International Journal on Advanced Science Engineering Information Technology

Vol.8 (2018) No. 4
ISSN: 2088-5334

# Personal Best Cuckoo Search Algorithm for Global Optimization

Kashif Hussain[#], Mohd Najib Mohd Salleh[#], Yuli Adam Prasetyo[*], Shi Cheng[$]

[#]*Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia, Batu Pahat, 86400, Johor, Malaysia*
*E-mail: najib@uthm.edu.my*

[*]*School of Industrial Engineering, Telkom University, 40257 Bandung, West Java, Indonesia*

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

***Abstract*— Real-life optimization problems demand robust algorithms that perform efficient search in the environment without trapping in local optimal locations. Such algorithms are equipped with balanced exploration and exploitation capabilities. Cuckoo search (CS) algorithm is also one of these optimization algorithms, which is inspired by nature. Despite effective search strategies such as Lévy flights and solution switching approach, CS suffers from a lack of population diversity when implemented in hard optimization problems. In this paper, enhanced local and global search strategies have been proposed in the CS algorithm. The proposed CS variant uses personal best information in solution generation process, hence called Personal Best Cuckoo Search (pBestCS). Moreover, instead of constant value for switching parameter, pBestCS dynamically updates switching parameter. The prior approach enhances local search ability, whereas the later modification enforces effective global search in the algorithm. The experimental results on test suite with different dimensions validated the efficiency of the proposed modification on optimization problems. Based on statistical and convergence analysis, pBestCS outperformed standard CS algorithm, as well as, particle swarm optimization (PSO) and artificial bee colony (ABC).**

***Keywords*— metaheuristic; global optimization; personal best; cuckoo search; particle swarm optimization; artificial bee colony**

## I. INTRODUCTION

Optimization is needed in business, engineering, and scientific fields to achieve the best results with minimum budget and resources. Finding optimum solutions is a difficult task; it requires plenty of resources and computation. Thanks to optimization algorithms including genetic algorithms (GA) [1], differential evolution (DE) [2], particle swarm optimization (PSO) [3], ant colony optimization (ACO) [4], artificial bee colony (ABC) [5], and cuckoo search (CS) [6] which have reduced computational cost and efficiently solved optimization problems. These powerful metaheuristic algorithms have been designed with the inspiration from intelligence found in nature.

Cuckoo search (CS) is also one of the recently introduced popular metaheuristic algorithms, which are categorized as population-based algorithms, where a set of solutions are evaluated to find the best one. CS has been implemented on a wide variety of problems in the domains of medical, image processing, data mining, engineering, energy and economics, etc. [7]. According to Mareli and Twala [8], CS is an efficient algorithm for global optimization problems than other population-based metaheuristics. This has been evidenced in multiple studies where CS has outperformed the counterparts. For example, [9] solved a non-linear optimization problem of optimal power flow at the distribution system, using CS and PSO. The research concluded that CS produced efficient results compared to PSO due to strong search ability and fast convergence speed. In another experimental study, [10] efficiently achieved an optimal design of truss structures using CS algorithms. The results of experiments on design problems with a different number of decision variables and design complexities validated that CS is a better algorithm than other state-of-the-art metaheuristics like PSO and GA. [11] employed CS on load frequency load problem for optimum tuning of PI controllers. In this study, a three-area power system was considered to evaluate the proposed model. In comparison with other famous counterparts GA and PSO, CS achieved optimum results. In a recent comparative experimental study, [12] found CS better than PSO solving the highly nonlinear problem of optimal power flow at distribution network.

Apart from the applications of CS mentioned above, various modifications and hybrids of the algorithm are proposed in the literature. [13] proposed a PSO-inspired modification in CS to enhance its convergence rate. The modification is made in two components of CS: firstly, to enhance diversity in population, a new population was injected with neighborhood information; secondly, two new

search strategies were introduced in CS to balance exploration and exploitation. The CS variant outperformed classic and latest metaheuristic algorithms, including PSO, ABC, Bat Algorithm (BA), etc., on 30 benchmark test functions. In another work, [8] introduced the strategy of dynamically adjusting the switching parameter which is fixed to 25% in the standard CS algorithm. The research proposed three different variants based on this strategy: linearly increasing switching parameter value with some iterations, exponential increase in a parameter with the increase in iterations, and lastly, increasing parameter value to the power three as the iterations increase. The variants were tested on ten benchmark mathematical functions and compared with the standards CS and other variants. The researchers found that the strategy of increasing switching parameter value exponentially was more effective than others. Another PSO-inspired modification in CS was proposed in [14]. The modified variant, the so-called Adaptive CS algorithm (ACSA), adopted search acceleration strategy in PSO which controls inertia weight. Same as PSO, ACSA also controlled step size parameter used in Lévy flight random walk. The policy of adapting step size was based on the higher survival rate of cuckoo eggs. Hence, the proposed CS variant was able to explore search space more rigorously for searching the suitable breeding place for cuckoos. The results of experiments on five benchmark test functions with different modalities and dimensions validated the effectiveness of the proposed ACSA in comparison with PSO, GA, DE, and other CS variants from literature.

CS is also hybridized with other heuristic and metaheuristic algorithms to benefit from one or the other technique. [15] hybridized CS with hill climbing algorithm in a way that CS started the global search. After finding potential neighborhoods, the search was handed over to hill climbing algorithm for accelerated convergence to optimum solutions. The proposed variant overcame a slow convergence issue in standard CS. The search evaluated the so-called CS Algorithm with Hill Climbing (CSAHC) on global optimization tasks and found CSAHC effective modification. Another hybrid variant of CS was proposed in [16] where CS was hybridized with PSO for solving continuous optimization and engineering design problems. The research proposed a modification in population initialization, adaptively adjusting control parameters and the incorporation of PSO. The primary purpose of hybridizing PSO with CS was to increase population diversity and increase convergence speed. In this research, the PSO update equation was embedded into CS following the cuckoo position update equation. The proposed CSPSO outperformed PSO, CS, BA, and different other metaheuristic algorithms from literature while solving benchmark numerical optimization problems. In [17], CS was hybridized with the GA algorithm to propose two hybrid schemes. In this research, CS or GA was first employed on an exploration of the search environment, and the other algorithm to get rid of local minima problem improved later global search. The experimental results on benchmark test functions proved the efficiency of the proposed hybrid strategy.

Despite multiple modifications and hybrids proposed by researchers in literature, the room of improvement in CS algorithm is still significant. Mostly, the hybrids and modifications often make the algorithm more complicated and challenging to implement. To address this, the current paper proposes a simple modification in CS inspired by PSO, using personal best information in position update equation. The proposed CS variant also adaptively adjusts switching parameter as the iterations proceed. For the validation of the effectiveness of the suggested changes in CS, numerical optimization problems have been solved in this experimental study. The results are then compared with PSO, ABC, and standard CS. The paper is organized as follows. The subsequent section presents materials and methods. The fundamental knowledge of CS algorithm is given, followed by the proposed modification. The results are presented and discussed in Section III along with the detail of the experimental environment. The study is duly concluded in Section IV.

## II. Materials and Methods

Deb and Yang [6] developed a Cuckoo Search (CS) algorithm in 2009. CS is inspired by the aggressive breeding behavior of the beautiful sound-making cuckoo bird. Cuckoo birds do not build their nests. Hence, they lay eggs in the nests of other host birds – with similar matching eggs. On the occasion when the host bird detects cuckoo eggs, it either destroys the cuckoo eggs or abandon the nest. It is, therefore, a cuckoo bird is always in search of the host nests where its eggs are highly likely to hatch. Once found unsuitable host nests, a cuckoo will search any other destination to lay more eggs.

The most popular CS algorithm employs a Lévy flight random walk [6], [18] which effectively enhances search efficiency.

### A. *Lévy Flights*

Different types of random walks can be used to enforce diversity in search mechanism of any metaheuristic algorithm. Lévy flights are also random walks based on step size derived from Lévy distribution. The Lévy flights effectively represent random moves performed by animals and insects, as compared to other random distribution methods.

### B. *Cuckoo Search Algorithm*

CS is one of the popular nature-inspired metaheuristic algorithms, which is developed on the reproduction system of the cuckoo bird. In CS, each cuckoo lays one egg in a nest at a time, and the egg represents solution vector. The basic principles behind the design of the CS algorithm are as following:

- The number of cuckoo eggs and host nests is equal,
- Each cuckoo lays one egg at a time in a randomly selected host nest,
- The eggs with a high probability of survival are carried to next generations,
- The host bird can discover the cuckoo bird's egg with probability $p_a \in [0,1]$.

The last rule implements exploration in CS as a certain percentage of solutions are abandoned, and new random solutions are generated to endorse population diversity. The quality of the solution is represented by the fitness value of

the objective function, whereas a solution in CS is a cuckoo egg, nest, or cuckoo itself. Last rule aims to replace a bad cuckoo egg, nest, or cuckoo with new and better one.

CS implements a balance between exploration and exploitation with the help of local random walk and the global random search with the help of Lévy flights. This balance is achieved through switching parameter, as mentioned in the last rule discussed above. This strategy makes the algorithm efficient than other famous counterparts. The step-by-step procedure of CS is explained below in Algorithm 1:

ALGORITHM I
CS PROCEDURE

Set parameters $P_a$, $a$, $\lambda$
Initialize $N$ nests
**Repeat**
1. Get a cuckoo (say $i$) by Lévy flights ($a$) and evaluate fitness $F_i$
2. Choose a nest among $N$ (say $b$) randomly
3. IF $F_i$ is better than $F_j$ THEN replace $F_j$ with $F_i$
4. $P_a$ nests are abandoned and replaced with new ones
5. Rank the nests and find the current best nest (global best solution)

**Until** *maximum iterations;*
Return best solution found.

As mentioned in Algorithm 1, CS commences search by initializing a population of host nests. However, before this, control parameters: switching parameter $P_a$, step size $a$, and Lévy flights step length $\lambda$ is set. The population initialization is performed as follows:

$$x_i = lb_i + Rand_i\,(ub_i - lb_i) \tag{1}$$

Where $x_i$, $lb_i$, $ub_i$, and $Rand_i$ are solution representing cuckoo egg in host's nest, lower and upper bounds of the problem domain, and a random variable is drawn from the uniform distribution within the interval [0, 1], respectively. Iteratively, CS covers two steps: first, generating new solutions using Lévy flights and replacing randomly chosen solutions with new ones if better in quality (fitness value), and second, abandoning a portion of old solutions to replace with new randomly generated ones; in other words, switching nests. The solutions are ranked based on fitness value, and only the top $N$ solutions are carried to next generation until a stopping criterion is met.

When generating a solution for next-generation $t$+1, in the first step, CS performs Lévy flight as (2):

$$X_i^{t+1} = x_i + a \oplus Lévy(\lambda) \tag{2}$$

Where $a > 0$ is step size and $x_i$ is existing search location. The value for step size parameter is set carefully according to problem landscape. The product-wise multiplication $\oplus$ is same as PSO. The Lévy flight is a random walk while the step length is drawn from Lévy distribution as (3):

$$Le'vy(\lambda) = \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{S^{1+\lambda}} \tag{3}$$

Where $\Gamma$ is a Gamma function representing random step length and $s$ is step size. $\lambda$ is random step length. The solution generation using Lévy flight is explained in detail via (4):

$$StepSize_i = 0.01.\left(\frac{Le'vy(\lambda).Rand_1}{Rand_2}\right)^{\frac{1}{\lambda}}.(x_i - X_{gbest}),$$
$$x_{new} = x_i + StepSize_i.Rand_3 \tag{4}$$

where $StepSize_i$ is calculated using Lévy flight and current global best solution $X_{gbest}$. $x_i$ is $i$th solution. $Rand_1$, $Rand_2$, and $Rand_3$ are three different random variables generated using Gaussian distribution. $x_{new}$ is the new solution generated which is then evaluated with the existing solution. If the fitness value of the new solution is better than the existing one then it is replaced with the new one; otherwise, it remains as is.

In the second step, a portion of solutions is abandoned with the new ones generated using (5):

$$x_{new} = x_i + Rand_1 \oplus H(p_a - Rand_2) \oplus (x_j - x_k) \tag{5}$$

Where $x_i$ is a current solution, $x_j$ and $x_k$ are two different solutions chosen randomly, $H(u)$ is the Heaviside function, $Rand_1$ and $Rand_2$ are two different random numbers generated with uniform distribution, and lastly, $P_a$ is the probability of abandoning the solution. After this step, all the solutions are ranked to find the global best solution. This global best solution is then used in the first step as shown in (4). Fig. 1 illustrates the step-by-step procedure of the CS algorithm.

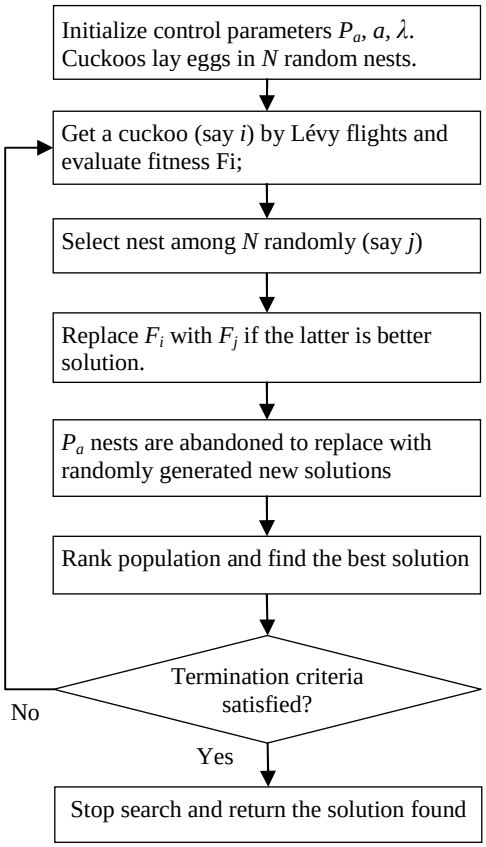


Fig. 1 CS step by step procedure

Despite wider range of applications, CS algorithm still needs improvement in search strategy as it suffers from imbalanced exploration and exploitation when the problem becomes complex [19]. This also causes unstable convergence. To address this, various modifications and hybrids have been proposed, as discussed in the previous section. However, it is often observed that these modifications increase the complexity of the algorithm. Keeping this in view, the current study proposes a simple modification, yet the search efficiency is improved. The next subsection discusses the proposed modification in the CS algorithm.

### C. Proposed Personal Best Cuckoo Search Algorithm

The standard CS algorithm utilizes the global best solution, which is carried to the next generation. This is similar to PSO, which maintains social memory. However, unlike PSO, CS does not maintain the personal memory of a population individual. CS replaces the current solution if the new solution is better than the existing solution, which means there is no personal memory of an individual. While creating a new solution, both in Lévy flights step (4) and solution-switching step (5), CS generates new solutions from the information of existing solutions with some randomization methods. This may result in a lack of diversity due to limited information about the search space visited so far. Moreover, switching parameter $P_a$ is fixed in existing CS, which means a fixed number of unpromising solutions to be abandoned. This parameter is crucial for dynamically adjusting convergence rate as the iterations proceed.

The drawbacks of the existing CS algorithm discussed above are addressed in the proposed modification in this section. To deal with lack of information about the search environment, the modified CS algorithm maintains the personal best memory of a population individual. This information is then utilized to generate new solutions both the solution generation steps in CS algorithm. Because the proposed modified variant utilizes personal best information, hence it is named as *Personal Best Cuckoo Search* (pBestCS) algorithm. The proposed variations in equations (4) and (5) are mathematically expressed as (6) and (7):

$$StepSize_i = 0.01.\left(\frac{Le'vy(\lambda).Rand_1}{Rand_2}\right)^{\frac{1}{\lambda}}.((x_i - X_{gbest}) + (pBest_i - x_i)) \quad (6)$$

$$x_{new} = x_i + (pBest_i - x_i) + Rand_1 \oplus H(p_a - Rand_2) \oplus (p_i - x_k) \quad (7)$$

where $pBest_i$ is the personal best memory of a population individual $i$.

Apart from injecting personal best information in solution generation equations of CS, the proposed pBestCS also dynamically adjusts abandon rate or switching parameter $P_a$ along with iterations, as inspired from inertia weight adjustment in PSO [20]. The switching parameter is high in initial iterations, and as the iterations proceed, it linearly reduces to minimum abandonment rate (8). Therefore, pBestCS introduces additional parameters ($maxP_a$ and $minP_a$).

$$p_a = \max p_a - \frac{\max p_a - \min p_a}{its_{\max}} \times k \quad (8)$$

where $maxp_a$ and $minp_a$ are maximum and minimum abandon rates respectively. $iter_{max}$ and $k$ are maximum iterations and current iteration number respectively. The personal best information is embedded in the proposed CS variant to endorse exploitation, whereas the dynamically updating switching parameter is for exploration purpose. These two features help pBestCS maintain balance between exploration and exploitation, which is crucial to any efficient optimization algorithm.

## III. Results and Discussion

### A. Experimental Settings

In order to validate the proposed modification in CS, pBestCS is tested on a suite of benchmark test functions, which comprises of unimodal and multimodal functions. The unimodal functions have one global optimum with no or one local optimal location, whereas multimodal functions maintain single global optimum solution hidden among several local or suboptimum solutions. The unimodal functions are employed in experiments to test exploitation capability, whereas exploration capability of the algorithm is verified using multimodal functions. Table 1 lists six benchmark test functions used in this research. The range of the search environment and theoretical optimum solutions are also presented. From $F_1$ to $F_3$ are unimodal functions, whereas from $F_4$ to $F_6$ are multimodal functions.

Dimension size is crucial to metaheuristic performance, as the curse of dimensionality is often a challenge for the optimization algorithms. The size of search-space is directly proportional to the dimension size of the optimization problem at hand. For practical evaluation, all the test functions were solved with a variety of dimensions: 10, 30, and 50 dimensions. The experiments were run 30 times and the results are averaged to present fair comparison. Apart from mean, best, worst, and standard deviation of objective functions values achieved over 30 runs are presented in results and discussion subsection. For all the algorithms, the maximum number of iterations was 1500. The experimental settings including test functions, number of dimensions, and the number of experimental runs are taken from the commonly used settings suggested in [21].

We used the Friedman test to determine if the proposed pBestCS performed significantly different from the counterparts. A null hypothesis suggests that the two algorithms performed almost similar; otherwise, the hypothesis is rejected. We used significance level 0.95 ($a$=0.05) for the Friedman test.

The parameters settings of each algorithm are presented in Table 2. In all cases, the population size for CS, pBCS, and PSO were 25, except for ABC with 50 as total neighborhood solutions. The initial population in every run was generated using uniformly distributed random initialization within the search range specified in Table 1.

TABLE I
BENCHMARK TEST FUNCTIONS USED IN EXPERIMENTS

| Function | Formula | Search Range | Theoretical Optimum |
|---|---|---|---|
| Sphere | $F_1(x)=\sum_{i=1}^{D} x_i^2$ | $[-100, 100]^D$ | 0 |
| Quartic | $F_2(x)=\sum_{i=1}^{D} i x_i^4$ | $[-1.28, 1.28]^D$ | 0 |
| Schwefel 2.22 | $F_3(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\sum_{i=1}^{D}\lvert x_i\rvert$ | $[-10, 10]^D$ | 0 |
| Ackley | $F_4(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(cx_i)\right)+a+\exp(1)$ | $[-32, 32]^D$ | 0 |
| Rastrigin | $F_5(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right)$ | $[-5.12, 5.12]^D$ | 0 |
| Griewank | $F_6(x)=\frac{1}{4000}\sum_{i=1}^{D} x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$ | $[-600, 600]^D$ | 0 |

TABLE II
ALGORITHM-SPECIFIC PARAMETER SETTINGS

| Algorithm | Parameters |
|---|---|
| CS | $N$ = 25 (Number of nests)<br>$P_a$ = 0.25 (Switching parameter)<br>$a$ = 0.01 (Lévy flights step length) |
| PSO | $N$ = 25 (Number of particles)<br>$C_1$ = 2 (Cognitive factor)<br>$C_2$ = 2 (Social factor)<br>$W_{max}$ = 0.9 (Maximum inertia weight)<br>$W_{min}$ = 0.4 (Minimum inertia weight) |
| ABC | $N$ = 25 (Number of bees)<br>*Limit* = $D$ X $N$ (Bee switching parameter) |
| pBestCS | $N$ = 25 (Number of nests)<br>$a$ = 0.01 (Lévy flights step length)<br>$maxP_a$ = 0.4 (Maximum switching rate)<br>$minP_a$ = 0.1 (Minimum switching rate) |

The results of the experiments are then compared with standard CS, PSO, and ABC.

### *B. Experimental Results*

In the experiments mentioned above, the proposed pBestCS algorithm was evaluated. The experimental results are reported and compared to standard CS, PSO, and ABC. The statistical results in the form of mean, best, worst, and standard deviation of the objective function values obtained over 30 independent runs are presented in Tables 3 and 4 for dimensions 10, 30, and 50. In these tables, the best results are bold-faced. The performances of the metaheuristic algorithms have been ranked in Table 5 for comparison purpose. These performances are reported dimension-wise while the overall performance rank is also given to summarize the results. Apart from experimental results, some statistical analysis has also been performed via Friedman tests with 5% significance level (or $a$=0.05), to determine the performance of the proposed pBestCS. The *p*-values of the *t*-tests on results are reported in Table 6.

From Table 3 and Table 4, averagely pBestCS outperformed PSO, ABC, and standard CS in most of the problems; only exceptions are $F_1$, $F_3$, and $F_4$ with 10 dimensions where PSO achieved best results. This is consistent with typical findings in the literature that PSO is good on low dimensional problems. According to Table 3, pBestCS yielded significantly better results on $F_1$ and $F_3$ (30 and 50 dimensions). Table 4 also suggests that, except for $F_4$ (10 dimensions), pBestCS achieved far better objective function values on multimodal test functions with 10, 30, and 50 dimensions. The dimension-wise performance ranking is summarized in Table 5 where the performance of pBestCS, CS, PSO, and ABC are ranked on mean objective function values obtained over 30 independent runs. From Table 5, we can say that pBestCS generally achieved top position in the experiments.

To better validate the results of pBestCS, Table 6 provides statistical evidence regarding *p*-value obtained from a *t*-test of Friedman test against the counterpart algorithms. According to the *p*-value, in Table 6, we can conclude that pBestCS has a significant difference from ABC on optimization problems with 10 dimensions, whereas it is contrary in case of CS and PSO. However, on rest of the cases, the performance of pBestCS is significantly different from CS, PSO, and ABC.

To summarize the results discussed above, it can be contended that the proposed pBestCS algorithm is suitable for high dimensional problems, which indicates its ability of finding global optimum solution in large and complex optimization problems. This is further evident in Fig. 2, Fig. 3, and Fig. 4 that pBestCS maintains better convergence ability then the counterparts. Especially, in case of $F_1$ (30 and 50 dimensions), $F_2$ and $F_3$ (50 dimensions), $F_4$ (30 and

50 dimensions), $F_5$ and F6 (10, 30, and 50 dimensions), pBestCS scaped local optimal locations and converged to global optimum points quickly after few initial iterations. Hence, we can say that pBestCS has strong exploration ability and is robust algorithm on large and complex optimization problems.

TABLE III
PERFORMANCE COMPARISON OF METAHEURISTIC ALGORITHMS ON UNIMODAL FUNCTIONS

| Function | Dimensions | Algorithms | Best | Worst | Mean | Std. Dev. |
|---|---|---|---|---|---|---|
| $F_1$ | 10 | CS | 6.94E-27 | 4.14E-25 | 1.87E-26 | 1.65E-26 |
| | | PSO | 2.79E-37 | 2.90E-36 | **2.18E-36** | 1.41E-36 |
| | | ABC | 2.37E-08 | 9.12E-07 | 1.25E-07 | 4.31E-08 |
| | | pBestCS | 1.10E-30 | 6.27E-25 | 5.61E-25 | 1.24E-22 |
| | 30 | CS | 6.82E-09 | 5.96E-08 | 2.31E-08 | 8.09E-09 |
| | | PSO | 6.85E-08 | 1.48E-06 | 5.55E-07 | 1.51E-07 |
| | | ABC | 8.53E-07 | 1.03E-06 | 9.83E-07 | 7.49E-08 |
| | | pBestCS | 9.47E-31 | 1.01E-16 | **2.48E-18** | 1.74E-18 |
| | 50 | CS | 1.32E-04 | 2.37E-04 | 1.68E-04 | 4.28E-05 |
| | | PSO | 1.23E-02 | 1.00E+02 | 6.58E+01 | 4.75E+01 |
| | | ABC | 7.97E-05 | 2.40E-04 | 1.23E-04 | 6.22E-05 |
| | | pBestCS | 3.72E-15 | 5.39E-12 | **1.19E-12** | 2.47E-13 |
| $F_2$ | 10 | CS | 1.83E-03 | 6.17E-03 | 4.64E-03 | 1.88E-03 |
| | | PSO | 2.09E-03 | 4.12E-03 | 2.46E-03 | 8.98E-04 |
| | | ABC | 9.42E-03 | 1.10E-02 | 1.04E-02 | 8.60E-04 |
| | | pBestCS | 5.38E-04 | 1.06E-03 | **8.32E-04** | 2.17E-04 |
| | 30 | CS | 3.32E-02 | 7.72E-02 | 6.31E-02 | 2.99E-02 |
| | | PSO | 2.09E-03 | 4.12E-03 | 2.68E-03 | 5.98E-04 |
| | | ABC | 5.07E-02 | 7.15E-02 | 6.02E-02 | 6.87E-03 |
| | | pBestCS | 9.33E-04 | 4.41E-03 | **2.14E-03** | 1.61E-03 |
| | 50 | CS | 2.39E-01 | 3.04E-01 | 3.79E-01 | 9.88E-02 |
| | | PSO | 3.09E-01 | 1.34E+02 | 4.34E+01 | 3.32E+01 |
| | | ABC | 2.10E-01 | 2.61E-01 | 2.47E-01 | 1.33E-02 |
| | | pBestCS | 5.43E-04 | 1.01E-02 | **2.93E-03** | 2.20E-03 |
| $F_3$ | 10 | CS | 2.09E-12 | 9.34E-12 | 5.00E-12 | 3.09E-12 |
| | | PSO | 5.23E-20 | 3.86E-21 | **1.92E-20** | 1.31E-20 |
| | | ABC | 2.38E-05 | 3.21E-05 | 2.18E-05 | 3.46E-06 |
| | | pBestCS | 5.44E-19 | 8.24E-12 | 2.71E-14 | 3.88E-12 |
| | 30 | CS | 3.13E-04 | 5.28E-04 | 3.92E-04 | 9.55E-05 |
| | | PSO | 3.47E-05 | 1.00E+01 | 6.66E+00 | 5.78E+00 |
| | | ABC | 5.28E-05 | 1.35E-04 | 2.89E-05 | 1.37E-05 |
| | | pBestCS | 1.53E-13 | 1.77E-11 | **2.96E-12** | 2.11E-12 |
| | 50 | CS | 1.10E-02 | 2.87E-02 | 1.96E-02 | 7.23E-03 |
| | | PSO | 3.00E+01 | 6.50E+01 | 4.34E+01 | 1.55E+01 |
| | | ABC | 8.00E-03 | 2.15E-02 | 1.93E-02 | 5.86E-03 |
| | | pBestCS | 1.42E-08 | 1.36E-05 | **5.66E-06** | 3.75E-06 |

TABLE IV
PERFORMANCE COMPARISON OF METAHEURISTIC ALGORITHMS ON MULTIMODAL FUNCTIONS

| Function | Dimensions | Algorithms | Best | Worst | Mean | Std. Dev. |
|---|---|---|---|---|---|---|
| $F_4$ | 10 | CS | 5.19E-12 | 9.42E-12 | 6.93E-12 | 1.86E-12 |
| | | PSO | 2.66E-15 | 6.22E-15 | **3.58E-15** | 1.69E-15 |
| | | ABC | 3.90E-04 | 7.30E-04 | 1.95E-04 | 1.38E-04 |
| | | pBestCS | 9.77E-15 | 1.02E-13 | 4.16E-14 | 1.26E-14 |
| | 30 | CS | 2.17E-04 | 6.74E-04 | 4.57E-04 | 1.13E-04 |
| | | PSO | 8.01E-05 | 2.24E-03 | 2.56E-04 | 9.69E-04 |
| | | ABC | 1.13E-03 | 2.13E-03 | 1.25E-03 | 5.27E-04 |
| | | pBestCS | 3.20E-11 | 8.25E-11 | **6.29E-11** | 2.21E-11 |
| | 50 | CS | 9.12E-01 | 1.61E+00 | 1.39E+00 | 3.11E-01 |
| | | PSO | 5.85E+00 | 8.79E+00 | 6.70E+00 | 2.31E+00 |
| | | ABC | 1.26E-02 | 1.98E-02 | 1.56E-02 | 3.04E-03 |
| | | pBestCS | 5.27E-11 | 1.58E-08 | **1.18E-09** | 6.44E-09 |
| $F_5$ | 10 | CS | 2.88E+00 | 6.06E+00 | 3.65E+00 | 1.36E+00 |
| | | PSO | 3.98E+00 | 4.97E+00 | 4.52E+00 | 4.69E-01 |
| | | ABC | 2.25E+01 | 2.41E+01 | 2.31E+01 | 6.53E-01 |
| | | pBestCS | 0.00E+00 | 5.08E-07 | **1.67E-07** | 2.36E-07 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 30 | CS | 4.66E+01 | 9.30E+01 | 6.26E+01 | 1.16E+01 |
| | | PSO | 6.51E+01 | 8.15E+01 | 6.54E+01 | 8.75E+00 |
| | | ABC | 1.83E+02 | 1.90E+02 | 1.95E+02 | 1.24E+01 |
| | | pBestCS | 0.00E+00 | 6.05E-04 | **1.24E-06** | 1.81E-03 |
| | 50 | CS | 1.32E+02 | 1.47E+02 | 1.38E+02 | 2.88E+00 |
| | | PSO | 2.90E+02 | 4.49E+02 | 3.96E+02 | 7.37E+01 |
| | | ABC | 4.17E+02 | 4.43E+02 | 4.41E+02 | 1.17E+01 |
| | | pBestCS | 0.00E+00 | 1.58E-12 | **1.95E-12** | 1.69E-12 |
| $F_6$ | 10 | CS | 2.01E-02 | 5.78E-02 | 4.32E-02 | 1.53E-02 |
| | | PSO | 7.87E-02 | 1.61E-01 | 1.30E-01 | 3.41E-02 |
| | | ABC | 2.52E-01 | 2.88E-01 | 1.61E-01 | 1.55E-02 |
| | | pBestCS | 8.77E-08 | 2.65E-05 | **1.16E-06** | 8.22E-05 |
| | 30 | CS | 6.28E-04 | 1.27E-02 | 8.84E-03 | 5.09E-03 |
| | | PSO | 7.40E-03 | 1.12E-02 | 1.52E-02 | 3.53E-03 |
| | | ABC | 1.10E-02 | 6.12E-02 | 3.08E-02 | 2.06E-02 |
| | | pBestCS | 0.00E+00 | 0.00E+00 | **0.00E+00** | 0.00E+00 |
| | 50 | CS | 1.42E-02 | 5.50E-02 | 5.40E-02 | 1.92E-02 |
| | | PSO | 1.81E+02 | 2.71E+02 | 2.70E+02 | 4.24E+01 |
| | | ABC | 1.09E-01 | 2.49E-01 | 1.62E-01 | 6.29E-02 |
| | | pBestCS | 0.00E+00 | 2.53E-08 | **1.01E-09** | 1.19E-08 |

TABLE V
PERFORMANCE RANK OF METAHEURISTICS ON TEST FUNCTIONS

| Function | Dimension | CS | PSO | ABC | pBestCS |
|---|---|---|---|---|---|
| $F_1$ | 10 | 3 | 1 | 4 | 2 |
| $F_2$ | | 4 | 3 | 2 | 1 |
| $F_3$ | | 2 | 4 | 3 | 1 |
| $F_4$ | | 3 | 1 | 4 | 2 |
| $F_5$ | | 3 | 4 | 2 | 1 |
| $F_6$ | | 2 | 3 | 4 | 1 |
| Mean Rank | | 2.83 | 2.67 | 3.17 | **1.33** |
| $F_1$ | 30 | 3 | 2 | 4 | 1 |
| $F_2$ | | 3 | 4 | 2 | 1 |
| $F_3$ | | 3 | 4 | 2 | 1 |
| $F_4$ | | 3 | 4 | 2 | 1 |
| $F_5$ | | 2 | 3 | 4 | 1 |
| $F_6$ | | 3 | 4 | 2 | 1 |
| Mean Rank | | 2.83 | 3.50 | 2.67 | **1.00** |
| $F_1$ | 50 | 3 | 4 | 2 | 1 |
| $F_2$ | | 3 | 4 | 2 | 1 |
| $F_3$ | | 3 | 4 | 2 | 1 |
| $F_4$ | | 3 | 4 | 2 | 1 |
| $F_5$ | | 2 | 3 | 4 | 1 |
| $F_6$ | | 3 | 4 | 2 | 1 |
| Mean Rank | | 2.83 | 3.83 | 2.33 | **1.00** |
| Overall Rank | | 2.83 | 3.33 | 2.72 | **1.11** |

TABLE VI
P-VALUES AT A = 0.05 BY FRIEDMAN TEST ON TEST FUNCTIONS

| Dimensions | pBestCS vs CS | pBestCS vs PSO | pBestCS vs ABC |
|---|---|---|---|
| 10 | 0.3545 | 0.7576 | 0.0308 |
| 30 | 0.0308 | 0.0136 | 0.0308 |
| 50 | 0.0136 | 0.0051 | 0.0136 |

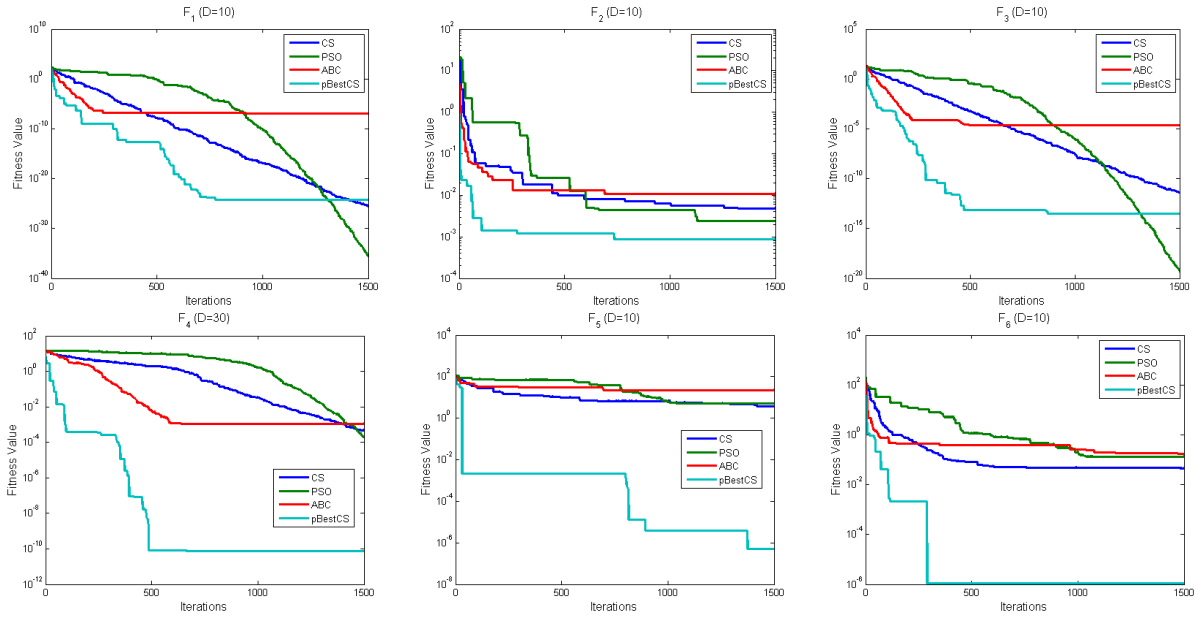


Fig. 2 Convergence comparison of CS, PSO, ABC, and pBestCS on 10 dimensional problems

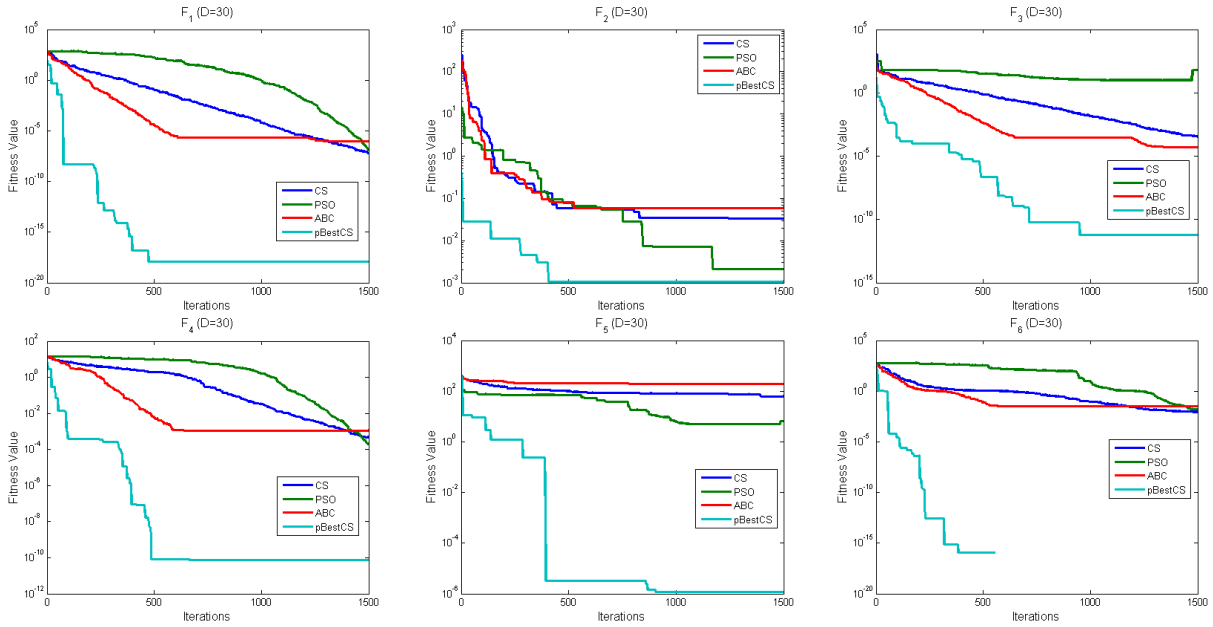


Fig. 3 Convergence comparison of CS, PSO, ABC, and pBestCS on 30 dimensional problems

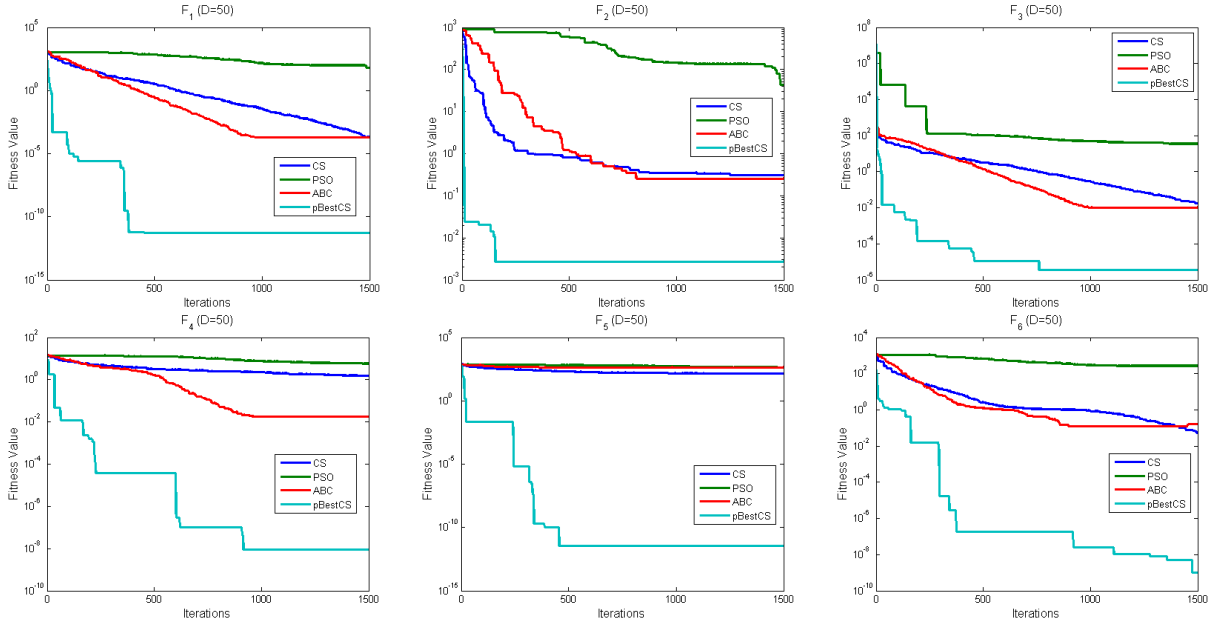


Fig. 4 Convergence comparison of CS, PSO, ABC, and pBestCS on 50 dimensional problems

## IV. Conclusions

CS algorithm is a recent addition to nature-inspired population-based metaheuristic algorithms, which have performed well in hard optimization problems. However, similar to other metaheuristic algorithms, CS also encounters performance drawbacks when it is implemented in large and complex optimization problems. Various modifications have been introduced in literature, but mostly such modifications are proposed by compromising algorithm complexity. In this paper, the simple and effective modification has been proposed in CS to solve unbalanced exploration and exploitation problem. The improved variant, so-called pBestCS algorithm employs personal best information – inspired from PSO – in the process of generating new neighborhood solutions. The personal best memory or information is embedded in both in

Lévy flights and solution switching steps of the CS algorithm. This modification enforces effective local search ability. On the other hand, the proposed pBestCS also employs the strategy of dynamically adjusting switching parameter to endorse high exploration in the beginning, which is linearly reduced towards the end of search process. The proposed modification resolves the problem of lack of population diversity in CS algorithm.

The comprehensive analysis of experimental results on both unimodal and multimodal test function with a variety of dimensionality validate the efficiency of the proposed pBestCS algorithm. Based on comparison with the standard CS, as well as, with other popular swarm-based metaheuristic algorithms PSO and ABC, it can be implied that pBestCS has improved search results than the peer algorithms tested in this study.

## ACKNOWLEDGMENT

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

## REFERENCES

[1] Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 267(1), 66-73.

[2] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.

[3] Shi, Y., & Eberhart, R. (1998, May). A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Conference on* (pp. 69-73). IEEE.

[4] Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* (Vol. 2, pp. 1470-1477). IEEE.

[5] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3), 459-471.

[6] Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (pp. 210-214). IEEE.

[7] Shehab, M., Khader, A. T., & Al-Betar, M. A. (2017). A survey on applications and variants of the cuckoo search algorithm. *Applied Soft Computing*.

[8] Mareli, M., & Twala, B. (2017). An adaptive Cuckoo search algorithm for optimisation. *Applied Computing and Informatics*.

[9] Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with computers*, 29(1), 17-35.

[10] Gandomi, A. H., Talatahari, S., Yang, X. S., & Deb, S. (2013). Design optimization of truss structures using cuckoo search algorithm. *The Structural Design of Tall and Special Buildings*, 22(17), 1330-1349.

[11] Abdelaziz, A. Y., & Ali, E. S. (2015). Cuckoo search algorithm based load frequency controller design for nonlinear interconnected power system. *International Journal of Electrical Power & Energy Systems*, 73, 632-643.

[12] T. Wang, M. Meskin and I. Grinberg, "Comparison between particle swarm optimization and Cuckoo Search method for optimization in unbalanced active distribution system," *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, 2017, pp. 14-19.

[13] Li, X., & Yin, M. (2016). A particle swarm inspired cuckoo search algorithm for real parameter optimization. *Soft Computing*, 20(4), 1389-1413.

[14] Ong, P. (2014). Adaptive cuckoo search algorithm for unconstrained optimization. *The Scientific World Journal*, 2014.

[15] Shehab, M., Khader, A. T., Al-Betar, M. A., & Abualigah, L. M. (2017, May). Hybridizing cuckoo search algorithm with hill climbing for numerical optimization problems. In *Information Technology (ICIT), 2017 8th International Conference on* (pp. 36-43). IEEE.

[16] Chi, R., Su, Y. X., Zhang, D. H., Chi, X. X., & Zhang, H. J. (2017). A hybridization of cuckoo search and particle swarm optimization for solving optimization problems. *Neural Computing and Applications*, 1-18.

[17] Abdel-Baset, M., & Hezam, I. (2016). Cuckoo search and genetic algorithm hybrid schemes for optimization problems. *Appl. Math*, 10(3), 1185-1192.

[18] Yang, X. S., & Deb, S. (2010). Engineering optimization by cuckoo search. International *Journal of Mathematical Modelling and Numerical Optimisation*, 1(4), 330-343.

[19] Li, X., & Yin, M. (2016). A particle swarm inspired cuckoo search algorithm for real parameter optimization. *Soft Computing*, 20(4), 1389-1413.

[20] Bansal, J. C., Singh, P. K., Saraswat, M., Verma, A., Jadon, S. S., & Abraham, A. (2011, October). Inertia weight strategies in particle swarm optimization. In *Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on* (pp. 633-640). IEEE.

[21] Hussain, K., Salleh, M. N. M., Cheng, S., & Naseem, R. (2017). Common benchmark functions for metaheuristic evaluation: A review. *JOIV: International Journal on Informatics Visualization*, 1(4-2), 218-223.

[22] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

[23] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[24] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.

[25] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.

[26] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[27] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[28] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.