency, in terms of exploration and exploitation, of metaheuristic algorithms were also performed while solving a real-world classification problem. For this, we employed metaheuristic algorithms on training the parameters of adaptive neuro-fuzzy inference system (ANFIS) [30] for solving the problem of classifying small medium enterprises (SME) of Malaysia based on strength. A brief introduction of ANFIS network is given later in this section.

Literature shows that ANFIS has produced highly accurate models for highly nonlinear problems in several areas of science, engineering, and economics [31, 32]. However, as the complexity of problem increases, the training of ANFIS parameters becomes a tedious job while using the standard gradient-based methods; hence, swarm-based algorithms have been proposed as efficient training methods [33]. This study also employed the selected swarm-based algorithms on the training of premise and consequent parameters of ANFIS model. Similar to numerical optimization problems, each swarm individual in a swarm-based metaheuristic algorithm represents solution vector in classification problem as well. The solution vector in this problem represents membership function parameters and consequent parameter which are to be tuned to find the best fit ANFIS model. Since, every optimization problem has an objective function, in case of this problem, the objective function is ANFIS model. It takes solution vector which includes membership function parameters and consequent parameters. These parameters are employed on ANFIS network to produce output in terms of root mean squared error (RMSE) which is the minimized by the metaheuristic algorithm.

The classification model, based on ANFIS, consisted of seven inputs and one output representing the class of an SME. The inputs are *Business Performance*, *Financial Capability*, *Technical Capability*, *Production Capability*, *Innovation*, *Quality System*, and *Management Capability*. The single output is star ranking (1–5) of an SME, which is taken as class in this problem. For each input, two membership functions of Gaussian type with input space 0–5 were used.

3.3.1 Adaptive neuro-fuzzy inference system (ANFIS)

ANFIS, introduced by Jang [30] in 1993, is a neural network like architecture with fuzzy logic embedded in the form of membership functions and fuzzy rules. As depicted

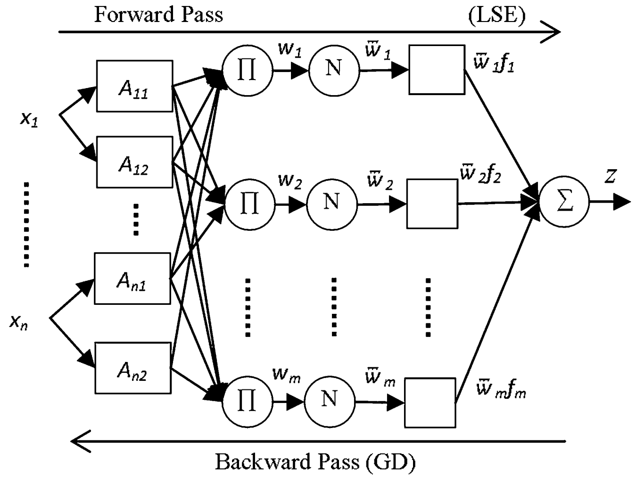


Fig. 3 ANFIS architecture

via Fig. 3, the five-layered architecture of ANFIS contains membership functions, second layer performs product on the membership degrees, third layers normalizes firing strength of a rule, fourth layer performs linear polynomial to calculate rule out, and lastly, the fifth layer simply performs aggregation of rules outputs to generate the model output. Mathematically, each layer of ANFIS architecture can be expressed as following:

Layer 1: Each node A_{ij} in this layer computes membership degree of associated with input variable x_i. The type or shape of membership function can be any—triangular, bell, trapezoidal, or Gaussian which can be defined as (15):

$$A_{ij}(x_i) = e^{-\frac{1}{2}\left(\frac{x_i - c}{\gamma}\right)^2} \tag{15}$$

where c is center and γ is width of the jth Gaussian membership function. These parameters are referred to as premise parameters which are trained by the training algorithm.

Layer 2: Each node $w_k, k = 1, 2, \ldots, m$ in second layer calculates firing strength of kth rule by performing product $\prod$ of membership degrees using (16):

$$w_k = \prod_{i=1}^{m} A_{ij}(x_i) \tag{16}$$

Layer 3: The rule strength computed in previous layer is normalized (17) in this layer to determine the overall strength associated to kth rule with respect to all the fuzzy rules.

$$\bar{w}_k = \frac{w_k}{\sum_{k=1}^{m} w_k} \tag{17}$$

Layer 4: This layer performs linear polynomial f_k on the input variables, which is then multiplied with the normalized firing strength $\bar{w}_k$ using (18):

$$\bar{w}_k f_k, f_k = \left[\left(\sum_{i=1}^{n} x_i p_{k,i}\right) + p_{k,n+1}\right] \tag{18}$$

where $\bar{w}_k$ and f_k represent normalized rule strength and the polynomial function of kth rule, x_i is ith input, $p_{k,i}$ is the real number representing weight associated to ith input in the polynomial function of kth rules, and $p_{k,n+1}$ is also a real number representing the linear coefficient. The parameters $p_{k,i}$ and $p_{k,n+1}$ are the consequent parameters which are trained by the training algorithm.

Layer 5: The single node in this layer represents ANFIS output by aggregating the outputs of m rules using (19):

$$z = \sum_{i=1}^{m} \bar{w}_k f_k \tag{19}$$

ANFIS learns by two-pass learning algorithm which uses least square estimation (LSE) to update consequent parameters in forward pass and in backward pass it uses gradient descend (GD) method to tune premise parameters. In this study, metaheuristic algorithm is employed to update both the membership function and consequent parameters, instead of the standard gradient-based two-pass learning algorithm. The accuracy of ANFIS model is measured through root mean squared error (RMSE) using (20) where $Target_i$ and $Output_i$ are the target output and ANFIS generated output for the ith tuple in a dataset with N instances.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Target_i - Output_i)^2} \tag{20}$$

4 Experiments

4.1 Experimental settings

To analyze the two highly influential factors (exploration and exploitation) of the metaheuristic algorithms under consideration, five commonly used numerical optimization problems with different modality were employed with 30 dimensions, Table 1 lists the test functions. The swarm size was 50 for each algorithm and maximum iterations were 1500 and 200 for numerical problems and for ANFIS training in classification problem, respectively. As mentioned earlier, the purpose of this study was to analyze the said two factors; therefore, the focus was mainly on calculating diversity in swarm during iterations instead of running the algorithm over certain number of independent runs and averaging the results. Accordingly, we executed algorithms once, as our preliminary experiments also

Table 2 Algorithm-specific parameter settings

Algorithm	Parameter settings
PSO	$\omega = [0.9 - 0.4], C_1 = C_2 = 2$ [34]
ABC	$Limit = SwarmSize \times D$ [35]
$ACO_{\mathbb{R}}$	$\tau_0 = 1, \rho = 0.5, \omega = 0.5(WeightFactor), z = 1(Deviation - DistanceRatio)$ [27]
CS	$\rho = 0.25$ [36]
FA	$\beta_0 = 1, \gamma = 1, \alpha = 0.2$ [37]

evidenced insignificant difference in results over multiple runs.

Besides the common settings explained above, the algorithm-specific parameter settings are presented in Table 2. For these settings, we performed a careful survey and took parameter values from literature solving the test functions adopted in this study.

4.2 Results

Here, the results are reported for the experiments performed to obtain exploration and exploitation measures of top five swarm-based metaheuristics on numerical optimization and classification problems. The statistics related to numerical problems are given in Table 3, whereas Table 4 presents results on classification problem. Along with statistical information, this section also illustrates the algorithm performances more comprehensively via figures and charts.

For numerical optimization problems, Table 3 presents the best objective function values found by different algorithms, percentage of exploration and exploitation showing the two abilities, diversity measurement indicating variety in solutions found during iterations, and number of function evaluations (NFEs) as each algorithm performs different number of evaluations during an iteration. Figures 4, 5, 6, 7, 8, 9 and 10 provide visual evidence of exploration, exploitation, and population diversity in

Table 3 Results of numerical optimization problems

Functions	Measurements	PSO	ABC	$ACO_{\mathbb{R}}$	CS	FA
Sphere	Best solution	1.65E−09	1.44E−15	4.50E−37	8.13E−08	2.73E−27
	Average Xpl%:Xpt%	39:61	56:44	70:30	65:35	91:09
	Diversity	117.05	120.53	155.28	149.74	163.60
Schwefel 2.22	Best solution	2.80E+01	1.16E−08	4.01E−20	3.98E−03	1.59E−13
	Average Xpl%:Xpt%	35:65	58:42	90:10	51:49	83:17
	Diversity	106.32	101.32	144.40	116.92	144.56
Ackley	Best solution	1.56E−05	3.95E−08	6.22E−15	0.0088	3.82E−14
	Average Xpl%:Xpt%	36:67	61:39	71:29	55:45	92:08
	Diversity	111.84	133.28	155.58	105.58	163.21
Rastrigin	Best solution	92.6531	182.7922	177.0499	85.4237	29.8487
	Average Xpl%:Xpt%	42	72	66:34	80:20	82:18
	Diversity	120.78	148.31	145.78	186.73	122.35
Generalized Penalized 1	Best solution	1.28E−01	9.99E−03	2.24E−27	5.34E−01	2.68E−27
	Average Xpl%:Xpt%	40:60	58:43	67:33	71:29	86:14
	Diversity	124.95	132.33	156.10	163.47	162.34
NFEs		75,000	112,600	120,050	150,050	1,837,550

Table 4 Results of SME classification problem

	PSO	ABC	$ACO_{\mathbb{R}}$	CS	FA
Training RMSE	0.0560	0.0871	0.1336	0.0564	0.0560
Testing RMSE	0.0574	0.1072	0.1508	0.0583	0.0583
Average Xpl%:Xpt%	19:81	63:37	96:04	97:03	38:62
Diversity	253.11	511.04	672.10	1118.35	371.78
NFEs	10,000	15,050	16,050	20,050	245,050

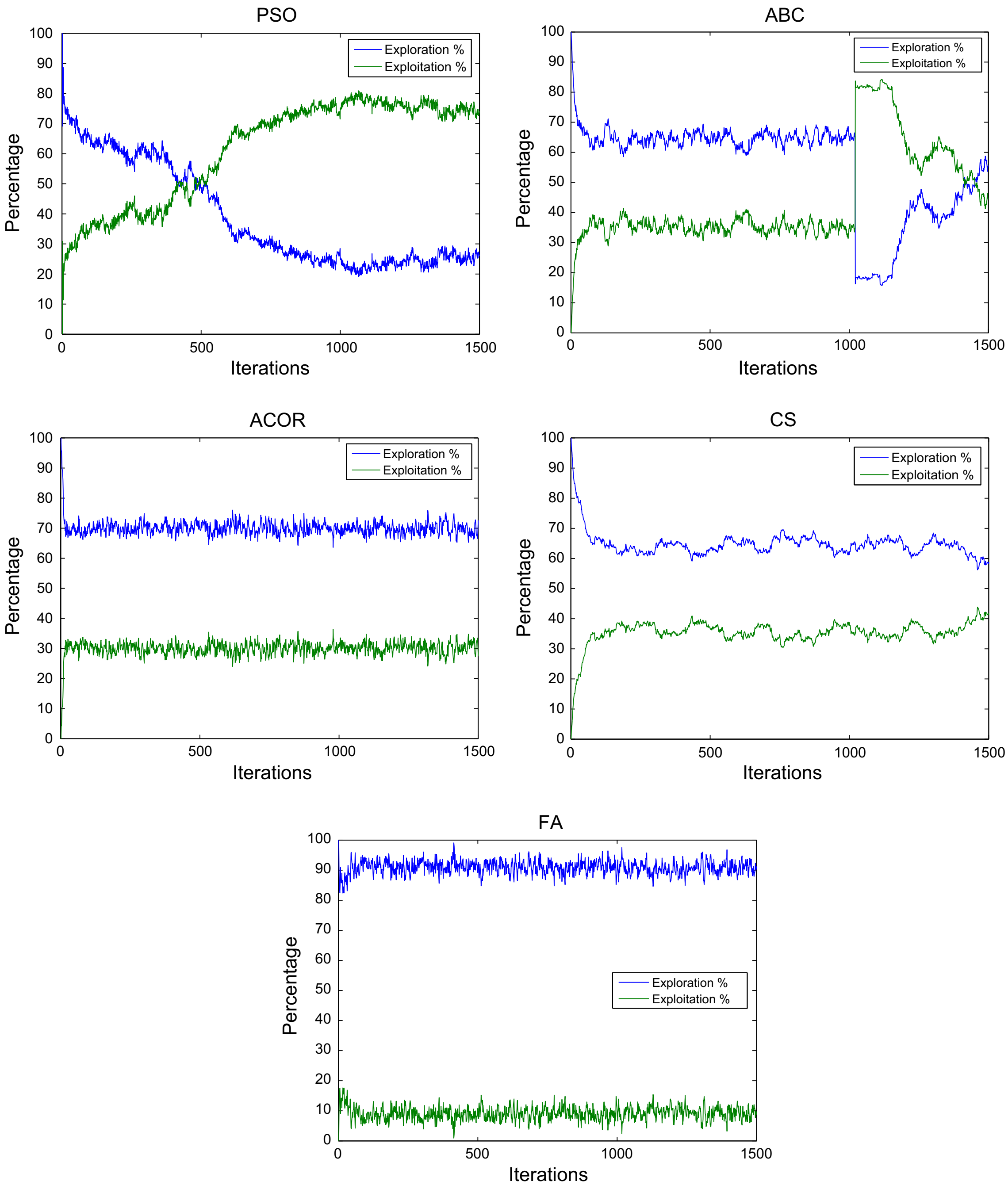


Fig. 4 Exploration and exploitation of metaheuristics on Sphere function

swarms of the selected swarm-based metaheuristic algorithms.

According to Table 3, ACO$_{\mathbb{R}}$ was the best performer overall on unimodal and multimodal problems, followed by FA which stood second best in all numerical problems. However, Rastrigin was the problem where ACO$_{\mathbb{R}}$ performed the second worst and FA obtained best result, as the said function is a highly multimodal problem. Overall, the least performer was CS but it managed to achieve second best objective function value in Rastrigin problem. ABC performed third best in all functions except for Rastrigin where it was least performer.

As per function-wise performances, ACO$_{\mathbb{R}}$ and FA achieved best results with around 70%:30% and around 90%:10% average exploration-exploitation ratios, respectively. In this case, the least and second least performers were CS and PSO with around 65%:35% and 40%:60% average exploration-exploitation ratios, respectively. On

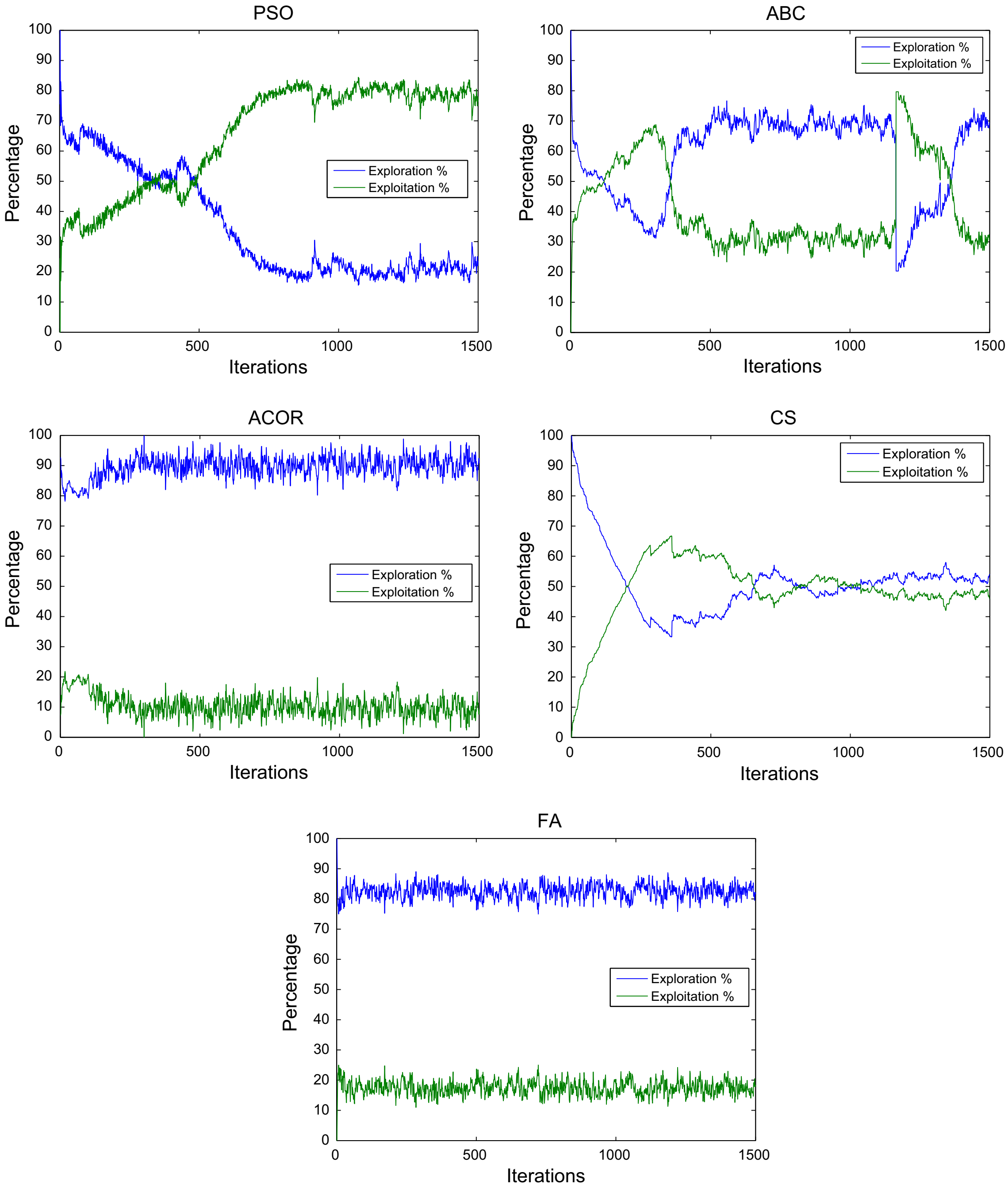


Fig. 5 Exploration and exploitation of metaheuristics on Schwefel 2.22 function

Schwefel 2.22, the first and second best values were obtained by $\text{ACO}_{\mathbb{R}}$ and FA with exploration above 80% and exploitation below 20%. Likewise, the third and fourth best performer ABC and CS maintained exploration greater than 50% exploitation below the said percentage. The best and second best performers on Ackley were $\text{ACO}_{\mathbb{R}}$ and FA with average exploration-exploitation 70%:30% and 90%:10%, respectively. Both the algorithms retained diversity above 150, whereas the least performer CS kept swarm diversity around 100, and maintained average exploration-exploitation ratio around 50%:50%. On Rastrigin function which proved to be difficult optimization problem for all the algorithms, the first and second best performances were reported with FA and CS which maintained the ratio of average exploration-exploitation around 80%:20%. ABC was the least performer in this case. ABC kept average exploration-exploitation ratio around 70%:30%. $\text{ACO}_{\mathbb{R}}$ and FA were also top performers

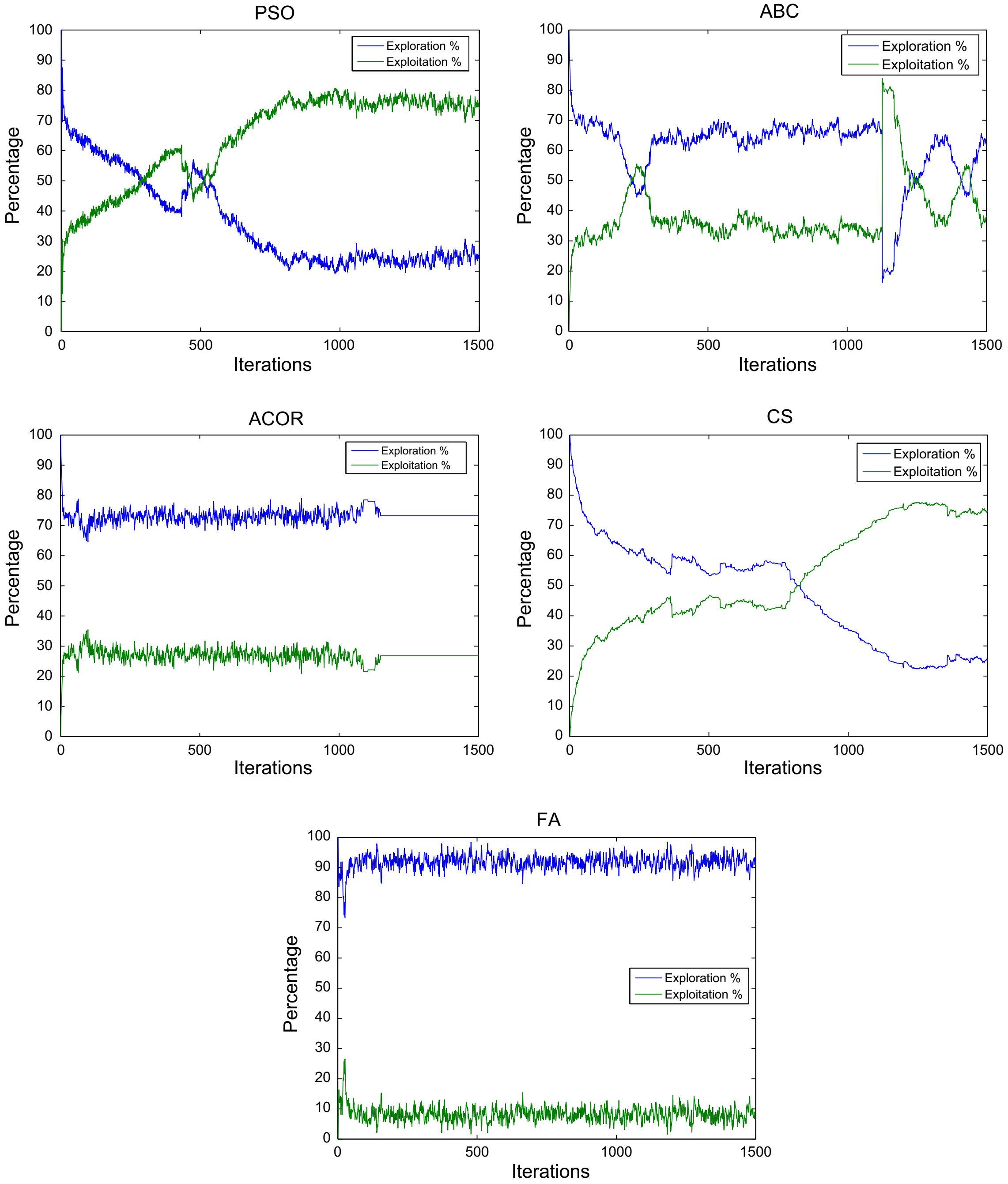


Fig. 6 Exploration and exploitation of metaheuristics on Ackley function

in Generalized Penalized 1 function in which the algorithms performed search with 66%:34% and 86%:14% ratios of exploration and exploitation, respectively. Generally, other than in Rastrigin problem, the top performers $ACO_{\mathbb{R}}$ and FA were more explorative than exploitative with on average ratios of 70%:30% and 85%:15%, respectively. On the other hand, the case was reverse in PSO which remained more exploitative as compared to explorative with ratio around 40%:60%. ABC and CS were opposite in ratios to PSO.

In terms of number of function evaluations (NFEs), Table 3 shows that the most expensive algorithm was FA and the least was PSO, whereas $ACO_{\mathbb{R}}$ was mediocre on numerical problems.

The statistical facts reported above can be graphically evidenced in Figs. 4, 5, 6, 7 and 8 which show exploration

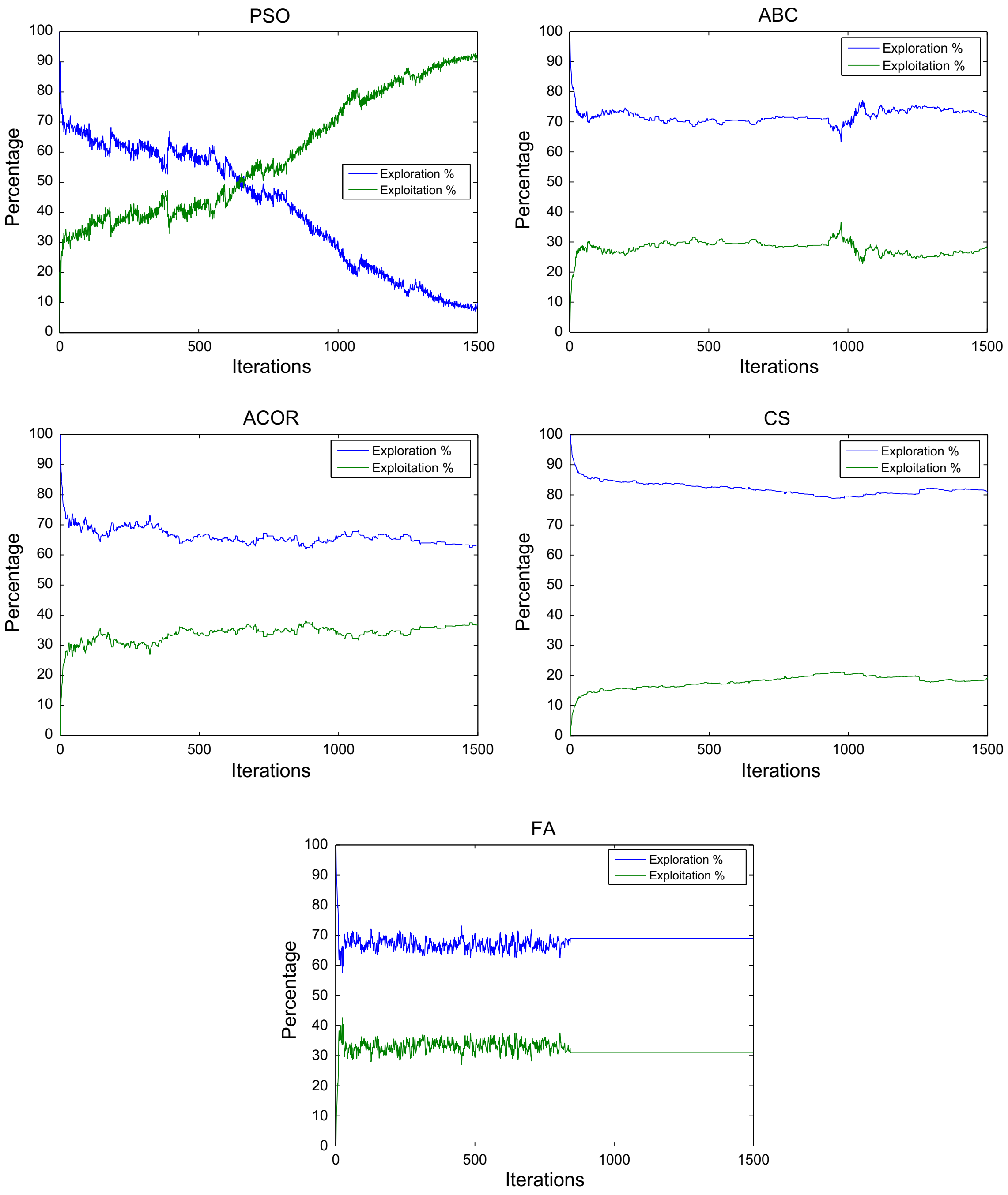


Fig. 7 Exploration and exploitation of metaheuristics on Rastrigin function

and exploitation ratios during search process maintained by different algorithms while solving numerical problems. In these figures, it can be observed that ACO$_{\mathbb{R}}$, FA, ABC, and CS retained exploration higher than exploitation either throughout iterations or most part of search process. Whereas, PSO started as explorative and later on, soon after few iterations, converted to exploitative algorithm in nature. This can be further observed in stacked bar charts (Fig. 9) in terms of exploration and exploitation percentages of the algorithms. Figure 10 illustrates the behavior of swarm in terms of diversity measurement during iterations.

Figure 10 shows diversity measurement during iterations in an algorithm. From these figures, it can be observed that the diversity in PSO was high initially, which dropped gradually soon after initial part of search process. This is consistent with Figs. 4, 5, 6, 7 and 8 where PSO was explorative in the beginning and later terned into

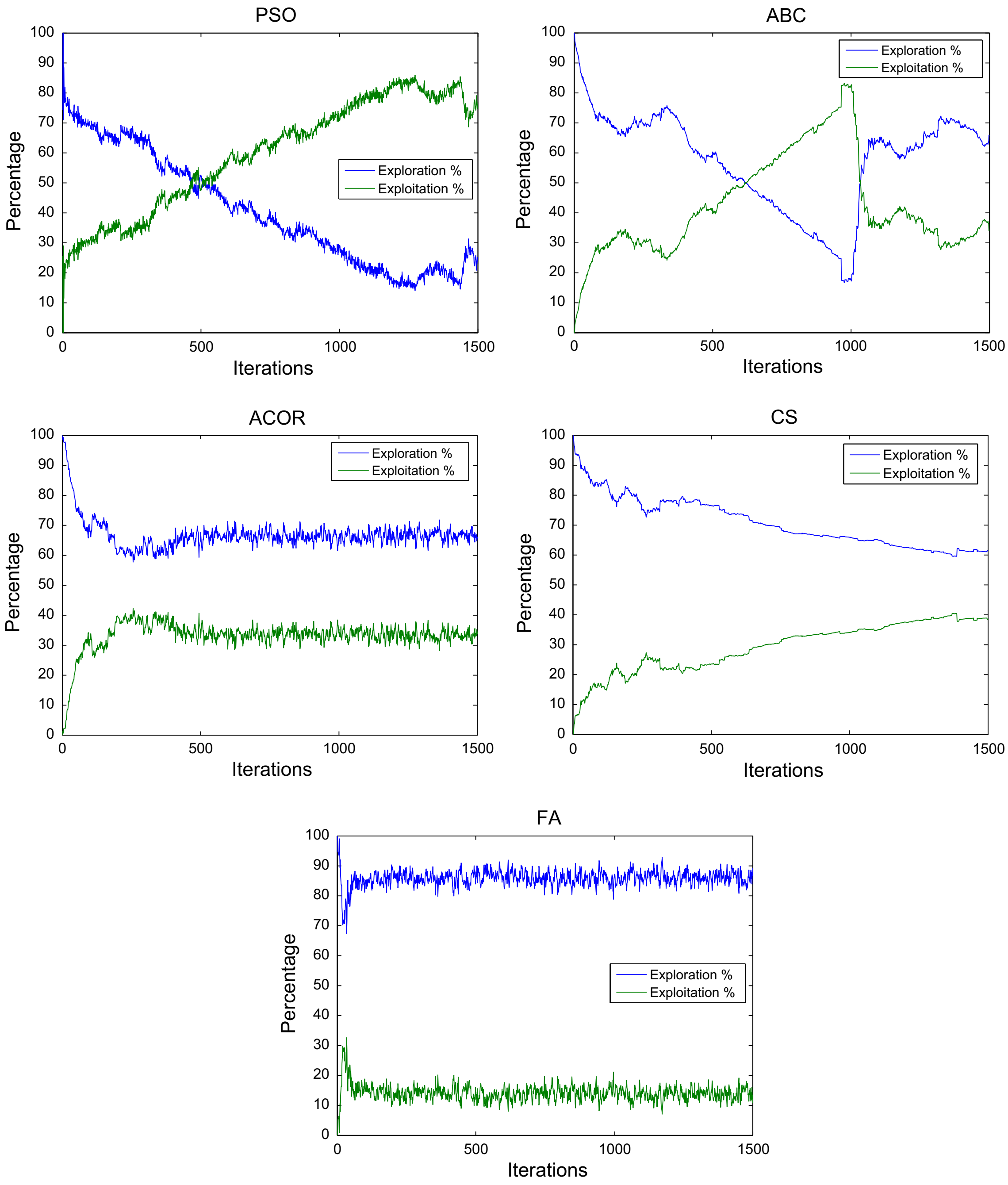


Fig. 8 Exploration and exploitation of metaheuristics on Generalized Penalized 1 function

exploitative. $ACO_{\mathbb{R}}$ and FA remained consistent with diversity in all the functions hence maintained regular exploration and exploitation throughout experiments (Fig. 9). ABC, on Sphere and Ackley, was consistent with diversity measurement until 1000 iterations; afterward, the introduction of scout bees disrupted the momentum. The jerk after 1000 iterations on Schwefel 2.22 and Generalized Penalized 1 also shows the appearance of scout bees in ABC, which produced random solutions in place of abandoned food sources. Rastrigin was on exception for ABC as compared to other test functions. CS was also consistent in diversity measurement except for Ackley where it became exploitative in later part of iterations.

Other than simulations on numerical problems, the algorithms were further tested on a real-world application of classification problem, Table 4 presents the results. From the statistics, the worst performer on numerical problems, PSO achieved best error on both training and testing datasets with around 19%:81% average exploration-exploitation ratio. PSO was followed by FA which

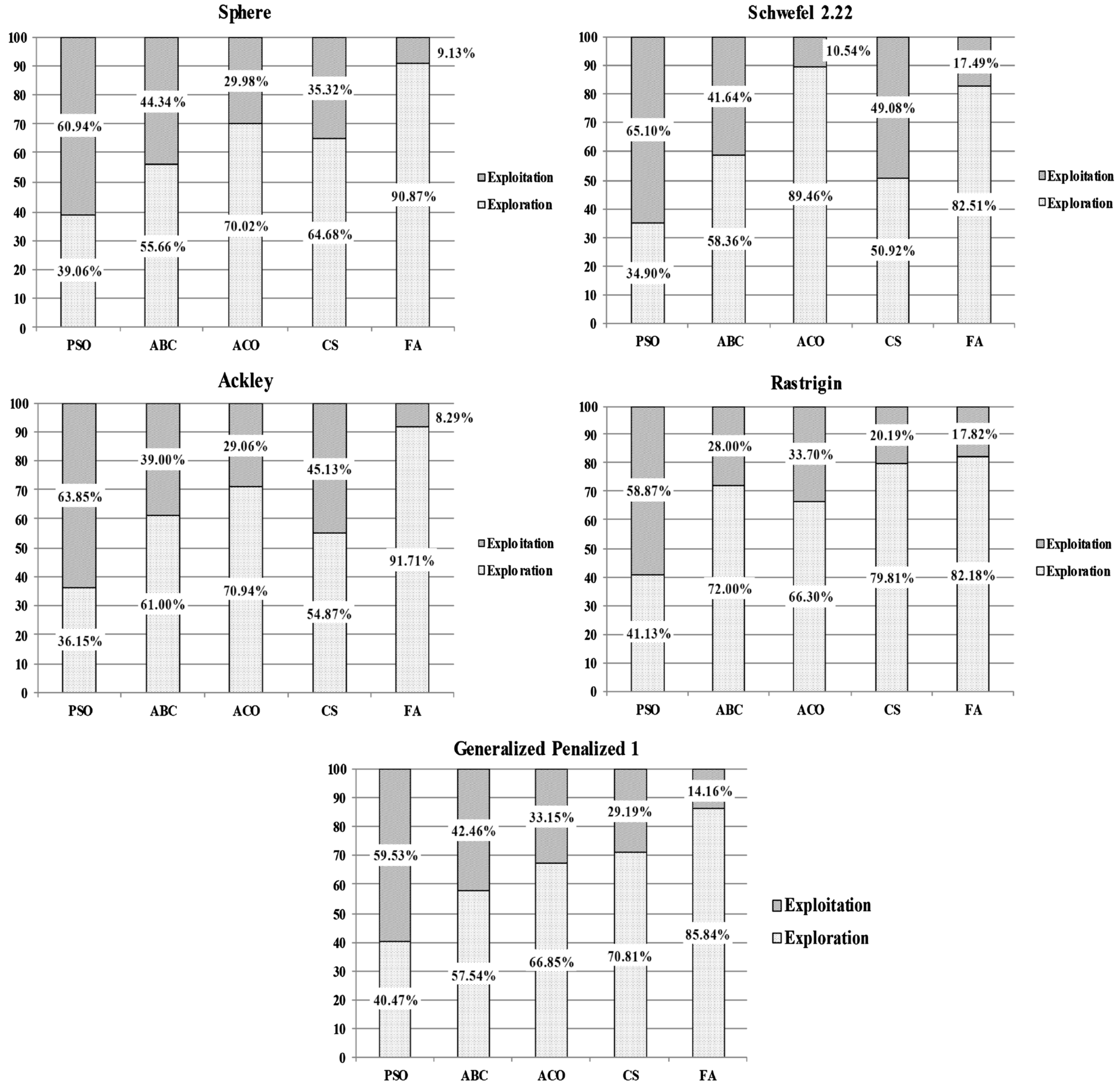


Fig. 9 Average exploration and exploitation of metaheuristics on numerical problems

produced second best errors with average exploration-exploitation ratio of around 39%:61%. Whereas, the best performer on numerical problems, $ACO_{\mathbb{R}}$ obtained worst error rates due to high exploration and low exploitation ratio of 96%:4%. Just like in most of the numerical problems, also in this problem, ABC maintained around 60%:40% ratio of exploration and exploitation, produced adequately better results (see Figs. 9, 11). The proportion of exploration and exploitation maintained by the algorithm throughout iterations is depicted via Fig. 12. According to the line graphs, it is clear the PSO and FA were exploitative in most of the iterations, whereas $ACO_{\mathbb{R}}$ and CS remained highly explorative during iterations. This is further evidenced from the diversity measurement presented via line graph in Fig. 13 which shows that PSO, ABC and FA maintained lower diversity as compared to $ACO_{\mathbb{R}}$ and CS.

5 Analysis and discussion

The performance of any swam-based metaheuristic algorithm strongly depends on the way swarm individuals are manipulated, meaning that search strategy adopted by

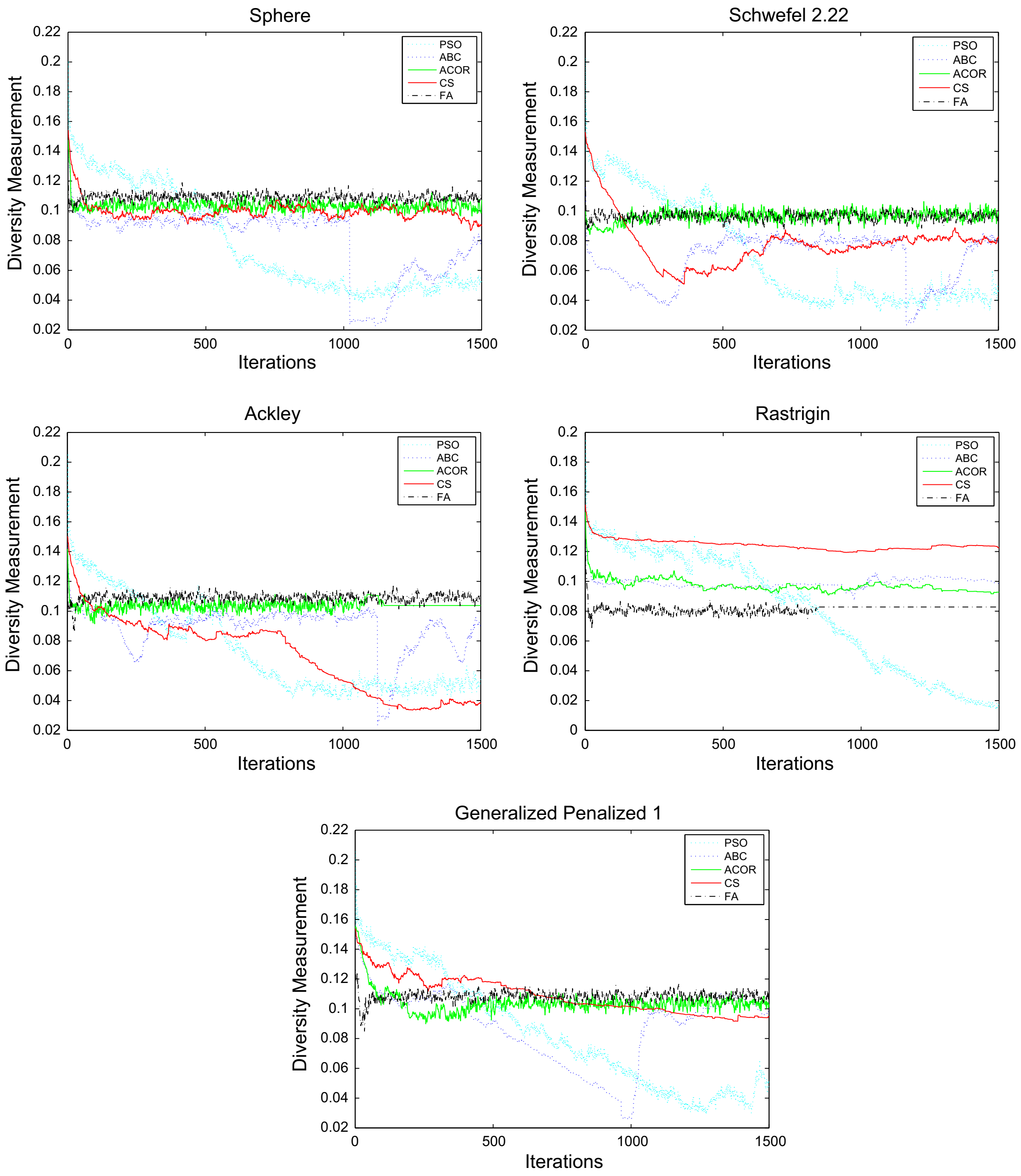


Fig. 10 Diversity in swarm individuals on numerical problems

swarm individuals reflects how the individuals coordinate search information during the course of iterations. More importantly, the major performing factor is balancing exploration and exploitation by maintaining adequate diversity in swarm individuals so that trapping in local optimal locations as well as ignorance of potential neighborhoods, because of unnecessary diversification, may be avoided.

According to the results of numerical problems mentioned in previous section, ACO$_\mathbb{R}$ outperformed the other well-known counterpart algorithms due to consistency and balance in explorative and exploitative capabilities. There are reasons: (a) ants endorse positive feedback from ants already found improved results, and (b) well distributed swarm individuals, resulted in consistent diversity in whole swarm. In terms of cost, ACO$_\mathbb{R}$ consumed mediocre NFEs—neither too low as PSO nor too high as FA. The performance of ACO$_\mathbb{R}$ is relevant to the evaporation parameter, with high parameter value the ACO$_\mathbb{R}$ becomes more explorative; otherwise exploitative.

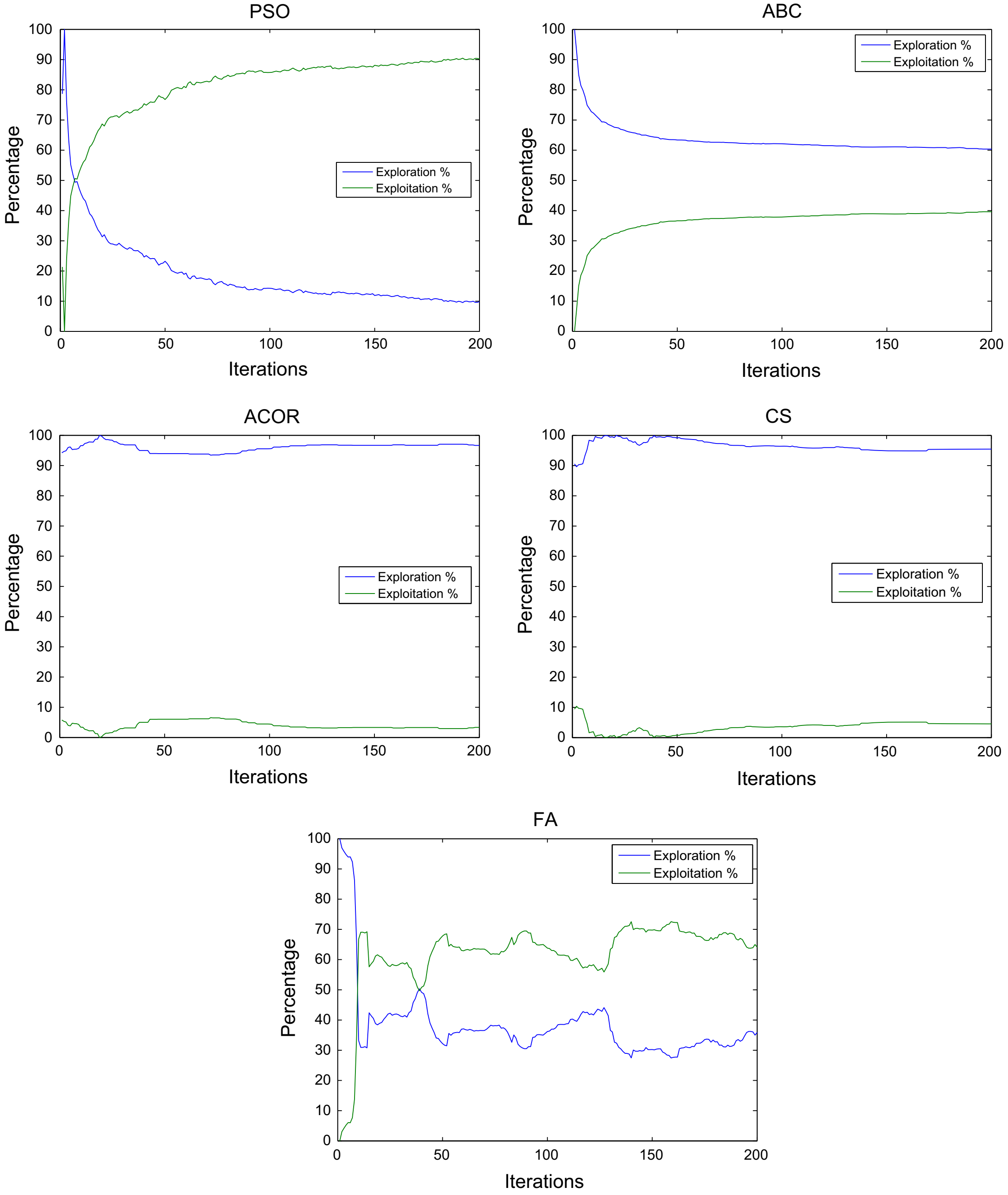


Fig. 11 Average exploration and exploitation of metaheuristics on SME classification problem

FA was inversely extreme in both the conflicting capabilities, significantly high on exploration and equally low on exploitation. This is because of the use of Lévy Flight which helps avoid local optima through long distance agility of the swarm individuals. Nevertheless, the algorithm still managed to obtain second best results on numerical problems, thanks to consistency in swarm individuals and swarm as a whole. Hybriding FA with any local search method will improve performance due to balance between exploration and exploitation gained.

ABC proved to be potential search algorithm even though the performance was third best in the experiments of this study. According to exploration and exploitation measurements, ABC maintained adequate balance between the two factors until the introduction of scout bees in the later part of iterations—affected the consistency in swarm

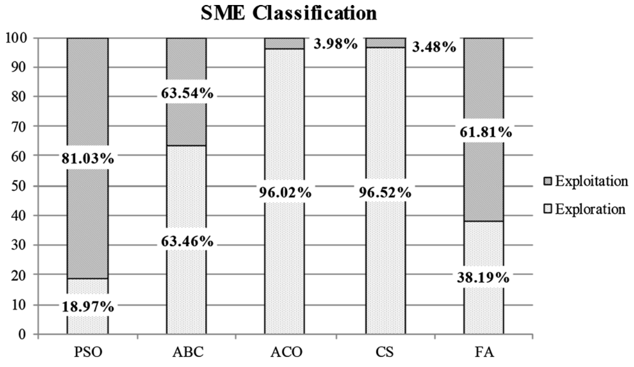


Fig. 12 Exploration and exploitation of metaheuristics on SME classification problem

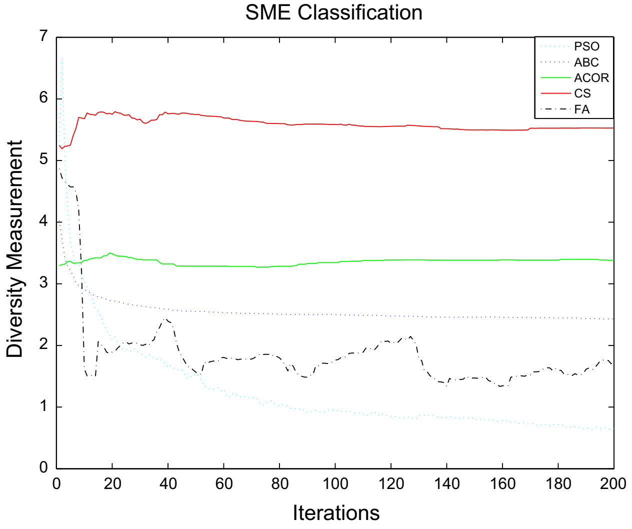


Fig. 13 Diversity in swarm individuals on SME classification problem

diversity. This disturbed the rhythm of the swarm individuals which later struggled to regain the coherence. It infers that the best solution found after the first three quarters of iterations was replaced with a randomly produced solution by a scout bee. The better handling of scout bees may help improve ABC performance since employed and onlooker bees already preserve consistency of diversity in the swarm.

PSO was opposite to $\text{ACO}_{\mathbb{R}}$ in maintaining diversity in the swarm individuals. It was low at exploration and high at exploitation. The premature convergence proved that the algorithm spent most of the time in locally optimal solutions. The ability of global search in PSO is considerably weak. After initial iterations, the explorative capability of PSO dropped dramatically since the social component of the update equation did not work as expected. Although, inertia weight is supposed to balance exploration and exploitation, but this approach also failed in this regard. Hence, better explorative approach embedded into PSO equation may help improve the results.

On numerical problems, CS proved that balancing exploration and exploitation does not mean 50%:50%. This algorithm performed worst because of lack of coherence in the swarm individuals. Moreover, opposite to PSO, CS converged to relatively potential (not globally) optimum solutions early in the iterations, but did not manage to find better solutions in the later part of search. Hence, both the search strategies (local and global) need to be revised and improved by any approach that maintain diversity in the swarm individuals.

Apart from numerical problems, the results on real-world application of training highly nonlinear fuzzy neural network on classification problem suggested the nature of optimization problem highly matters for metaheuristic algorithms. The difficulty in real applications, instead of simulation problems, poses variety of challenges for these algorithms. It is therefore, the poor performer PSO outperformed the rest of the algorithms by producing better training and testing errors, probably by better exploitation. The same was with FA which produced second best errors due to low diversity. Hence, it shows that the desired exploration and exploitation capability is problem specific.

6 Conclusion

The purpose of this study was to evaluate explorative and exploitative capabilities of the top five commonly used swarm-based metaheuristic algorithms, using diversity measurement. Unlike existing literature often merely observing convergence graph and end results for the performance analysis, this study proposed an effective approach to insightful analyses that revealed the answer to question ‘why and how it happened’ related to metaheuristic performance. The measurement of exploration and exploitation helped draw comprehensive inference on the reasons behind poor or better results.

From the experimental results, it was obvious that coherence among swarm individuals is the key to success for any swarm-based algorithm. The consistency and adequate diversity in the swarm are the core ingredients to search strategy adopted. The trade-off balance between exploration and exploitation does not mean 50:50, the search mechanism that avoids too much of exploration and considerably scarce exploitation may achieve efficient results. Among the other algorithms, $\text{ACO}_{\mathbb{R}}$ appropriately maintained the trade-off balance in exploration and exploitation for unimodal and multimodal problems. FA and ABC also proved to be potential choices in the list of swarm-based metaheuristics, however modification in local search ability may improve consistency in swarm individuals—resulting in convergence to globally potential neighborhoods.

Other than numerical optimization problems, the application on real-world classification problem suggested that merely high-level analyses of experiments on test functions may not help conclude on algorithm robustness. Real applications pose variety of difficulties inherent in problem landscapes hence metaheuristic performances should also be analyzed on actual problems in the domains of engineering, business, science, etc. It was observed that, as opposite to numerical problems, PSO performed best in classification problem and $\mathrm{ACO}_{\mathbb{R}}$ results were the worst. The analyses suggested that the algorithms that maintained better exploitation ability produced better results, as compared to algorithms with high exploration. The measurement of exploration and exploitation not only helps understand swarm-behavior on real-life problems but also it reveals the level of difficulty in the problem.

In future, this study may be extended to analyze exploration and exploitation in variety of other metaheuristic algorithms on wide range numerical optimization problems, as well as, on real-life problems with varying difficulty.

Acknowledgements The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) for supporting this research under Postgraduate Incentive Research Grant, Vote No. U560.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Cheng S, Zhang Q, Qin Q (2016) Big data analytics with swarm intelligence. Ind Manag Data Syst 116(4):646–666
2. Adekanmbi O, Green P (2015) Conceptual comparison of population based metaheuristics for engineering problems. Sci World J. https://doi.org/10.1155/2015/936106
3. Castro M, Sörensen K, Vansteenwegen P, Goos P (2015) A fast metaheuristic for the travelling salesperson problem with hotel selection. 4OR 13(1):15–34
4. Maya PA, Sörensen K, Goos P (2010) An efficient metaheuristic to improve accessibility by rural road network planning. Electron Notes Discrete Math 36:631–638
5. Hidalgo IG, de Barros RS, Fernandes J, Estrócio JPF, Correia PB (2015) Metaheuristic approaches for hydropower system scheduling. J Appl Math. https://doi.org/10.1155/2015/701851
6. Yang X-S (2012) Efficiency analysis of swarm intelligence and randomization techniques. J Comput Theor Nanosci 9(2):189–198
7. Kennedy J, Eberhart R (1995) Particle swarm optimization (pso). In: Proceedings of the IEEE international conference on neural networks, Perth, Australia, pp 1942–1948
8. Dorigo M, Di Caro G (1999) Ant colony optimization: a new meta-heuristic. In: Proceedings of the 1999 congress on evolutionary computation, 1999 (CEC99), vol 2. IEEE, pp 1470–1477
9. Tereshko V, Loengarov A (2005) Collective decision making in honey-bee foraging dynamics. Comput Inf Syst 9(3):1
10. Yang X-S, Deb S (2009) Cuckoo search via lévy flights. In: World congress on nature & biologically inspired computing, 2009 (NaBIC 2009). IEEE, pp 210–214
11. Yang X-S (2010) Firefly algorithm. In: Engineering Optimization. Wiley, Hoboken, NJ, USA, pp 221–230
12. Tan Y, Zhu Y (2010) Fireworks algorithm for optimization. Adv Swarm Intell 6145:355–364
13. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010), pp 65–74
14. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
15. Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
16. Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst 22(3):52–67
17. Zhang Y, Wang S, Ji G (2015) A comprehensive survey on particle swarm optimization algorithm and its applications. Math Probl Eng. https://doi.org/10.1155/2015/931256
18. Karaboga D, Gorkemli B, Ozturk C, Karaboga N (2014) A comprehensive survey: artificial bee colony (abc) algorithm and applications. Artif Intell Rev 42(1):21–57
19. Mavrovouniotis M, Li C, Yang S (2017) A survey of swarm intelligence for dynamic optimization: algorithms and applications. Swarm Evol Comput 33:1–17
20. Shehab M, Khader AT, Al-Betar MA (2017) A survey on applications and variants of the cuckoo search algorithm. Appl Soft Comput 61:1041–1059
21. Tilahun SL, Ngnotchouye JMT, Hamadneh NN (2017) Continuous versions of firefly algorithm: a review. Artif Intell Rev. https://doi.org/10.1007/s10462-017-9568-0
22. Gao S, Wang Y, Cheng J, Inazumi Y, Tang Z (2016) Ant colony optimization with clustering for solving the dynamic location routing problem. Appl Math Comput 285:149–173
23. Yang X-S (2011) Metaheuristic optimization: algorithm analysis and open problems. In: International symposium on experimental algorithms. Springer, Berlin, pp 21–32
24. Glover F, Sörensen K (2015) Metaheuristics. Scholarpedia 10(4):6532
25. Sorensen K, Sevaux M, Glover F (2017) A history of metaheuristics. arXiv preprint arXiv:1704.00853
26. Yang X-S (2012) Swarm-based metaheuristic algorithms and no-free-lunch theorems. Theory and New Applications of Swarm Intelligence Rafael Parpinelli. IntechOpen. https://doi.org/10.5772/30852
27. Leguizamón G, Coello CAC (2010) An alternative $\mathrm{ACO}_{\mathbb{R}}$ algorithm for continuous optimization problems. In: International conference on swarm intelligence. Springer, Berlin, pp 48–59
28. Cheng S, Shi Y, Qin Q, Zhang Q, Bai R (2014) Population diversity maintenance in brain storm optimization algorithm. J Artif Intell Soft Comput Res 4(2):83–97
29. Jamil M, Yang X-S (2013) A literature survey of benchmark functions for global optimisation problems. Int J Math Model Numer Optim 4(2):150–194
30. Jang J-SR (1993) Anfis: adaptive-network-based fuzzy inference system. IEEE Trans Syst, Man Cybern 23(3):665–685
31. Kar S, Das S, Ghosh PK (2014) Applications of neuro fuzzy systems: a brief review and future outline. Appl Soft Comput 15:243–259

32. Najafzadeh M, Etemad-Shahidi A, Lim SY (2016) Scour prediction in long contractions using anfis and SVM. Ocean Eng 111:128–135
33. Karaboga D, Kaya E (2013) Training anfis using artificial bee colony algorithm. In: 2013 IEEE international symposium on innovations in intelligent systems and applications (INISTA). IEEE, pp 1–5
34. Zhan Z, Zhang J, Shi Y, Liu H (2012) A modified brain storm optimization. In: 2012 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
35. Karaboga D, Akay B (2009) A comparative study of artificial bee colony algorithm. Appl Math Comput 214(1):108–132
36. Nawi NM, Rehman MZ, Khan A, Chiroma H, Herawan T (2016) A modified bat algorithm based on gaussian distribution for solving optimization problem. J Comput Theor Nanosci 13(1):706–714
37. Zhang L, Liu L, Yang X-S, Dai Y (2016) A novel hybrid firefly algorithm for global optimization. PloS ONE 11(9):e0163230