

Research Article

An Improved Cockroach Swarm Optimization

I. C. Obagbuwa and A. O. Adewumi

School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville, Durban 4000, South Africa

Correspondence should be addressed to I. C. Obagbuwa; ibidunobagbuwa@yahoo.com

Received 6 December 2013; Accepted 18 March 2014; Published 14 May 2014

Academic Editors: P. Agarwal, S. Balochian, V. Bhatnagar, and Y. Zhang

Copyright © 2014 I. C. Obagbuwa and A. O. Adewumi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hunger component is introduced to the existing cockroach swarm optimization (CSO) algorithm to improve its searching ability and population diversity. The original CSO was modelled with three components: chase-swarming, dispersion, and ruthless; additional hunger component which is modelled using partial differential equation (PDE) method is included in this paper. An improved cockroach swarm optimization (ICSO) is proposed in this paper. The performance of the proposed algorithm is tested on well known benchmarks and compared with the existing CSO, modified cockroach swarm optimization (MCSO), roach infestation optimization RIO, and hungry roach infestation optimization (HRIO). The comparison results show clearly that the proposed algorithm outperforms the existing algorithms.

1. Introduction

Swarm intelligence (SI) is a method of computing whereby simple decentralized agents get information by interacting locally with one another and their environment [1]. The local information received is not controlled centrally; local interaction of agents results in amazing and emergent global patterns which can be adopted for solving problems [1].

SI algorithms draw inspiration from insects and animals social behaviour and have been proven in literature to be efficient in solving global optimization problems. Examples of existing SI algorithms include particle swarm optimization (PSO), ant colony optimization (ACO), and bee colony optimization (BCO). PSO based on bird social behaviour, introduced by Kennedy and Eberhart [2], has been applied to several problems, including power and management processes [3, 4] and combinatorial optimization problem in [5]. ACO based on ant social behaviour, introduced by Dorigo [6], has been applied to problems such as vehicle routing problem [7] and network routing problem [8]. BCO based on bees social behaviour, introduced by Pham et al. [9], has been applied to real world problems by Karaboga and his research group [10–12].

One of the recent developments in SI is cockroach optimization [13–16]. Cockroach belongs to Insecta Blattodea,

abodes in warm, dark, and moist shelters, and exhibits habits which include chasing, swarming, dispersing, being ruthless and omnivorous, and food searching. Cockroaches interact with peers and respond to their immediate environment and make decisions based on their interaction such as selecting shelter, searching for food sources and friends, dispersing when danger is noticed, and eating one another when food is scarce.

The original cockroach swarm optimization (CSO) algorithm, introduced by ZhaoHui and Haiyan [14], was modified by ZhaoHui with the introduction of inertial weight [15]. CSO algorithms [14, 15] mimic chase swarming, dispersion, and ruthless social behaviour of cockroaches.

Global optimization problems are considered as very hard problems, ever increasing in complexity. It became necessary to design better optimization algorithms; this necessitated the design of a better cockroach algorithm. This paper extends MCSO with the introduction of another social behaviour called hunger behaviour. Hunger behaviour prevents local optimum and enhances diversity of population. An improved cockroach swarm optimization (ICSO) is presented in this paper.

The organization of this paper is as follows: Section 2 presents CSO, MCSO, and ICSO models with algorithmic

steps; Section 3 shows the experiments carried out and results obtained; the paper is summarised in Section 4.

2. Cockroach Swarm Optimization

CSO algorithm is a population based global optimization algorithm which has been applied to problems in literature including [17–19]. CSO [14] models are given as follows.

(1) Chase-Swarming Behaviour.

$$x_i = \begin{cases} x_i + \text{step} \cdot \text{rand} \cdot (p_i - x_i), & x_i \neq p_i \\ x_i + \text{step} \cdot \text{rand} \cdot (p_g - x_i), & x_i = p_i, \end{cases} \quad (1)$$

where x_i is the cockroach position, step is a fixed value, rand is a random number within $[0, 1]$, p_i is the personal best position, and p_g is the global best position. Consider

$$p_i = \text{Opt}_j \{x_j, |x_i - x_j| \leq \text{visual}\}, \quad (2)$$

where perception distance visual is a constant, $j = 1, 2, \dots, N$, $i = 1, 2, \dots, N$. Consider

$$p_g = \text{Opt}_i \{x_i\}. \quad (3)$$

(2) Dispersion Behaviour.

$$x_i = x_i + \text{rand}(1, D), \quad i = 1, 2, \dots, N, \quad (4)$$

where $\text{rand}(1, D)$ is a D -dimensional random vector that can be set within a certain range.

(3) Ruthless Behaviour.

$$x_k = p_g, \quad (5)$$

where k is a random integer within $[1, N]$ and p_g is the global best position.

2.1. Modified Cockroach Swarm Optimization. ZhaoHui presented a modified cockroach swarm optimization (MCSO) [15] with the introduction of inertial weight to chase swarming component of original CSO as shown below. Other models remain as in original CSO.

Chase-swarming behaviour is as follows:

$$x_i = \begin{cases} w \cdot x_i + \text{step} \cdot \text{rand} \cdot (p_i - x_i), & x_i \neq p_i \\ w \cdot x_i + \text{step} \cdot \text{rand} \cdot (p_g - x_i), & x_i = p_i, \end{cases} \quad (6)$$

where w is an inertial weight which is a constant.

2.2. Improved Cockroach Swarm Optimization. In this paper, MCSO is extended with additional component called hunger behaviour.

2.2.1. Hunger Behaviour. At interval of time, when cockroach is hungry, it migrates from its comfortable shelter and friends company to look for food [13, 20]. Hunger behaviour is

modelled using partial differential equation (PDE) migration techniques [21]. Cockroach migrates from its shelter to any available food source x_{food} within the search space. A threshold hunger is defined, when cockroach is hungry and threshold hunger is reached; it migrates to food source. Hunger behaviour prevents local optimum and enhances diversity of population.

PDE migration equation is described by Kerckhove [21]:

$$\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x} \quad (7)$$

with $u(0, x) = u_0(x)$.

Parameter c is the controlling speed of the migration. u is the population size, t is time, and x is location or position. $u(t, x)$ is the population size at time t in location x with $u(0, x) = u_0(x)$ being the initial population distribution. Consider

$$\begin{aligned} \frac{\partial u}{\partial t} &= -c \frac{\partial u}{\partial x}, \\ \frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} &= 0. \end{aligned} \quad (8)$$

The characteristic equations are

$$\frac{dt}{1} = \frac{dx}{c} = \frac{du}{0}, \quad (9)$$

$$dx - cdt = 0.$$

By integration, we have

$$\begin{aligned} x - ct &= \alpha, \\ u &= u(\alpha), \\ u &= u(x - ct), \\ u[t, x] &= u_0[-ct + x]. \end{aligned} \quad (10)$$

Consider displacement = speed \times time.

In $u_0(x - ct)$, $u_0(x)$ displaces ct .

$u_0(x - ct)$ satisfies migration equation at any initial population distribution $u_0(x)$ [21].

Hunger behaviour is modelled as follows:

If (hunger == t_{hunger})

$$x_i = x_i + (x_i - ct) + x_{\text{food}}, \quad (11)$$

where x_i denotes cockroach position, $(x_i - ct)$ denotes cockroach migration from its present position, c is a constant which controls migration speed at time t , x_{food} denotes food location, t_{hunger} denotes hunger threshold, and hunger is a random number $[0, 1]$.

2.2.2. Improved Cockroach Swarm Optimization Models

(1) Chase-Swarming Behaviour.

$$x_i = \begin{cases} w \cdot x_i + \text{step} \cdot \text{rand} \cdot (p_i - x_i), & x_i \neq p_i \\ w \cdot x_i + \text{step} \cdot \text{rand} \cdot (p_g - x_i), & x_i = p_i, \end{cases} \quad (12)$$

```

INPUT: Fitness function:  $f(x), x \in R^D$ 
set parameters and generate an initial population of cockroach
set  $p_g = x_1$ 
for  $i = 2$  to  $N$  do
    if  $f(x_i) < f(p_g)$  then
         $p_g = x_i$ 
    end if
end for
for  $t = 1$  to  $T_{\max}$  do
    for  $i = 1$  to  $N$  do
        for  $j = 1$  to  $N$  do
            if  $\text{abs}(x_i - x_j) < \text{visual}; f(x_j) < f(x_i)$  then
                 $p_i = x_j$ 
            end if
        end for
        if  $p_i == x_i$  then
             $x_i = w \cdot x_i + \text{step} \cdot \text{rand} \cdot (p_g - x_i)$ 
        else
             $x_i = w \cdot x_i + \text{step} \cdot \text{rand} \cdot (p_i - x_i)$ 
        end if
        if  $f(x_i) < f(p_g)$  then
             $p_g = x_i$ 
        end if
    end for
    if Hunger ==  $t_{\text{hunger}}$  then
         $x_i = x_i + (x_i - ct) + x_{\text{fd}}$ 
        hunger $i$  = 0
        Increment hunger $i$  counters
    end if
    for  $i = 1$  to  $N$  do
         $x_i = x_i + \text{rand}(1, D)$ 
        if  $f(x_i) < f(p_g)$  then
             $p_g = x_i$ 
        end if
    end for
     $k = \text{randint}([1, N])$ 
     $x_k = p_g;$ 
end for
Check termination condition

```

ALGORITHM 1: An improved cockroach swarm optimization algorithm.

where w is an inertial weight which is a constant, step is a fixed value, rand is a random number within $[0, 1]$, p_i is the personal best position, and p_g is the global best position. Consider

$$p_i = \text{Opt}_j \{x_j, |x_i - x_j| \leq \text{visual}\}, \quad (13)$$

where perception distance visual is a constant, $j = 1, 2, \dots, N$, $i = 1, 2, \dots, N$. Consider

$$p_g = \text{Opt}_i \{x_i\}. \quad (14)$$

(2) *Hunger Behaviour*. If hunger == t_{hunger} ,

$$x_i = x_i + (x_i - ct) + x_{\text{food}}, \quad (15)$$

where x_i denotes cockroach position, $(x_i - ct)$ denotes cockroach migration from its present position, c is a constant

which controls migration speed at time t , x_{food} denotes food location, t_{hunger} denotes hunger threshold, and hunger is a random number within $[0, 1]$.

(3) Dispersion Behaviour.

$$x_i = x_i + \text{rand}(1, D), \quad i = 1, 2, \dots, N, \quad (16)$$

where $\text{rand}(1, D)$ is a D -dimensional random vector that can be set within a certain range.

(4) Ruthless Behaviour.

$$x_k = p_g, \quad (17)$$

where k is a random integer within $[1, N]$ and p_g is the global best position.

The algorithm for ICSO is illustrated in Algorithm 1 and its computational steps given as follows.

- (1) Initialise cockroach swarm with uniform distributed random numbers and set all parameters with values.
- (2) Find p_i and p_g using (12) and (13).
- (3) Perform chase-swarming using (11).
- (4) Perform hunger behaviour using (14)
- (5) Perform dispersion behaviour using (15).
- (6) Perform ruthless behaviour using (16).
- (7) Repeat the loop until stopping criterion is reached.

Series of experiments are conducted in Section 3 using established global optimization problems to test ICSO performance. The performance of ICSO is compared with that of existing algorithms RIO, HRIO, CSO, and MCSO.

3. Simulation Studies

The speed, accuracy, robustness, stability, and searching capabilities of ICSO are evaluated in this section with 23 benchmark test functions. The test functions were adopted from [22–24]; any further information about the test functions can be found in these references. The test functions are of different characteristics such as unimodal (U), multimodal (M), separable (S), and nonseparable (N). Table 1 of this paper shows the test functions used, whose problem ranges from 2 to 30 in dimension as in [22–24].

All algorithms were implemented in MATLAB 7.14 (R2012a) and run on a computer with 2.30 GHz processor with 4.00 GB of RAM. Experimental setting of [13–15] is used for the experiments of this paper; experiment runs 20 times with maximum iteration 1000, perception distance visual = 5, the largest step was step = 2, and inertia weight was $w = 0.618$; we defined hunger threshold $t_{\text{hunger}} = 0.5$ and hunger as a randomly generated number $[0, 1]$ in each iteration for ICSO. Cockroach parameters [13] are used for RIO and HRIO; $c_0 = 0.7$ and $c_{\max} = 1.43$, hunger threshold $t_{\text{hunger}} = 100$, and hunger as randomly generated number $[0, (t_{\text{hunger}} - 1)]$. Cockroach population size $N = 50$ is used in this paper for all the algorithms. Further details about RIO, HRIO, CSO, and MCSO can be found in [13–15].

ICSO along with similar algorithms, that is, CSO, MCSO, RIO, and HRIO, was implemented with several simulation experiments conducted and reported. Success rate, average and best fitness, standard deviation (STD), and execution time in seconds are used as performance measure for comparative purpose (see Tables 2, 3, and 4 of this paper).

ICSO locates minimum values for the tested benchmark problems such as Bohachevsky, Rastrigin, Easom, Schaffer, Step, and Storn's Tchebychev problems as shown in Tables 2, 3, and 4. The comparison of the average performance of ICSO with that of RIO, HRIO, CSO, and MCSO is shown in Table 5; the comparison result clearly shows that ICSO outperforms other algorithms. Similarly, the best performance of ICSO with that of RIO, HRIO, CSO, and MCSO is shown in Table 6; ICSO has better performance than others.

ICSO algorithm has consistent performance in each iteration. This is proved by very low standard deviation of the average optimal recorded during experiments. The

ICSO average optimal STD is compared with the STD of RIO, HRIO, CSO, and MCSO in Table 7. ICSO has better minimum STD than others.

ICSO locates good solutions in each experiment; this is proved by the success rate of the algorithm. Table 8 shows the comparison of the success rate of the proposed algorithm with the existing algorithms RIO, HRIO, CSO, and MCSO. ICSO has 100% success rate in all test functions except Rosenbrock.

ICSO utilizes minimum time in executing the selected test function. Table 9 shows the comparison of the execution time of ICSO and that of RIO, HRIO, CSO, and MCSO; ICSO is shown to have utilized minimum time.

To determine the significant difference between the performance of the proposed algorithm and the existing algorithms, test statistic of Jonckheere-Terpstra (J-T) test was conducted using the statistical package for the social science (SPSS). The Null hypothesis test for J-T test is that there is no difference among several independent groups. As the usual practice in most literature, P value threshold value for hypothesis test was set to 0.05. If P value is less than 0.05, the Null is rejected which means there is significant difference between the groups. Otherwise the Null hypothesis is accepted. Table 10 shows the result of J-T test; P value (Asymp. Sig.) was computed to be 0.001. The P value is less than the threshold value 0.05; therefore, there is significant difference in performance of ICSO and that of RIO, HRIO, CSO, and MCSO for benchmarks evaluated.

Effect size of the significant difference is the measure of the magnitude of the observed effect. The effect size r , ($1 > r < 0$) of the significant difference of J-T test, was calculated as

$$r = \frac{Z}{\sqrt{N}}, \quad (18)$$

where Z is the standard data of J-T statistic as shown in Table 10, N is the total number of samples, and $N = 114$. Consider

$$Z = \frac{x - \mu}{\sigma}, \quad (19)$$

where x denotes observed J-T statistic, μ denotes the mean J-T statistic, and σ denoted the standard deviation of J-T statistic. Consider

$$\begin{aligned} Z &= \frac{1952 - 2599}{199.355} = -3.245, \\ r &= \frac{-3.245}{\sqrt{114}} = -0.3. \end{aligned} \quad (20)$$

The distance between the observed data and the mean in units of standard deviation is absolute value of $|Z|$ (Z is negative when observed data is below the mean and positive when above). The effect size 0.3 is of medium size, using Cohen's guideline on effect size [25, 26]. The statistics of 0.3 effect size shows that there is significant difference of medium magnitude between proposed algorithm and existing algorithms.

TABLE 1: Benchmark test functions.

Number	Range	D	C	Functions	Description
1	[-100, 100]	30	US	Step	$f(x) = \sum_{i=1}^n ([x_i + 0.5])^2$
2	[-100, 100]	30	US	Sphere	$f(x) = \sum_{i=1}^n x_i^2$
3	[-10, 10]	30	US	Sumsquares	$f(x) = \sum_{i=1}^n ix_i^2$
4	[-100, 100]	2	MS	Bohachevsky1	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$
5	[-100, 100]	2	MN	Bohachevsky2	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1)(4\pi x_2) + 0.3$
6	[-100, 100]	2	MN	Bohachevsky3	$f(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$
7	[0, 180]	20	UN	Sinusoidal20	$f(x) = - \left[A \prod_{i=1}^n \sin(x_i - z) + \prod_{i=1}^n \sin(B(x_i - z)) \right]$
8	[-100, 100]	30	UN	Quadratic	$A = 2.5, B = 5, z = 30$
9	[-100, 100]	2	UN	Easom	$f(x) = -\cos x_1 \cos x_2 \cdot \exp(-((x_1 - \pi)^2 + (x_2 - \pi)^2))$
10	[-10, 10]	2	UN	Matyas	$f(x) = 0.26(x_1 + x_2) - 0.48x_1x_2$
11	[-5, 10]	10	UN	Zakharov	$f(x) = \sum_{i=1}^n (x_i)^2 + \left(\sum_{i=1}^n 0.5ix_i \right)^2 + \left(\sum_{i=1}^n 0.5ix_i \right)^4$
12	[-10, 10]	24	UN	Powell	$f(x) = \sum_{i=1}^{n/2} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$
13	[-10, 10]	30	UN	Schwefel2.22	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
14	[-30, 30]	30	UN	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
15	[-5.12, 5.12]	30	MS	Rastrigin	$f(x) = \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i) + 10$

TABLE I: Continued.

Number	Range	D	C	Functions	Description
16	$[-100, 100]$	2	MN	Schaffer1	$f(x) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$
17	$[-100, 100]$	30	MN	Schaffer2	$f(x) = (x_1^2 + x_2^2)^{0.25} (\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1)$
18	$[-600, 600]$	30	MN	Griewangk	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
19	$[-32, 32]$	30	MN	Ackley	$f(x) = -20 \exp(-0.2\sqrt{\sum_{i=1}^n (x_i^2/n)}) - \exp(\sum_{i=1}^n \cos(2\pi x_i/n)) + 20 + e$
20	$[-5, 5]$	2	MN	Three hump camel back	$f(x) = 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 + x_1x_2 + x_2^2$
21	$[-5, 5]$	2	MN	Six hump camel back	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
22	$[-128, 128]^n$	9	UN	Storn's Tchebychev	$f(x) = p_1 + p_2 + p_3,$ $p_1 = \begin{cases} (u-d)^2 & \text{if } u < d \\ 0 & \text{if } u \geq d \end{cases} \quad u = \sum_{i=1}^n (1.2)^{n-i} x_i$ $p_2 = \begin{cases} (v-d)^2 & \text{if } v < d \\ 0 & \text{if } v \geq d \end{cases} \quad v = \sum_{i=1}^n (-1.2)^{n-i} x_i$ $p_3 = \sum_{j=0}^m \begin{cases} (w_j - 1)^2 & \text{if } w_j > 1 \\ 0 & \text{if } w_j < -1 \\ 0 & \text{if } -1 \leq w_j \leq 1 \end{cases} \quad w_j = \sum_{i=1}^n \left(\frac{2j}{m} - 1\right)^{n-i} x_i,$
23	$[-32768, 32768]^n$	17		Storn's Tchebychev	for $n = 9: d = 72.661$, and $m = 60$ for $n = 17: d = 10558.145$, and $m = 100$.

D : dimension; C: characteristic; U: unimodal; S: separable; N: non-separable.

TABLE 2: Simulation results of RIO, HRIO, CSO, MCSO, and ICSO.

SN	Fn.	Dim.	Opt.	RIO	HRIO	CSO	MCSO	ICSO
1	Bohal1	2	0	Ave.	$3.4405E - 05$	$3.2877E - 04$	$2.9893E02$	$3.5153E - 09$
				STD	$2.5963E - 05$	$3.0334E - 04$	$5.0332E02$	$1.4392E - 08$
				Best	$1.3520E - 07$	$5.2651E - 06$	$2.0651E - 05$	0.0000
				Success	20/20	20/20	5/20	20/20
2	Boha2	2	0	Time	1.137525	0.886356	23.913237	0.075212
				Ave.	$4.2829E - 05$	$4.6703E - 04$	$9.0941E02$	$8.4459E - 12$
				STD	$3.0070E - 05$	$3.4047E - 04$	$1.7794E03$	$2.9240E - 11$
				Best	$2.2910E - 06$	$9.374E - 06$	$1.3775E - 05$	0.0000
3	Boha3	2	0	Success	20/20	20/20	4/20	20/20
				Time	0.998178	0.946887	26.492095	0.072021
				Ave.	$5.3479E - 05$	$4.7575E - 04$	$7.4284E02$	$2.1388E - 14$
				STD	$2.9141E - 05$	$2.3273E - 04$	$1.6739E03$	$4.8670E - 14$
4	3camel	2	0	Best	$3.1200E - 06$	$4.6981E - 05$	$2.3093E - 07$	0.0000
				Success	20/20	20/20	3/20	20/20
				Time	1.089920	0.885252	25.028054	0.080908
				Ave.	$1.4962E - 02$	$4.3021E - 04$	$5.003E09$	$7.098E - 11$
5	6camel	2	-1.03163	STD	$6.6769E - 02$	$2.8371E - 04$	$1.7137E10$	$3.0201E - 10$
				Best	$1.1739E - 06$	$2.2449E - 05$	$1.7642E - 05$	$3.1395E - 19$
				Success	19/20	20/20	12/20	20/20
				Time	4.231533	0.794983	18.281683	0.104132
6	Easom	2	-1	Ave.	$-4.3522E - 01$	$-4.7652E - 01$	$1.5763E05$	$-1.0263E - 08$
				STD	$3.3322E - 01$	$3.1284E - 01$	$7.0503E05$	$4.4391E - 08$
				Best	-1.0215	-1.0034	$-9.4052E - 01$	$-1.9879E - 07$
				Success	20/20	20/20	19/20	20/20
7	Matyax	2	0	Time	0.406355	0.330198	5.723039	0.0945856
				Ave.	-1	-1	$-4.3165E - 01$	-1
				STD	$3.7518E - 02$	$2.1031E - 02$	$3.4470E - 01$	$1.4897E - 08$
				Best	-1	-1	-1	-1
8	Schaffer1	2	-1	Success	20/20	20/20	20/20	20/20
				Time	0.124022	0.107303	0.106738	0.077179
				Ave.	$4.9470E - 05$	$3.2297E - 04$	7.5712	$2.6876E - 13$
				STD	$3.0244E - 05$	$2.6018E - 04$	$1.1247E01$	$8.9347E - 13$
9	Schaffer2	2	0	Best	$6.2897E - 06$	$1.2684E - 05$	$8.8777E - 06$	$6.6695E - 21$
				Success	20/20	20/20	11/20	20/20
				Time	0.973322	0.711734	13.559576	0.88536
				Ave.	-1.9069	-1.6211	$-2.9174E - 01$	-1
				STD	$7.0381E - 01$	$5.9214E - 01$	$7.5142E - 01$	$5.9575E - 07$
				Best	-2.7458	-2.7164	-2.7438	-1
				Success	20/20	20/20	20/20	20/20
				Time	0.109048	0.086433	0.119076	0.072400
				Ave.	$2.0179E - 03$	$1.6566E - 03$	7.1618	$3.3168E - 04$
				STD	$2.6407E - 03$	$1.4451E - 03$	5.3095	$3.0328E - 04$
				Best	$6.2423E - 05$	$4.1422E - 04$	$2.8354E - 01$	$1.5810E - 05$
				Success	2/20	13/20	0/20	20/20
				Time	62.567654	31.415836	29.194283	0.084127
				Ave.	$2.0179E - 03$	$1.6566E - 03$	7.1618	$3.3168E - 04$
				STD	$2.6407E - 03$	$1.4451E - 03$	5.3095	$3.0328E - 04$
				Best	$6.2423E - 05$	$4.1422E - 04$	$2.8354E - 01$	$1.5810E - 05$

Dim. denotes dimension. Opt. denotes optimum value. Bohal denotes Bohachevsky1. Boha2 denotes Bohachevsky2. Boha3 denotes Bohachevsky3. 3camel denotes three hump camel back. 6camel denotes six hump camel back.

TABLE 3: Simulation results of RIO, HRIO, CSO, MCSO, and ICSO.

SN	Fn.	Dim.	Opt.	RIO	HRIO	CSO	MCSO	ICSO
10	Sphere	30	0	Ave.	2.2168E - 05	1.6676E - 04	1.8123E02	1.5201E - 12
				STD	2.4528E - 05	2.4018E - 04	8.1048E02	6.7224E - 12
				Best	5.7627E - 09	5.5635E - 08	4.9195E - 07	2.9978E - 24
				Success	20/20	20/20	19/20	20/20
				Time	0.617544	0.557871	25.378161	0.82512
11	Rastrigin	30	0	Ave.	3.8135E - 05	3.2150E - 04	3.6022E03	9.1994E - 11
				STD	3.4436E - 05	3.0003E - 04	5.5728E03	3.9456E - 10
				Best	2.7098E - 07	2.1450E - 07	3.1340E - 04	0.0000
				Success	20/20	20/20	5/20	20/20
				Time	0.956329	0.826770	71.811170	0.175563
12	Rosenbrock	30	0	Ave.	2.5281E06	3.3571E06	9.5067E11	2.9000E01
				STD	4.0528E06	7.1150E06	2.2713E12	0.0000
				Best	1.6773E04	3.7562E04	4.4068E01	2.9000E01
				Success	0/20	0/20	0/20	0/20
				Time	126.618734	127.469638	81.361663	76.084929
13	Ackley	30	0	Ave.	2.0001E01	2.0005E01	1.9222E01	5.1593E - 06
				STD	3.0455E - 03	1.5671E - 02	5.8258	1.9149E - 05
				Best	2.0001E01	1.9998E01	2.0133E01	6.4623E - 09
				Success	0/20	0/20	0/20	20/20
				Time	122.216187	117.635854	82.227210	0.235012
14	Quadric	30	0	Ave.	2.4498E - 05	2.2711E - 04	3.4991E - 04	4.4754E - 13
				STD	2.7957E - 05	2.3635E - 04	3.3725E - 04	1.9751E - 12
				Best	1.1360E - 08	5.8230E - 07	4.1551E - 08	5.6309E - 23
				Success	20/20	20/20	20/20	20/20
				Time	0.718785	0.512242	31.075809	0.247456
15	Schwefel2.22	30	0	Ave.	2.3131E02	2.4395E02	2.9013E54	6.3587E - 06
				STD	1.3193E02	1.2341E02	1.2971E55	1.1936E - 05
				Best	6.7400E01	1.7354E01	3.6854E01	5.9410E - 08
				Success	0/20	0/20	0/20	20/20
				Time	128.445013	127.084387	79.924516	0.217104
16	Griewangk	30	0	Ave.	7.9510E - 01	7.7746E - 01	2.6148E01	3.3151E - 11
				STD	3.7583E - 01	2.5454E - 01	3.6626E01	1.4672E - 10
				Best	2.9324E - 01	3.2031E - 01	6.3912E - 05	0.0000
				Success	0/20	0/20	5/20	20/20
				Time	126.872461	126.210153	70.852376	0.211351
17	Sumsquare	30	0	Ave.	1.9818E03	4.6771E03	9.0499E05	4.2446E - 11
				STD	2.8370E03	6.7104E03	1.0253E06	1.2930E - 10
				Best	1.6463E01	2.0516E02	1.8730E02	1.49990E - 16
				Success	0/20	0/20	0/20	20/20
				Time	122.748646	125.154349	78.809270	0.273780
18	Sinusoidal	30	-3.5	Ave.	-4.2587E - 01	-3.7898E - 01	-2.449	-3.1030
				STD	2.6632E - 01	1.9791E - 01	1.0203	5.0473E - 05
				Best	-1.1922	-8.3111E - 01	-3.3087	-3.1032
				Success	20/20	20/20	20/20	20/20
				Time	0.204559	0.240200	0.234205	0.200361

Dim. denotes dimension. Opt. denotes optimum value.

TABLE 4: Simulation results of RIO, HRIO, CSO, MCSO, and ICSO.

SN	Function	Dim.	Opt.	RIO	HRIO	CSO	MCSO	ICSO
19	Zakharov	30	0	Ave.	1.0167E04	1.0216E04	6.3663E18	2.3878E - 09
				STD	3.8643E03	5.1012E03	2.2732E19	8.8529E - 09
				Best	2.6634E03	2.3151E03	1.3578E09	1.8549E - 25
				Success	0/20	0/20	0/20	20/20
				Time	115.192226	114.691827	79.926232	0.205280
20	Step	30	0	Ave.	0.0000	0.0000	2.0004E04	0.0000
				STD	0.0000	0.0000	8.4815E04	0.0000
				Best	0.0000	0.0000	0.0000	0.0000
				Success	20/20	20/20	16/20	20/20
				Time	0.686403	0.633264	39.136696	0.239525
21	Powell	24	0	Ave.	1.8348E - 03	3.7434E - 03	1.0840E08	2.6031E - 12
				STD	1.6248E - 03	6.1711E - 03	4.1180E08	6.9959E - 12
				Best	9.6693E - 05	6.8033E - 04	5.2392E01	1.2287E - 19
				Success	2/20	12/20	0/20	20/20
				Time	122.796991	92.876086	74.794730	1.527170
22	ST	9	0	Ave.	0.0000	0.0000	0.0000	0.0000
				STD	0.0000	0.0000	0.0000	0.0000
				Best	0.0000	0.0000	0.0000	0.0000
				Success	20/20	20/20	20/20	20/20
				Time	0.435911	0.426320	0.437944	0.431122
23	ST	17	0	Ave.	0.0000	0.0000	0.0000	0.0000
				STD	0.0000	0.0000	0.0000	0.0000
				Best	0.0000	0.0000	0.0000	0.0000
				Success	20/20	20/20	20/20	20/20
				Time	1.066161	1.052169	1.159830	1.089657

Dim. denotes dimension. Opt. denotes optimum value.

TABLE 5: Comparison of average performance of RIO, HRIO, CSO, MCSO, and ICSO.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO	Optimum
1	Bohachevsky1	3.4405E - 05	3.2877E - 04	2.9893E02	3.5153E - 09	0.0000	0
2	Bohachevsky2	4.2829E - 05	4.6703E - 04	9.0941E02	8.4459E - 12	0.0000	0
3	Bohachevsky3	5.3479E - 05	4.7575E - 04	7.4284E02	2.1388E - 14	0.0000	0
4	3 Hump camel back	1.4962E - 02	4.3021E - 04	5.003E09	7.098E - 11	5.9853E - 31	0
5	6 Hump camel back	-4.3522E - 01	-4.7652E - 01	1.5763E05	-1.0263E - 08	-2.9798E - 25	-1.03163
6	Easom	-1	-1	-4.3165E - 01	-1	-1	-1
7	Matyax	4.9470E - 05	3.2297E - 04	7.5712	2.6876E - 13	4.0732E - 35	0
8	Schaffer1	-1.9069	-1.6211	-2.9174E - 01	-1	-1	-1
9	Schaffer2	2.0179E - 03	1.6566E - 03	7.1618	3.3168E - 04	2.2149E - 09	0
10	Sphere	2.2168E - 05	1.6676E - 04	1.8123E02	1.5201E - 12	3.3448E - 34	0
11	Rastrigin	3.8135E - 05	3.2150E - 04	3.6022E03	9.1994E - 11	0.0000	0
12	Rosenbrock	2.5281E06	3.3571E06	9.5067E11	2.9000E01	2.9000E01	0
13	Ackley	2.0001E01	2.0005E01	1.9222E01	5.1593E - 06	1.0651E - 15	0
14	Quadric	2.4498E - 05	2.2711E - 04	3.4991E - 04	4.4754E - 13	7.2183E - 28	0
15	Schwefel2.22	2.3131E02	2.4395E02	2.9013E54	6.3587E - 06	6.0407E - 16	0
16	Griewangk	7.9510E - 01	7.7746E - 01	2.6148E01	3.3151E - 11	0.0000	0
17	Sumsquare	1.9818E03	4.6771E03	9.0499E05	4.2446E - 11	1.5600E - 24	0
18	Sinusoidal	-4.2587E - 01	-3.7898E - 01	-2.449	-3.1030	-3.1030	-3.5
19	Zakharov	1.0167E04	1.0216E04	6.3663E18	2.3878E - 09	4.1579E - 26	0
20	Step	0.0000	0.0000	2.0004E04	0.0000	0.0000	0
21	Powell	1.8348E - 03	3.7434E - 03	1.0840E08	2.6031E - 12	1.8207E - 24	0
22	ST9	0.0000	0.0000	0.0000	0.0000	0.0000	0
23	ST17	0.0000	0.0000	0.0000	0.0000	0.0000	0
Number of good optimums		4	4	2	7	23	

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

TABLE 6: Comparison of best performance of RIO, HRIO, CSO, MCSO, and ICSO.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO	Optimum
1	Bohachevsky1	$1.3520E - 07$	$5.2651E - 06$	$2.0651E - 05$	0.0000	0.0000	0
2	Bohachevsky2	$2.2910E - 06$	$9.374E - 06$	$1.3775E - 05$	0.0000	0.0000	0
3	Bohachevsky3	$3.1200E - 06$	$4.6981E - 05$	$2.3093E - 07$	0.0000	0.0000	0
4	3 hump camel back	$1.1739E - 06$	$2.2449E - 05$	$1.7642E - 05$	$3.1395E - 19$	2.2320E - 53	0
5	6 hump camel back	-1.0215	-1.0034	-9.4052E - 01	-1.9879E - 07	5.9589E - 24	-1.03163
6	Easom	-1	-1	-1	-1	-1	-1
7	Matyax	$6.2897E - 06$	$1.2684E - 05$	$8.8777E - 06$	$6.6695E - 21$	1.1292E - 55	0
8	Schaffer1	-2.7458	-2.7164	-2.7438	-1	-1	-1
9	Schaffer2	$6.2423E - 05$	$4.1422E - 04$	$2.8354E - 01$	$1.5810E - 05$	1.9383E - 14	0
10	Sphere	$5.7627E - 09$	$5.5635E - 08$	$4.9195E - 07$	$2.9978E - 24$	2.8205E - 54	0
12	Rosenbrock	$1.6773E04$	$3.7562E04$	$4.4068E01$	2.9000E01	2.9000E01	0
14	Quadric	$1.1360E - 08$	$5.8230E - 07$	$4.1551E - 08$	$5.6309E - 23$	5.910E - 52	0
15	Schwefel2.22	$6.7400E01$	$1.7354E01$	$3.6854E01$	$5.9410E - 08$	5.1670E - 24	0
16	Griewangk	$2.9324E - 01$	$3.2031E - 01$	$6.3912E - 05$	0.0000	0.0000	0
17	Sumsquare	$1.6463E01$	$2.0516E02$	$1.8730E02$	$1.49990E - 16$	1.3765E - 47	0
18	Sinusoidal	-1.1922	-8.3111E - 01	-3.3087	-3.1032	-3.1030	-3.5
19	Zakharov	$2.6634E03$	$2.3151E03$	$1.3578E09$	$2.0954E - 15$	6.3965E - 57	0
20	Step	0.0000	0.0000	0.0000	0.0000	0.0000	0
21	Powell	$9.6693E - 05$	$6.8033E - 04$	$5.2392E01$	$1.2287E - 19$	1.2265E - 54	0
22	ST9	0.0000	0.0000	0.0000	0.0000	0.0000	0
23	ST17	0.0000	0.0000	0.0000	0.0000	0.0000	0
Number of good optima		4	4	5	11	22	

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

TABLE 7: Comparison of standard deviation of mean global optimum values of RIO, HRIO, CSO, MCSO, and ICSO.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO
1	Bohachevsky1	$2.5963E - 05$	$3.0334E - 04$	$5.0332E02$	$1.4392E - 08$	0.0000
2	Bohachevsky2	$3.0070E - 05$	$3.4047E - 04$	$1.7794E03$	$2.9240E - 11$	0.0000
3	Bohachevsky3	$2.9141E - 05$	$2.3273E - 04$	$1.6739E03$	$4.8670E - 14$	0.0000
4	3 hump camel back	$6.6769E - 02$	$2.8371E - 04$	$1.7137E10$	$3.0201E - 10$	2.5457E - 30
5	6 hump camel back	$3.3322E - 01$	$3.1284E - 01$	$7.0503E05$	$4.4391E - 08$	1.3325E - 24
6	Easom	$3.7518E - 02$	$2.1031E - 02$	$3.4470E - 01$	$1.4897E - 08$	4.4116E - 17
7	Matyax	$3.0244E - 05$	$2.6018E - 04$	$1.1247E01$	$8.9347E - 13$	1.8125E - 34
8	Schaffer1	$7.0381E - 01$	$5.9214E - 01$	$7.5142E - 01$	$5.9575E - 07$	4.1325E - 15
9	Schaffer12	$2.6407E - 03$	$1.4451E - 03$	5.3095	$3.0328E - 04$	2.9483E - 09
10	Sphere	$2.4528E - 05$	$2.4018E - 04$	$8.1048E02$	$6.7224E - 12$	1.3324E - 33
11	Rastrigin	$3.4436E - 05$	$3.0003E - 04$	$5.5728E03$	$3.9456E - 10$	0.0000
12	Rosenbrock	$4.0528E06$	$7.1150E06$	$2.2713E12$	0.0000	0.0000
13	Ackley	$3.0455E - 03$	$1.5671E - 02$	5.8258	$1.9149E - 05$	7.9441E - 16
14	Quadric	$2.7957E - 05$	$2.3635E - 04$	$3.3725E - 04$	$1.9751E - 12$	3.2218E - 27
15	Schwefel2.22	$1.3193E02$	$1.2341E02$	$1.2971E55$	$1.1936E - 05$	1.2203E - 15
16	Griewangk	$3.7583E - 01$	$2.5454E - 01$	$3.6626E01$	$1.4672E - 10$	0.0000
17	Sumsquare	$2.8370E03$	$6.7104E03$	$1.0253E06$	$1.2930E - 10$	6.9785E - 24
18	Sinusoidal	$2.6632E - 01$	$1.9791E - 01$	1.0203	$5.0473E - 05$	1.9436E - 14
19	Zakharov	$3.8643E03$	$5.1012E03$	$2.2732E19$	$8.8529E - 09$	1.8549E - 25
20	Step	0.0000	0.0000	$8.4815E04$	0.0000	0.0000
21	Powell	$1.6248E - 03$	$6.1711E - 03$	$4.1180E08$	$6.9959E - 12$	5.6824E - 24
22	ST9	0.0000	0.0000	0.0000	0.0000	0.0000
23	ST17	0.0000	0.0000	0.0000	0.0000	0.0000
Number of good STD		2	2	2	4	23

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

TABLE 8: Comparison of success performance of RIO, HRIO, CSO, MCSO, and ICSO.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO
1	Bohachevsky1	1	1	2.5	1	1
2	Bohachevsky2	1	1	0.2	1	1
3	Bohachevsky3	1	1	0.15	1	1
4	3 hump camel back	0.95	1	0.6	1	1
5	6 hump camel back	1	1	0.95	1	1
6	Easom	1	1	1	1	1
7	Matyax	1	1	0.55	1	1
8	Schaffer1	1	1	1	1	1
9	Schaffer2	0.1	0.65	0	1	1
10	Sphere	1	1	0.95	1	1
11	Rastrigin	1	1	0.25	1	1
12	Rosenbrock	0	0	0	0	0
13	Ackley	0	0	0	1	1
14	Quadric	1	1	1	1	1
15	Schwefel2.22	0	0	0	1	1
16	Griewangk	0	0	0.25	1	1
17	Sumsquare	0	0	0	1	1
18	Sinusoidal	1	1	1	1	1
19	Zakharov	0	0	0	1	1
20	Step	1	1	0.8	1	1
21	Powell	0.1	0.6	0	1	1
22	ST9	1	1	1	1	1
23	ST17	1	1	1	1	1
Number of 100% success rates		14	15	6	22	22

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

TABLE 9: Comparison of execution time of RIO, HRIO, CSO, MCSO, and ICSO.

SN	Function	RIO	HRIO	CSO	MCSO	ICSO
1	Bohachevsky1	1.137525	0.886356	23.913237	0.075212	0.097187
2	Bohachevsky2	0.998178	0.946887	26.492095	0.072021	0.074106
3	Bohachevsky3	1.089920	0.885252	25.028054	0.080908	0.068189
4	3 hump camel back	4.231533	0.794983	18.281683	0.104132	0.078845
5	6 hump camel back	0.406355	0.330198	5.723039	0.0945856	0.086637
6	Easom	0.124022	0.107303	0.106738	0.077179	0.092393
7	Matyax	0.973322	0.711734	13.559576	0.88536	0.076693
8	Schaffer1	0.109048	0.086433	0.119076	0.072400	0.081599
9	Schaffer2	62.567654	31.415836	29.194283	0.084127	0.082320
10	Sphere	0.617544	0.557871	25.378161	0.82512	0.199373
11	Rastrigin	0.956329	0.826770	71.811170	0.175563	0.369987
12	Rosenbrock	126.618734	127.469638	81.361663	76.084929	78.572185
13	Ackley	122.216187	117.635854	82.227210	0.235012	0.192339
14	Quadric	0.718785	0.512242	31.075809	0.247456	0.227244
15	Schwefel2.22	128.445013	127.084387	79.924516	0.217104	0.219296
16	Griewangk	126.872461	126.210153	70.852376	0.211351	0.210934
17	Sumsquare	122.748646	125.154349	78.809270	0.273780	0.236129
18	Sinusoidal	0.204559	0.240200	0.234205	0.200361	0.217635
19	Zakharov	115.192226	114.691827	79.926232	0.205280	0.259202
20	Step	0.686403	0.633264	39.136696	0.239525	0.225102
21	Powell	122.796991	92.876086	74.794730	1.527170	0.853751
22	ST9	0.435911	0.426320	0.437944	0.431122	0.436741
23	ST17	1.066161	1.052169	1.159830	1.089657	1.147114
Number of minimum execution times		—	2	—	9	12

ST9 denotes Storn's Tchebychev 9. ST17 denotes Storn's Tchebychev 17.

TABLE 10: Jonckheere-Terpstra test statistics^a.

	Fitness
Number of levels in algorithm	5
N	114
Observed J-T statistic	1952.000
Mean J-T statistic	2599.500
STD of J-T statistic	199.355
Standard data of J-T statistic	−3.245
Asymp. Sig. (2-tailed)	0.001

^aGrouping variable: algorithm.

4. Conclusion

Cockroach swarm optimization algorithm is extended in this paper with a new component called hunger component. Hunger component enhances the algorithm diversity and searching capability. An improved cockroach swarm optimization algorithm is proposed. The efficiency of the proposed algorithm is shown through empirical studies where its performance was compared with that of existing algorithms, that is, CSO, MSCO, RIO, and HRIO. Results show its outstanding performance compared to the existing algorithms. Application of the algorithm to real life problems can be considered in further studies.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors thank College of Agriculture, Engineering and Science, University of KwaZulu-Natal, for supporting this paper through bursary award.

References

- [1] S. Camazine, J. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*, Princeton University Press, Princeton, NJ, USA, 2001.
- [2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *IEEE Neural Networks Proceedings*, vol. 4, pp. 1942–1948, 1995.
- [3] F. Yoshikazu, N. Hideyuki, and T. Yuji, "Particle swarm optimization for the optimal operational planning of energy plants," in *Innovations in Swarm Intelligence Studies in Computational Intelligence*, C. P. Lim, L. C. Jain, and S. Dehuri, Eds., pp. 159–174, Springer, Berlin, Germany, 2009.
- [4] P. Chen, "Particle swarm optimization for power dispatch with pumped hydro," in *Particle Swarm Optimization*, A. Lazinica, Ed., pp. 131–144, In-Tech, Zagreb, Croatia, 2009.
- [5] I. Omar and B. Cees, "A particle optimization approach to graph permutation," in *Particle Swarm Optimization*, A. Lazinica, Ed., pp. 291–312, In-Tech, Zagreb, Croatia, 2009.
- [6] M. Dorigo, *Optimization, learning and natural algorithms [Ph.D. thesis]*, Politecnico di Milano, Milan, Italy, 1992.
- [7] A. Rizzoli A, R. Montemanni, E. Lucibello, and L. Gambardella, *Ant Colony Optimization for Real-World Vehicle Routing Problems: from Theory to Applications*, Springer, 2007.
- [8] A. A. A. Radwan, T. M. Mahmoud, and E. H. Hussein, "AntNet-RSLR: a proposed Ant routing protocol for MANETs," in *Proceedings of the IEEE Saudi International Electronics, Communications and Photonics Conference (SIECPC '11)*, April 2011.
- [9] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm," Tech. Rep., Manufacturing Engineering Centre, Cardiff University, Cardiff, UK, 2005.
- [10] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.
- [11] D. Karaboga and B. Akay, "A survey: algorithms simulating bee swarm intelligence," *Artificial Intelligence Review*, vol. 31, no. 1–4, pp. 61–85, 2009.
- [12] B. Basturk and D. Karaboga, "An Artificial Bees Colony (ABC) algorithm for numeric computation," in *Proceedings of the IEEE Swarm Intelligence Symposium*, Indianapolis, Ind, USA, 2006.
- [13] T. C. Havens, C. J. Spain, N. G. Salmon, and J. M. Keller, "Roach infestation optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '08)*, September 2008.
- [14] C. ZhaoHui and T. HaiYan, "Cockroach swarm optimization," in *Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICCET '10)*, vol. 6, pp. 652–655, April 2010.
- [15] C. ZhaoHui, "A modified cockroach swarm optimization," *Energy Procedia*, vol. 11, p. 49, 2011.
- [16] I. C. Obagbuwa, A. O. Adewumi, and A. A. Adebiyi, "A dynamic step-size adaptation roach infestation optimization," in *Proceedings of the IEEE International Conference on Advance Computing (IACC '14)*, Gurgaon, Indian, 2014.
- [17] L. Cheng, Z. Wang, S. Yanhong, and A. Guo, "Cockroach swarm optimization algorithm for TSP," *Advanced Engineering Forum*, vol. 1, pp. 226–229, 2011.
- [18] C. ZhaoHui and T. HaiYan, "Cockroach swarm optimization for vehicle routing problems," *Energy Procedia*, vol. 13, pp. 30–35, 2011.
- [19] I. C. Obagbuwa, A. O. Adewumi, and A. A. Adebiyi, "Stochastic constriction cockroach swarm optimization for multidimensional space function problems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 430949, 12 pages, 2014.
- [20] J. B. Williams, M. Louis, R. Christine, and A. Nalepal, *Cock-Roaches Ecology, Behaviour and Natural History*, Johns Hopkins University Press, Baltimore, Md, USA, 2007.
- [21] M. Kerckhove, "From population dynamics to partial differential equations," *The Mathematica Journal*, vol. 14, 2012.
- [22] M. M. Ali, C. Khompatraporn, and Z. B. Zabinsky, "A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems," *Journal of Global Optimization*, vol. 31, no. 4, pp. 635–672, 2005.
- [23] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [24] P. Civicioglu and E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, vol. 39, pp. 1–32, 2013.

- [25] J. Cohen, *Statistical Power Analysis for the Behavioural Science*, L. Erlbaum Associates, Hillsdale, NJ, USA, 2nd edition, 1988.
- [26] J. Cohen, "Statistical power analysis for the behavioural science," *Current Direction in Psychological Science*, vol. 1, no. 3, pp. 98–101, 1992.

