VOL. 11, NO. 22, NOVEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ENHANCING THE CUCKOO SEARCH WITH LEVY FLIGHT THROUGH POPULATION ESTIMATION

Nazri Mohd Nawi, Shah Liyana Shahuddin, Muhammad Zubair Rehman and Abdullah Khan
Soft Computing and Data Mining Centre, Faculty of Computer Science and Information Technology, Malaysia
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor Darul Takzim, Malaysia
E-Mail: nazri@uthm.edu.my

ABSTRACT

This paper proposed the use of population estimation in a new meta-heuristic called Cuckoo search (CS) algorithm to minimize the training error, achieve fast convergence rate and to avoid local minimum problem. The CS algorithm which imitates the cuckoo bird's search behavior for finding the best nest has been applied independently to solve several engineering design optimization problems based on cuckoo bird's behavior. The algorithm is tested on five benchmark functions such as Ackley function, Griewank function, Rastrigin function, Rosenbrock function and Schwefel function. The performance of the proposed algorithm was compared with Particle Swarm Optimization (PSO), Wolf Search Algorithm (WSA) and Artificial Bee Colony (ABC). The simulation results show that the CS with Levy flight out performs PSO, WSA and ABC, when the cuckoo population is varied.

Keywords: meta-heuristics, cuckoo search, levy flight, optimization.

INTRODUCTION

A simple way to look at the nature of the search algorithm's is to divide the algorithm into two categories; i.e. Deterministic and stochastic. Deterministic algorithms follow a rigorous procedure and the functions are repeatable like following the same path. While, stochastic algorithm always have some randomness and they do not follow the same path twice.

The word meta-heuristic was coined by Fred Glover, and can be considered as a master strategy that guides and modifies other heuristics to produce solutions. Heuristic means to find or to discover by trial and error(X.-S. Yang, 2010). Generally, meta-heuristics perform better than the simple heuristics and are suitable for global optimization. Many modern meta-heuristic algorithms inspired by nature are emerging and becoming popular. For example, particles swarm optimization (PSO) was inspired by fish and bird swarm intelligence. Meanwhile, the Firefly Algorithm is inspired by the flashing pattern of tropical fireflies. These nature-inspired meta-heuristic algorithms have been used in a wide range of optimization problems.

Recently, a new meta-heuristic search algorithm called Cuckoo search (CS) is made available by Yang and Deb (X. S. Yang and Deb, 2009; X. S. Yang, 2013). CS imitates animal behavior (bird) is proposed to minimize the training error, achieve fast convergence rate and avoid local minimum problem. The CS algorithm has been applied independently to solve several engineering design optimization problems based on cuckoo bird's behavior. Based on the idealized rules of CS followed, when generate a new solution for a Cuckoo, a Levy flight is performed. Various studies have shown that flight behavior of many animals and insect has demonstrated the typical characteristics of Levy Flight.

This paper aims to enhance the Cuckoo Search (CS) with Levy flight through population estimation. The structure of the paper is organized as follows: In the next sections, meta-heuristic search algorithms are discussed. Section 3 shed some light on the simulation results. Section 4 discuss the result comparison of all methods and finally the paper is concluded in the Section 5.The CS algorithm is trained on five benchmark function and compare the proposed search strategy with other popular optimization algorithms. The performance of the algorithm will be discussed in detail throughout this paper.

SWARM INTELLIGENCE (SI)

Swarm intelligence (SI) is briefly defined as the collective behaviors that result from the local interactions of the individuals with each other and with their environment. The examples of swarms are bird flocks, fish schools and the colony of social insects such as termites, ants and bees(D Karaboga and Akay, 2009). According to Millonas in 1994, the swarm must satisfy the following principles to be considered intelligent(Millonas, 1994):

(i) The swarm should be able to do simple space and time computations (the proximity principle).
(ii) The swarm should be able to respond to quality factors in the environment (the quality principle).
(iii) The swarm should not commit its activities along excessively narrow channels (the principle of diverse response).
(iv) The swarm should not change its mode of behavior upon every fluctuation of the environment.
(v) The swarm must be able to change behavior mode when needed (the adaptability principle).

Besides, human also fell into the domain of swarm intelligence, especially some multi-robot systems, and also certain computer programs that are written to tackle optimization and data analysis problems. However, most of the research in this area is inspired from nature, especially biological systems.

Artificial bee colony (ABC)

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga for solving unimodal and multi-modal numerical optimization problems(Dervis Karaboga and Basturk, 2007). The algorithm simulates the intelligent foraging behavior of honey bee swarms. ABC used only common control parameters such as colony size and maximum cycle number. It is a very simple, robust and population based stochastic optimization algorithm. Honey bee swarms consists of three essential components such as food sources, unemployed foragers and employed foragers and the model defines two leading modes of the behavior: the abandonment of a source and recruitment to a nectar source.

In ABC system, artificial bees fly around in a multidimensional search space and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and they will adjust their positions. Some (scouts) fly and choose the food sources randomly without using experience. If the nectar amount of a new source is higher than the previous one, they will memorize the new position and forget the previous one. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees is equal to the number of solutions in the population. At the first step, a randomly distributed initial population (food source positions) is generated. After initialization, the population is subjected to repeat the cycles of the search processes of the employed, onlooker, and scout bees, respectively. After all employed bees complete the search process, they share the position information of the sources with the onlookers on the dance area. Each onlooker evaluates the nectar information taken from all employed bees and then chooses a food source depending on the nectar amounts of sources(D. Karaboga & Basturk, 2008; Dervis Karaboga & Basturk, 2007).

Particle swarm optimization (PSO)

In past several years, PSO has been successfully applied in many research and application areas. Particle swarm optimization (PSO) was developed by Kennedy and Eberhart in 1995 is a population based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling(Kennedy & Eberhart, 1995).The system is initialized with a population of random solutions and searches for optima by updating generations. PSO has no evolution operators such as crossover and mutation.

In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. This value is called p^{best}. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called l^{best}. When a particle takes all the population as its topological neighbors, the best value is a global best and is called g^{best}. The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its p^{best} and l^{best} locations (local version of PSO)(Kulkarni & Venayagamoorthy, 2011). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward p^{best} and l^{best}locations.

Wolf search algorithm (WSA)

Based on wolf preying behavior, a new bio-inspired heuristic optimization algorithm, the Wolf Search Algorithm (WSA) was proposed to solve optimization problems. WSA is different from bio-inspired meta-heuristics because each wolf will hunts independently by remembering its prey and attack their prey when in appropriate conditions. In this way, long-range inter-communication among the wolves that represent the searching points for candidate solutions is eliminated because wolves are known to stalk their prey in silence(Tang, Fong, Yang, & Deb, 2012). Wolves have developed semi cooperative characteristics that is, they move in a group, but tend to take down their prey individually. This detail is important because some optimization algorithms, such as swarm-based, focus on group coordination whereas this algorithm emphasizes individual movements. When hunting, wolves will attempt to conceal themselves when approaching their prey. This searching agents always look for better position in the same way that wolves continuously change their positions for better ones with more shelter, fewer terrain obstacles or less vulnerability.

Cuckoo search (CS)

Many improved learning algorithms have been proposed to overcome the weakness of gradient-based techniques. New meta-heuristic search algorithm, called Cuckoo search (CS) developed by Yang and Deb which imitates animal behavior (bird) and is useful for global optimization. This meta-heuristic search algorithm is proposed to minimize the training error, achieve fast convergence rate and avoid local minimum problem. The CS algorithm hasbeen applied independently to solve several engineering design optimization problems based on cuckoo bird's behavior (X. S. Yang & Deb, 2009).The aim is to use a new and a potentially better solution (cuckoo) to replace a not so good solution in the nests. In the simplest form, each nest has one egg. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions.CS is based on three idealized rules:

1. Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest;
2. The best nests with high quality of eggs will carry over to the next generation;
3. The number of available hosts nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability pa ∈ [0, 1].

The host bird can either throw the egg away or abandon the nest, and build a completely new nest. The rule defined above can be approximated by the fraction pa ∈ [0, 1] of the n nests that are replaced by new nests (with new random solutions) [10]. There are two parameters used in the Cuckoo search algorithm that are the population size, n and probability of discovery of alien eggs, Pa. Once n is changes, Pa will control the elitism and the balance of randomization. Few parameters make an algorithm less complex. For further understanding of the process, refer to the pseudo code for the CS algorithm below:

Step 1: *Generate initial population of n host nests x_i (i = 1, 2,..., n)*
Step 2:
While
(t <MaxGeneration) or (stop criterion)
Step 3: *Get a cuckoo randomly by L´evy flights evaluate its quality/fitness F_i*
Step 4: *Choose a nest among n (say, j) randomly*
Step 5: ***if*** *(F_i>F_j), replace j by the new solution;*
end if
Step 6: *A fraction (p_a) of worse nests are abandoned and new ones are built;*
Step 7: *Keep the best solutions (or nests with quality solutions);*
Step 8: *Rank the solutions and find the current best* ***end while***

Based on three rules of CS, when generate a new solution for a Cuckoo, a Levy flight is performed. Levy Flights are random walk which the steps lengths are heavy-tailed probability distribution. The distance from the origin of the random walk tends to stable distribution after a large number of steps.When generating a new solution x_i^{t+1} for a Cuckoo *i*, a Levy flight is performed as shown in the Equation:

$x_i^{t+1} = x_i^t + \alpha \otimes \text{levy}(\lambda)$

Where α> 0 is the step size, which should be related to the scales of the problem of interest. The product ⊗ means entry wise multiplications. The random walk via Levy flight is more efficient in exploring the search space as its step length is much longer in the long run.

RESULTS AND DISCUSSIONS

In this section, the result and analysis of the simulations will be discussed. The proposed Cuckoo Search algorithm's with Levy Flight algorithm are tested on five benchmark functions such as Ackley function, Griewank function, Rastrigin function, Rosenbrock function and Schwefel function. The performance of the proposed algorithm was compare with Particle Swarm Optimization (PSO), Wolf Search Algorithm (WSA) and Artificial Bee Colony (ABC). The performance evaluation has been carried out based on its average convergence that has the optimal value near to zero.

Each simulations result of CS algorithm will presented in Table form. For more understanding, the performance of CS on each benchmark function have been highlighted with different colors such as green color for best performance (lowest optimal value), blue color for second best performance, yellow color for third best performance and red color for worst performance (highest optimal value). The performance are evaluated based on the standard deviation (SD), standard error of mean (SEM), and mean. The parameters used are number of nests (*n*) and probability of discovery rate of alien eggs (*p*).

Ackley function

Ackley is a widely used multimodal test function and testing optimization algorithms. Table-1 indicates the result of CS algorithm on Ackley function.

Table-1. Result of CS algorithm on Ackley function.

	Mean	SD	SEM
n=10,p=0.05	9.55×10^{-6}	3.83×10^{-7}	5.42×10^{-8}
n=10,p=0.10	9.85×10^{-6}	1.99×10^{-7}	2.82×10^{-8}
n=10,p=0.15	9.77×10^{-6}	2.31×10^{-7}	3.27×10^{-8}
n=10,p=0.20	9.66×10^{-6}	2.70×10^{-7}	3.82×10^{-8}
n=10,p=0.25	9.55×10^{-6}	5.96×10^{-7}	8.43×10^{-8}
n=15,p=0.05	9.82×10^{-6}	1.56×10^{-7}	2.20×10^{-8}
n=15,p=0.10	9.63×10^{-6}	3.33×10^{-7}	4.71×10^{-8}
n=15,p=0.15	9.61×10^{-6}	3.80×10^{-7}	5.37×10^{-8}
n=15,p=0.20	9.47×10^{-6}	4.38×10^{-7}	6.20×10^{-8}
n=15,p=0.25	9.37×10^{-6}	6.60×10^{-7}	9.34×10^{-8}
n=20,p=0.05	9.64×10^{-6}	4.04×10^{-7}	5.71×10^{-8}
n=20,p=0.10	9.46×10^{-6}	7.14×10^{-7}	1.01×10^{-7}
n=20,p=0.15	9.51×10^{-6}	4.73×10^{-7}	6.68×10^{-8}
n=20,p=0.20	9.38×10^{-6}	6.53×10^{-7}	9.23×10^{-8}
n=20,p=0.25	9.50×10^{-6}	4.98×10^{-7}	7.05×10^{-8}
n=25,p=0.05	9.57×10^{-6}	4.44×10^{-7}	6.28×10^{-8}
n=25,p=0.10	9.40×10^{-6}	4.62×10^{-7}	6.53×10^{-8}
n=25,p=0.15	9.44×10^{-6}	5.02×10^{-7}	7.10×10^{-8}
n=25,p=0.20	9.27×10^{-6}	4.99×10^{-7}	7.05×10^{-8}
n=25,p=0.25	9.24×10^{-6}	7.10×10^{-7}	1.00×10^{-7}
n=30,p=0.05	9.46×10^{-6}	5.62×10^{-7}	7.95×10^{-8}
n=30,p=0.10	9.29×10^{-6}	7.18×10^{-7}	1.02×10^{-7}
n=30,p=0.15	9.38×10^{-6}	6.32×10^{-7}	8.94×10^{-8}
n=30,p=0.20	9.59×10^{-6}	6.99×10^{-7}	9.89×10^{-8}
n=30,p=0.25	9.35×10^{-6}	5.91×10^{-7}	8.36×10^{-8}
n=35,p=0.05	9.29×10^{-6}	8.41×10^{-7}	1.19×10^{-7}
n=35,p=0.10	9.29×10^{-6}	7.17×10^{-7}	1.01×10^{-7}
n=35,p=0.15	9.16×10^{-6}	8.17×10^{-7}	1.16×10^{-7}
n=35,p=0.20	9.62×10^{-6}	6.46×10^{-7}	9.13×10^{-8}
n=35,p=0.25	9.28×10^{-6}	7.12×10^{-7}	1.01×10^{-7}
n=40,p=0.05	9.39×10^{-6}	6.03×10^{-7}	8.53×10^{-8}
n=40,p=0.10	9.33×10^{-6}	6.24×10^{-7}	8.82×10^{-8}
n=40,p=0.15	9.29×10^{-6}	8.21×10^{-7}	1.16×10^{-7}
n=40,p=0.20	9.35×10^{-6}	4.44×10^{-7}	6.27×10^{-8}
n=40,p=0.25	9.38×10^{-6}	7.21×10^{-7}	1.02×10^{-7}

Based on Table-1, the best performance of CS algorithm on Ackley function that have the lowest optimal value are on parameter n=35,p=0.15 with 9.16×10^{-6}. The performance in terms of SD and SEM are 8.17×10^{-7} and 1.16×10^{-7} respectively. The second performance are on

parameter n=35,p=0.25 with optimal value 9.21 x 10^{-6}, SD value of 7.12 x 10^{-7} and SEM, 1.01 x 10^{-7}. The third optimal value that shows effective performance are 9.24 x 10^{-6} with the parameter n=25,p=0.25. The performance in terms of SD and SEM are 7.10 x 10^{-7} and 1.00 x 10^{-7} respectively. The highest optimal value where the value not close to zero are on parameter n=10,p=10 with 9.85 x10^{-6}, SD value of 1.99 x10^{-7} and SEM, 2.82 x 10^{-8}.

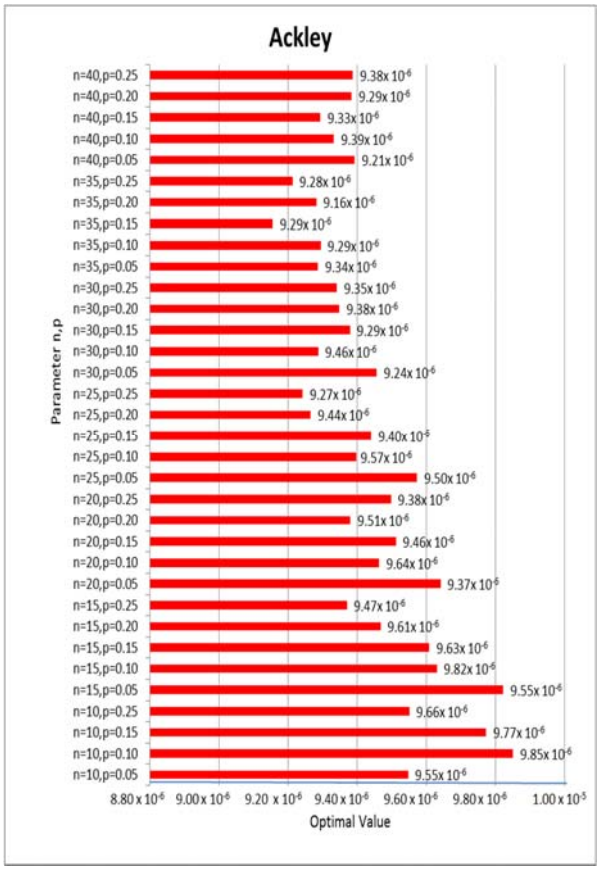

Figure-1. Performance of CS algorithm on Ackley function.

Based on Figure-1, the best performance of CS algorithm on Ackley function that have the lowest optimal value are on parameter n=35,p=0.15 with 9.16 x 10^{-6}. The second performance are on parameter n=35,p=0.25 with optimal value 9.21 x 10^{-6}. The third optimal value that shows effective performance are 9.24 x 10^{-6} with the parameter n=25, p=0.25. The highest optimal value where the value not close to zero are on parameter n=10,p=10 with 9.85 x10^{-6}.

Griewank function

Griewank function is similar to Rastrigin function. It has many widespread local minima. However, the location of the minima is regularly distributed. Table-2 indicates the result of CS algorithm on Griewank function.

Table-2. Result of CS algorithm on Griewank function.

	Mean	SD	SEM
n=10,p=0.05	9.70 x 10^{-6}	3.06 x 10^{-7}	4.33 x 10^{-8}
n=10,p=0.10	9.65 x 10^{-6}	3.43 x 10^{-7}	4.86 x 10^{-8}
n=10,p=0.15	9.59 x 10^{-6}	4.20 x 10^{-7}	5.94x 10^{-8}
n=10,p=0.20	9.48 x 10^{-6}	5.83 x 10^{-7}	8.25 x 10^{-8}
n=10,p=0.25	9.31 x 10^{-6}	9.06 x 10^{-7}	1.28 x 10^{-7}
n=15,p=0.05	9.46 x 10^{-6}	6.04 x 10^{-7}	8.54 x 10^{-8}
n=15,p=0.10	9.24 x 10^{-6}	8.75 x 10^{-7}	1.24 x 10^{-7}
n=15,p=0.15	9.20 x 10^{-6}	8.37 x 10^{-7}	1.18 x 10^{-7}
n=15,p=0.20	9.12 x 10^{-6}	8.93 x 10^{-7}	1.26 x 10^{-7}
n=15,p=0.25	9.10 x 10^{-6}	8.38 x 10^{-7}	1.19 x 10^{-7}
n=20,p=0.05	9.22 x 10^{-6}	9.76 x 10^{-7}	1.38 x 10^{-7}
n=20,p=0.10	9.25 x 10^{-6}	7.98 x 10^{-7}	1.13 x 10^{-7}
n=20,p=0.15	9.17 x10^{-6}	7.36 x 10^{-7}	1.04 x 10^{-7}
n=20,p=0.20	8.96 x10^{-6}	8.71 x 10^{-7}	1.23 x 10^{-7}
n=20,p=0.25	8.87 x 10^{-6}	1.07 x 10^{-6}	1.51 x 10^{-7}
n=25,p=0.05	9.04 x 10^{-6}	9.34 x 10^{-7}	1.32 x 10^{-7}
n=25,p=0.10	8.92 x 10^{-6}	9.43 x 10^{-7}	1.33 x 10^{-7}
n=25,p=0.15	8.96 x 10^{-6}	8.26 x 10^{-7}	1.17 x 10^{-7}
n=25,p=0.20	8.84 x 10^{-6}	1.13 x 10^{-6}	1.60 x 10^{-7}
n=25,p=0.25	8.68 x 10^{-6}	1.26 x 10^{-6}	1.78 x 10^{-7}
n=30,p=0.05	9.14 x 10^{-6}	8.46 x 10^{-7}	1.20 x 10^{-7}
n=30,p=0.10	8.77 x 10^{-6}	1.15 x 10^{-6}	1.63 x 10^{-7}
n=30,p=0.15	8.81 x 10^{-6}	1.33 x 10^{-6}	1.88 x 10^{-7}
n=30,p=0.20	8.74 x 10^{-6}	1.15 x 10^{-6}	1.63 x 10^{-7}
n=30,p=0.25	9.04 x 10^{-6}	8.46 x 10^{-7}	1.20 x 10^{-7}
n=35,p=0.05	8.91 x 10^{-6}	8.65 x 10^{-7}	1.22 x 10^{-7}
n=35,p=0.10	8.92 x 10^{-6}	9.58 x 10^{-7}	1.36 x 10^{-7}
n=35,p=0.15	8.74 x 10^{-6}	1.21 x 10^{-6}	1.71x 10^{-7}
n=35,p=0.20	8.95 x 10^{-6}	1.08 x 10^{-6}	1.52 x 10^{-7}
n=35,p=0.25	9.04 x 10^{-6}	9.26 x 10^{-7}	1.31x 10^{-7}
n=40,p=0.05	9.02 x 10^{-6}	7.83 x 10^{-7}	1.11 x 10^{-7}
n=40,p=0.10	8.68 x 10^{-6}	1.13 x 10^{-6}	1.60 x 10^{-7}
n=40,p=0.15	8.92 x 10^{-6}	8.71 x 10^{-7}	1.23 x 10^{-7}
n=40,p=0.20	8.72 x 10^{-6}	1.32 x 10^{-6}	1.87 x 10^{-7}
n=40,p=0.25	8.99 x 10^{-6}	1.09 x 10^{-6}	1.55 x 10^{-7}

Based on Table-2, the best performance of CS algorithm on Griewank function that have the lowest optimal value are on parameters n=25, p=0.25 and n=40, p=0.10 that have the same value 8.68 x10^{-6}. The performance in terms of SD are 1.26 x 10^{-6} and 1.13 x 10^{-6} respectively. While, SEM value are 1.78x10^{-7} and 1.60x

10^{-7}. The second performance are on parameter n=40,p=0.20 with optimal value 8.72 x10^{-6}, SD value of 1.32 x $10^{-6}$and SEM, 1.87x10^{-7}. The third optimal value that shows effective performance are 8.74 x10^{-6} with the parameter n=35,p=0.15, SD value of 1.21 x 10^{-6} and SEM, 1.71 x 10^{-7}. The highest optimal value where the value not close to zero are on parameter n=10,p=0.05 with 9.70 x10^{-6}, SD value of 3.06 x 10^{-7} and SEM,4.33 x 10^{-8}.

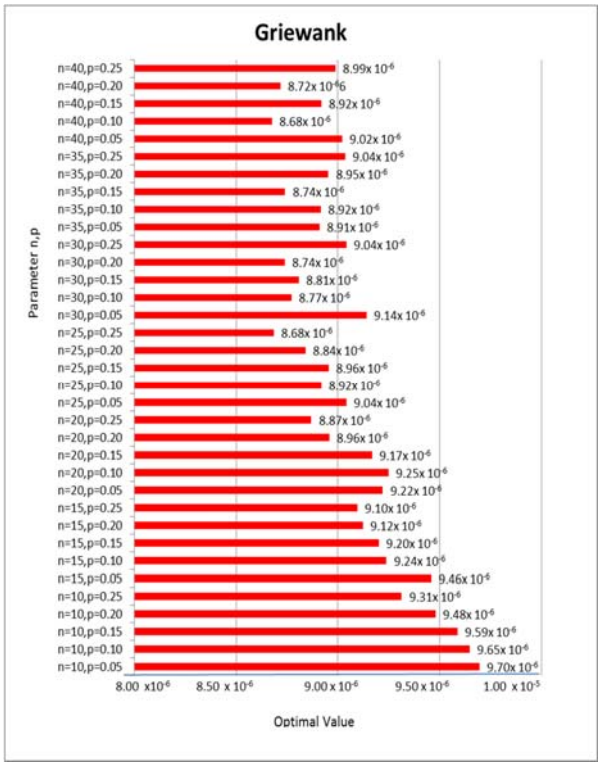

Figure-2. Performance of CS algorithm on Griewank function.

According to Figure-2, the best performance of CS algorithm on Griewank function that have the lowest optimal value are on parameters n=25, p=0.25 and n=40,p=0.10 that have the same value 8.68 x10^{-6}. The second performance are on parameter n=40, p=0.20 with optimal value 8.72 x10^{-6}. The third optimal value that shows effective performance are 8.74 x10^{-6} with the parameter n=35, p=0.15. The highest optimal value where the value not close to zero are on parameter n=10,p=0.05 with 9.70 x10^{-6}.

Rastrigin function

Rastrigin function produces many local minima in its trajectory. Thus, the test function is multimodal. However, the location of the minima is regularly distributed. Table-3 indicates the result of CS algorithm on Rastrigin function.

Based on Table-3, the best performance of CS algorithm on Rastrigin function that have the lowest optimal value that nearest to zero are on parameters n=25,p=0.10 with 8.51 x10^{-6} . The performance in terms of SD and SEM value are 1.44 x 10^{-6} and 2.04 x 10^{-7} respectively. While, the second optimal value that shows effective performance are 8.54 x10^{-6} on parameter n=35,p=0.20, SD value of 1.22 x 10^{-6} and SEM, 1.73x10^{-7}. The third performance are on parameter n=20, p=0.25 with optimal value 8.71 x10^{-6}, SD value of 8.89 x 10^{-7} and SEM, 1.26 x 10^{-7}.The highest optimal value where the value not close to zero are on parameter n=10,p=0.05 with 9.92 x10^{-6}. The performance in terms of SD and SEM value 1.33 x 10^{-7} and 1.88 x 10^{-8} respectively.

Table-3. Result of CS algorithm on Rastrigin function.

	Mean	SD	SEM
n=10,p=0.05	9.92 x 10^{-6}	1.33 x 10^{-7}	1.88 x 10^{-8}
n=10,p=0.10	9.87 x 10^{-6}	2.05 x 10^{-7}	2.90 x 10^{-8}
n=10,p=0.15	9.82 x 10^{-6}	1.86 x 10^{-7}	2.63 x 10^{-8}
n=10,p=0.20	9.74 x 10^{-6}	3.50 x 10^{-7}	4.94 x 10^{-8}
n=10,p=0.25	9.71 x 10^{-6}	2.84 x 10^{-7}	4.02 x 10^{-8}
n=15,p=0.05	9.60 x 10^{-6}	4.16 x 10^{-7}	5.88 x 10^{-8}
n=15,p=0.10	9.37 x 10^{-6}	5.59 x 10^{-7}	7.90 x 10^{-8}
n=15,p=0.15	9.11 x 10^{-6}	9.50 x 10^{-7}	1.34 x 10^{-7}
n=15,p=0.20	9.29 x 10^{-6}	9.98 x 10^{-7}	1.41 x 10^{-7}
n=15,p=0.25	9.28 x 10^{-6}	7.34 x 10^{-7}	1.04 x 10^{-7}
n=20,p=0.05	9.29 x 10^{-6}	7.17 x 10^{-7}	1.01 x 10^{-7}
n=20,p=0.10	8.92 x 10^{-6}	1.17 x 10^{-6}	1.65 x 10^{-7}
n=20,p=0.15	8.96 x 10^{-6}	9.29 x 10^{-7}	1.31 x 10^{-7}
n=20,p=0.20	9.08 x 10^{-6}	7.84 x 10^{-7}	1.11 x 10^{-7}
n=20,p=0.25	8.71 x 10^{-6}	8.89 x 10^{-7}	1.26 x 10^{-7}
n=25,p=0.05	8.98 x 10^{-6}	1.02 x 10^{-6}	1.45 x 10^{-7}
n=25,p=0.10	8.51 x 10^{-6}	1.44 x 10^{-6}	2.04 x 10^{-7}
n=25,p=0.15	9.05 x 10^{-6}	7.49 x10^{-7}	1.06 x 10^{-7}
n=25,p=0.20	8.94 x 10^{-6}	9.27 x 10^{-7}	1.31 x 10^{-7}
n=25,p=0.25	8.88 x 10^{-6}	1.06 x 10^{-6}	1.50 x 10^{-7}
n=30,p=0.05	8.79 x 10^{-6}	1.06 x 10^{-6}	1.49 x 10^{-7}
n=30,p=0.10	8.92 x 10^{-6}	9.33 x 10^{-7}	1.32 x 10^{-7}
n=30,p=0.15	8.79 x 10^{-6}	1.16 x 10^{-6}	1.65 x 10^{-7}
n=30,p=0.20	8.89 x 10^{-6}	9.96 x 10^{-7}	1.41 x 10^{-7}
n=30,p=0.25	8.72 x 10^{-6}	1.20 x 10^{-6}	1.70 x 10^{-7}
n=35,p=0.05	8.95 x 10^{-6}	9.26 x 10^{-7}	1.31 x 10^{-7}
n=35,p=0.10	8.94 x 10^{-6}	9.72 x 10^{-7}	1.37 x 10^{-7}
n=35,p=0.15	8.82 x 10^{-6}	1.08 x 10^{-6}	1.53 x 10^{-7}
n=35,p=0.20	8.54 x 10^{-6}	1.22 x 10^{-6}	1.73 x 10^{-7}
n=35,p=0.25	8.91 x 10^{-6}	9.51 x 10^{-7}	1.34 x 10^{-7}
n=40,p=0.05	8.80 x 10^{-6}	1.11 x 10^{-6}	1.57 x 10^{-7}
n=40,p=0.10	8.86 x 10^{-6}	8.88 x 10^{-7}	1.26 x 10^{-7}
n=40,p=0.15	8.84 x 10^{-6}	1.23 x 10^{-6}	1.74 x 10^{-7}
n=40,p=0.20	8.82 x 10^{-6}	1.22 x 10^{-6}	1.72 x 10^{-7}
n=40,p=0.25	8.83 x 10^{-6}	1.06 x 10^{-6}	1.50 x 10^{-7}

Based on Figure-3, the best performance of CS algorithm on Rastrigin function that have the lowest optimal value that nearest to zero are on parameters n=25,p=0.10 with 8.51 x10^{-6}. The second optimal value that shows effective performance are 8.54 x10^{-6} on parameter n=35, p=0.20. The third performance are on parameter n=20, p=0.25 with optimal value 8.71 x10^{-6}.The highest optimal value where the value not close to zero are on parameter n=10, p=0.05 with 9.92 x10^{-6}.

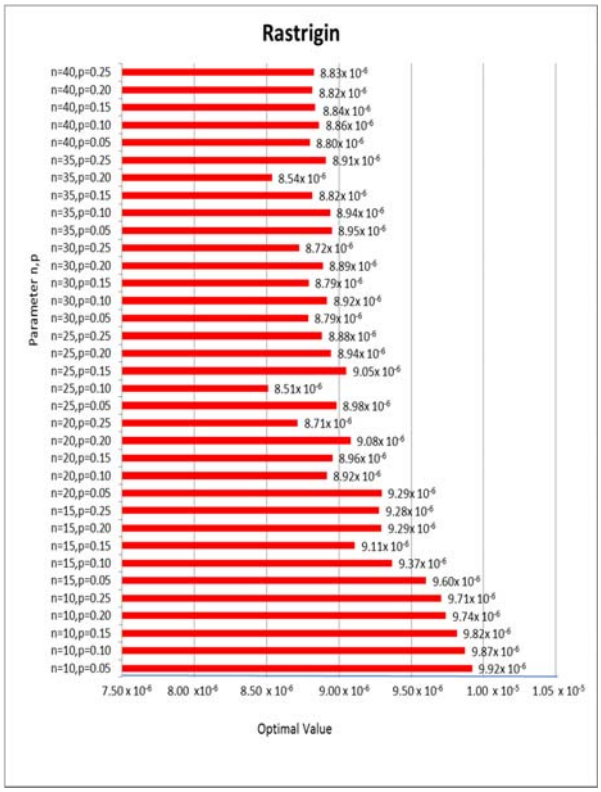

Figure-3. Performance of CS algorithm on Rastrigin function.

Rosenbrock function

The Rosenbrock function, also referred to as the Valley or Banana function is a popular test problem for gradient-based optimization algorithms. The function is unimodal, and the global minimum lies in a narrow, parabolic valley. Table-4 indicates the result of CS algorithm on Rosenbrock function.

Based on Table-4, the best performance of CS algorithm on Rosenbrock function that have the lowest optimal value that nearest to zero are on parameters n=25, p=0.20 with 8.44 x10^{-6}. The performance in terms of SD and SEM value are 1.52 x 10^{-6} and 2.15 x 10^{-7} respectively. The second optimal value that shows effective performance are 8.46x10^{-6} on parameter n=40, p=0.15. The performance in terms of SD and SEM value are 1.34 x 10^{-6} and 1.90 x 10^{-7}. The third performance are on parameter n=35, p=0.20 with optimal value 8.64 x10^{-6}, SD value of 1.16 x 10^{-6} and SEM, 1.63 x 10^{-7}. The highest optimal value where the value not close to zero are on parameter n=10, p=0.05 with 9.93 x 10^{-6}. The performance in terms of SD and SEM value 1.11 x 10^{-7} and 1.58 x 10^{-8} respectively.

Table-4. Result of CS algorithm on Rosenbrock function.

	Mean	SD	SEM
n=10,p=0.05	9.93 x 10^{-6}	1.11 x 10^{-7}	1.58 x 10^{-8}
n=10,p=0.10	9.89 x 10^{-6}	1.33 x 10^{-7}	1.88 x 10^{-8}
n=10,p=0.15	9.76 x 10^{-6}	3.48 x 10^{-7}	4.92 x 10^{-8}
n=10,p=0.20	9.69 x 10^{-6}	4.01 x 10^{-7}	5.67 x 10^{-8}
n=10,p=0.25	9.75 x 10^{-6}	2.79 x 10^{-7}	3.95 x 10^{-8}
n=15,p=0.05	9.64 x 10^{-6}	4.11 x 10^{-7}	5.81 x 10^{-8}
n=15,p=0.10	9.58 x 10^{-6}	5.20 x 10^{-7}	7.35 x 10^{-8}
n=15,p=0.15	9.38 x 10^{-6}	6.12 x 10^{-7}	8.66 x 10^{-8}
n=15,p=0.20	9.34 x 10^{-6}	6.84 x 10^{-7}	6.84 x 10^{-7}
n=15,p=0.25	9.05 x 10^{-6}	9.91 x 10^{-7}	1.40 x 10^{-7}
n=20,p=0.05	9.39 x 10^{-6}	5.86 x 10^{-7}	8.28 x 10^{-8}
n=20,p=0.10	8.90 x 10^{-6}	9.18 x 10^{-7}	1.30 x 10^{-7}
n=20,p=0.15	9.20 x 10^{-6}	7.16 x 10^{-7}	1.01 x 10^{-7}
n=20,p=0.20	9.05 x 10^{-6}	7.40 x 10^{-7}	1.05 x 10^{-7}
n=20,p=0.25	9.08 x 10^{-6}	9.45 x 10^{-7}	1.34 x 10^{-7}
n=25,p=0.05	8.98 x 10^{-6}	1.00 x 10^{-6}	1.42 x 10^{-7}
n=25,p=0.10	8.97 x 10^{-6}	9.13 x 10^{-7}	1.29 x 10^{-7}
n=25,p=0.15	8.80 x 10^{-6}	1.09 x 10^{-6}	1.54 x 10^{-7}
n=25,p=0.20	8.44 x 10^{-6}	1.52 x 10^{-6}	2.15 x 10^{-7}
n=25,p=0.25	8.89 x 10^{-6}	1.23 x 10^{-6}	1.74 x 10^{-7}
n=30,p=0.05	8.84 x 10^{-6}	1.24 x 10^{-6}	1.75 x 10^{-7}
n=30,p=0.10	8.81 x 10^{-6}	1.13 x 10^{-6}	1.60 x 10^{-7}
n=30,p=0.15	9.00 x 10^{-6}	8.92 x 10^{-7}	1.26 x 10^{-7}
n=30,p=0.20	8.65 x 10^{-6}	1.07 x 10^{-6}	1.51 x 10^{-7}
n=30,p=0.25	8.95 x 10^{-6}	1.09 x 10^{-6}	1.55 x 10^{-7}
n=35,p=0.05	8.72 x 10^{-6}	1.13 x 10^{-6}	1.60 x 10^{-7}
n=35,p=0.10	8.67 x 10^{-6}	1.11 x 10^{-6}	1.57 x 10^{-7}
n=35,p=0.15	8.95 x 10^{-6}	9.22 x 10^{-7}	1.30 x 10^{-7}
n=35,p=0.20	8.64 x 10^{-6}	1.16 x 10^{-6}	1.63 x 10^{-7}
n=35,p=0.25	8.81 x 10^{-6}	1.31 x 10^{-6}	1.85 x 10^{-7}
n=40,p=0.05	8.91 x 10^{-6}	8.95 x 10^{-7}	1.27 x 10^{-7}
n=40,p=0.10	8.77 x 10^{-6}	1.12 x 10^{-6}	1.58 x 10^{-7}
n=40,p=0.15	8.46 x 10^{-6}	1.34 x 10^{-6}	1.90 x 10^{-7}
n=40,p=0.20	8.95 x 10^{-6}	1.08 x 10^{-6}	1.52 x 10^{-7}
n=40,p=0.25	8.80 x 10^{-6}	1.11 x 10^{-6}	1.57 x 10^{-7}

Based on Figure-4, the best performance of CS algorithm on Rosenbrock function that have the lowest

optimal value that nearest to zero are on parameters n=25, p=0.20 with 8.44 x10^{-6}. The second optimal value that shows effective performance are 8.46x10^{-6} on parameter n=40, p=0.15. The third performance are on parameter n=35, p=0.20 with optimal value 8.64 x10^{-6}.The highest optimal value where the value not close to zero are on parameter n=10, p=0.05 with 9.93 x 10^{-6}.

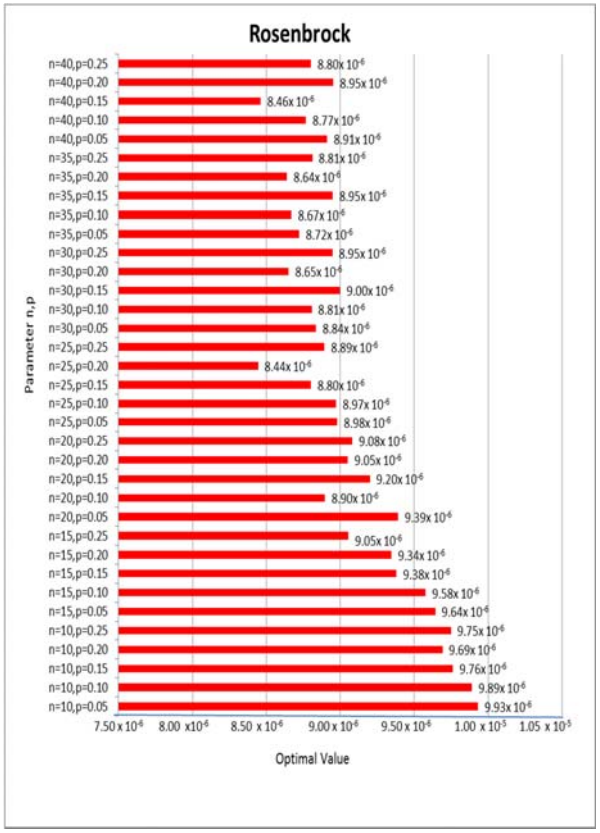

Figure-4. Performance of CS algorithm on Rosenbrock function.

Schwefel function

The Schwefel function is complex, with many local minima. The function is multimodal. Table-5 indicates the result of CS algorithm on Schwefel function.

Based on Table-5, the best performance of CS algorithm on Schwefel function that have the lowest optimal value that nearest to zero are on parameters n=35, p=0.10 with 8.38 x10^{-6}. The performance in terms of SD and SEM value are 1.30 x10^{-6} and 1.83x10^{-7} respectively. The second performance that have the lowest optimal value are on parameters n=30, p=0.25 and n=35, p=0.15 that have the same value 8.50 x 10^{-6}. The performances in terms of SD are 1.46 x 10^{-6} and 1.47 x 10^{-6} respectively. While, it have same value of SEM 2.07 x 10^{-7}. The third performance are on parameter n=35, p=0.25 with optimal value 8.55 x10^{-6}, SD value of 1.32 x 10^{-6} and SEM, 1.86 x 10^{-7}. The highest optimal value where the value not close to zero are on parameter n=10, p=0.05 with 9.89 x10^{-6}. The performance in terms of SD and SEM value 1.23 x 10^{-7} and 1.74 x 10^{-8} respectively.

Table-5. Result of CS algorithm on Schwefel function.

	Mean	SD	SEM
n=10,p=0.05	9.89 x 10^{-6}	1.23 x 10^{-7}	1.74 x 10^{-8}
n=10,p=0.10	9.85 x 10^{-6}	1.53 x 10^{-7}	2.16 x 10^{-8}
n=10,p=0.15	9.82 x 10^{-6}	1.67 x 10^{-7}	2.37 x 10^{-8}
n=10,p=0.20	9.77 x 10^{-6}	2.36 x 10^{-7}	3.34 x 10^{-8}
n=10,p=0.25	9.69 x 10^{-6}	3.66 x 10^{-7}	5.17 x 10^{-8}
n=15,p=0.05	9.63 x 10^{-6}	3.86 x 10^{-7}	5.45 x 10^{-8}
n=15,p=0.10	9.50 x 10^{-6}	5.59 x 10^{-7}	7.91 x 10^{-8}
n=15,p=0.15	9.19 x 10^{-6}	8.26 x 10^{-7}	1.17 x 10^{-7}
n=15,p=0.20	9.30 x 10^{-6}	8.41 x 10^{-7}	1.19 x 10^{-7}
n=15,p=0.25	9.18 x 10^{-6}	6.91 x 10^{-7}	9.78 x 10^{-8}
n=20,p=0.05	9.44 x 10^{-6}	5.55 x 10^{-7}	7.84 x 10^{-8}
n=20,p=0.10	9.01 x 10^{-6}	1.02 x 10^{-6}	1.44 x 10^{-7}
n=20,p=0.15	8.93 x 10^{-6}	8.25 x 10^{-7}	1.17 x 10^{-7}
n=20,p=0.20	9.05 x 10^{-6}	8.40 x 10^{-7}	1.19 x 10^{-7}
n=20,p=0.25	8.84 x 10^{-6}	1.03 x 10^{-6}	1.46 x 10^{-7}
n=25,p=0.05	9.19 x 10^{-6}	7.31 x 10^{-7}	1.03 x 10^{-7}
n=25,p=0.10	8.91 x 10^{-6}	1.04 x 10^{-6}	1.47 x 10^{-7}
n=25,p=0.15	8.83 x 10^{-6}	1.02 x 10^{-6}	1.45 x 10^{-7}
n=25,p=0.20	9.12 x 10^{-6}	7.11 x 10^{-7}	1.00 x 10^{-7}
n=25,p=0.25	8.97 x 10^{-6}	8.92 x 10^{-7}	1.26 x 10^{-7}
n=30,p=0.05	9.02 x 10^{-6}	1.00 x 10^{-6}	1.42 x 10^{-7}
n=30,p=0.10	8.90 x 10^{-6}	9.76 x 10^{-7}	1.38 x 10^{-7}
n=30,p=0.15	8.89 x 10^{-6}	1.05 x 10^{-6}	1.49 x 10^{-7}
n=30,p=0.20	8.95 x 10^{-6}	8.69 x 10^{-7}	1.23 x 10^{-7}
n=30,p=0.25	8.50 x 10^{-6}	1.46 x 10^{-6}	2.07 x 10^{-7}
n=35,p=0.05	8.82 x 10^{-6}	1.00 x 10^{-6}	1.42 x 10^{-7}
n=35,p=0.10	8.38 x 10^{-6}	1.30 x 10^{-6}	1.83 x 10^{-7}
n=35,p=0.15	8.50 x 10^{-6}	1.47 x 10^{-6}	2.07 x 10^{-7}
n=35,p=0.20	8.83 x 10^{-6}	9.25 x 10^{-7}	1.31 x 10^{-7}
n=35,p=0.25	8.55 x 10^{-6}	1.32 x 10^{-6}	1.86 x 10^{-7}
n=40,p=0.05	8.60 x 10^{-6}	1.15 x 10^{-6}	1.62 x 10^{-7}
n=40,p=0.10	8.79 x 10^{-6}	1.08 x 10^{-6}	1.53 x 10^{-7}
n=40,p=0.15	8.66 x 10^{-6}	1.21 x 10^{-6}	1.71 x 10^{-7}
n=40,p=0.20	9.01 x 10^{-6}	1.08 x 10^{-6}	1.53 x 10^{-7}
n=40,p=0.25	9.01 x 10^{-6}	1.08 x 10^{-6}	1.53 x 10^{-7}

Based on Table-5, the best performance of CS algorithm on Schwefel function that have the lowest optimal value that nearest to zero are on parameters n=35,p=0.10 with 8.38 x10^{-6}. The second performance that have the lowest optimal value are on parameters n=30, p=0.25 and n=35, p=0.15 that have the same value 8.50

$x10^{-6}$. The third performance are on parameter n=35, p=0.25 with optimal value 8.55 $x10^{-6}$.The highest optimal value where the value was close to zero are on parameter n=10, p=0.05 with 9.89 $x10^{-6}$.

RESULTS COMPARISON

The best optimal values of Cuckoo search algorithm with Levy flight are compared with four algorithms. Every algorithm has different characteristics that give different result on benchmark function. Table-6 indicates the average optimization of four benchmark testing functions.

Table-6. Average optimization results for benchmark testing functions.

Function(s)		PSO	WSA	ABC	CS
Ackley	Mean	0.16	5.13	0.21	9.16×10^{-6}
	SD	0.49	0.74	0.27	8.17×10^{-7}
	SEM	0.09	0.13	0.05	1.16×10^{-7}
Griewank	Mean	0.02	0.12	0.33	8.68×10^{-6}
	SD	0.02	0.01	0.23	1.13×10^{-6}
	SEM	0.004	0	0.04	1.60×10^{-7}
Rastrigin	Mean	43.97	114.6	0.1	8.51×10^{-6}
	SD	11.73	15.27	0.30	1.44×10^{-6}
	SEM	2.14	2.79	0.05	2.04×10^{-7}
Rosenbrock	Mean	15.08	5.04	9.32	8.44×10^{-6}
	SD	24.17	1.72	10.9	1.52×10^{-6}
	SEM	4.41	0.31	1.99	2.15×10^{-7}
Schwefel	Mean	0.03	0.01	0	8.38×10^{-6}
	SD	0.06	0	0	1.30×10^{-6}
	SEM	0.01	0	0	1.83×10^{-7}

Table-6 shows CS algorithm perform better on five benchmark functions, Ackley (9.16×10^{-6}), Griewank (8.68×10^{-6}), Rastrigin (8.51×10^{-6}), Rosenbrock (8.44×10^{-6}) and Schwefel (8.38×10^{-6}) that have the optimal value (average) that nearest to zero than other algorithms. The second best performance was given by ABC which performed well during convergence on four benchmark functions, Ackley (0.21), Griewank (0.33), Rastrigin (0.1) and Schwefel (0). While, PSO shows better performance on Ackley (0.16), Griewank (0.02) and Rastrigin (43.97). Lastly, WSA effectively performs on the optimal value that is almost zero on two benchmark functions, Griewank (0.12) and Schwefel (0.01). Overall, it can be seen that during simulationsall the algorithms used in this researchperformed very well onSchwefel function.

CONCLUSIONS

This paper proposed the use of population estimation in a new meta-heuristic called Cuckoo search (CS) algorithm to minimize the training error, achieve fast convergence rate and to avoid local minimum problem. The CS algorithm has been applied independently to solve several engineering design optimization problems based on cuckoo bird's behavior. The algorithm are tested on five benchmark functions such as Ackley function, Griewank function, Rastrigin function, Rosenbrock function and Schwefel function. The performance of the proposed algorithm was compare with Particle Swarm Optimization (PSO), Wolf Search Algorithm (WSA) and Artificial Bee Colony (ABC). The performance evaluation has been carried out based on their average convergences that have the optimal value near to zero. The simulation results show that the proposed algorithm, CS with Levy flight performed very well and more efficient in finding the global optima. Furthermore, the results clearly showed that the proposed algorithm has significantly improved the convergence rate and avoid local minimum problem of benchmark function compared with PSO, WSA, and ABC.

ACKNOWLEDGEMENT

The authors would like to thank ORICCM, Universiti Tun Hussein Onn Malaysia (UTHM) for supporting this Research under Fundamental Research Grant Scheme (FRGS) vote. 1236.

REFERENCES

Karaboga, D., and Akay, B. 2009. Artificial bee colony (ABC), harmony search and bees algorithms on numerical optimization. In Proceedings of Innovative Production Machines and Systems Virtual Conference, IPROMS (pp. 1-6).

Karaboga, D. and Basturk, B. 2007. Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization. In LNAI 4529 (pp. 789-798). doi:10.1007/978-3-540-72950-1_77.

Karaboga, D. and Basturk, B. 2008. On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing Journal, 8, 687-697. doi:10.1016/j.asoc.2007.05.007.

Kennedy, J. and Eberhart, R. 1995. Particle swarm optimization. In Neural Networks, 1995. Proceedings., IEEE International Conference on (Vol. 4, pp. 1942-1948 vol.4). doi:10.1109/ICNN.1995.488968.

Kulkarni, R. V. and Venayagamoorthy, G. K. 2011. Particle swarm optimization in wireless-sensor networks: A brief survey. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 41(2), 262-267. doi:10.1109/TSMCC.2010.2054080.

Millonas, M. M. 1994. Swarms, phase transitions, and collective intelligence. In Artificial Life III. MA: Addison Wesley.

Tang, R., Fong, S., Yang, X.-S. and Deb, S. 2012. Wolf search algorithm with ephemeral memory. Seventh International Conference on Digital Information Management (ICDIM 2012), 165-172. doi:10.1109/ICDIM.2012.6360147.

Yang, X. S. 2013. Bat algorithm and cuckoo search: A tutorial. Studies in Computational Intelligence. doi:10.1007/978-3-642-29694-9-17.

Yang, X. S. and Deb, S. 2009. Cuckoo search via Lévy flights. In 2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings (pp. 210–214). doi:10.1109/NABIC.2009.5393690.

Yang, X.-S. 2010. Nature-Inspired Metaheuristic Algorithms (Second Edi). Luniver Press, Cambridge, UK.