Deb, Sanchari; Gao, Xiao Zhi; Tammi, Kari; Kalita, Karuna; Mahanta, Pinakeswar

A New Teaching–Learning-based Chicken Swarm Optimization Algorithm

Published in:
Soft Computing

DOI:
10.1007/s00500-019-04280-0

Published: 01/01/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Deb, S., Gao, X. Z., Tammi, K., Kalita, K., & Mahanta, P. (2019). A New Teaching–Learning-based Chicken Swarm Optimization Algorithm. *Soft Computing*. https://doi.org/10.1007/s00500-019-04280-0

A New Teaching-Learning-based Chicken Swarm Optimization Algorithm

Sanchari Deb[a*], Xiao-Zhi Gao[b], Kari Tammi[c], Karuna Kalita[d], Pinakeswar Mahanta[d]

[a]Centre of Energy Indian Institute of Technology, Guwahati, India

[b]School of Computing, University of Eastern Finland, Finland

[c]Department of Mechanical Engineering, Aalto University, Finland

[d]Department of Mechanical Engineering, Indian Institute of Technology, Guwahati, India

Corresponding Author: Sanchari Deb

Centre for Energy Indian Institute of Technology, Guwahati, India

Email id: sancharideb@yahoo.co.in

A New Teaching-Learning-based Chicken Swarm Optimization Algorithm

Abstract-Chicken Swarm Optimization (CSO) is a novel swarm intelligence based algorithm known for its good performance on many benchmark functions as well as real world optimization problems. However, it is observed that CSO sometimes gets trapped in local optima. This work proposes an improved version of the CSO algorithm with modified update equation of the roosters and a novel constraint handling mechanism. Further, the work also proposes synergy of the improved version of CSO (ICSO) with Teaching Learning Based Optimization (TLBO) algorithm. The proposed ICSOTLBO algorithm possesses the strengths of both CSO and TLBO. The efficacy of the proposed algorithm is tested on eight basic benchmark functions, fifteen computationally expensive benchmark functions as well as two real-world problems. Further, the performance of ICSOTLBO is also compared with a number of state of art algorithms. It is observed that the proposed algorithm performs better than or as good as many of the existing algorithms.

Keywords

Algorithm, Benchmark, Chicken Swarm Optimization, Function, Hybrid,Teaching Learning Based Optimization

Abbreviations

Abbreviation	Full form	Abbreviation	Full form
GA	Genetic Algorithm	APSO	Adaptive Particle Swarm Optimization
SA	Simulated Annealing	OLPSO	Orthogonal Particle Swarm Optimization
GSA	Gravitational Search Algorithm	CLPSO	Comprehensive Learning Particle Swarm Optimization
PSO	Particle Swarm Optimization	CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CS	Cuckoo Search	SPC-PNX	Real Parameter Genetic Algorithm
EHO	Elephant Herding Optimization	BPSOGSA	Binary Particle Swarm Optimization Gravitational Search Algorithm
EWA	Earthworm Optimization Algorithm	BGSA	Binary Gravitational Search Algorithm
GWO	Grey Wolf Optimization	SD	Standard Deviation
WOA	Whale Optimization Algorithm	EV	Electric Vehicle
ABC	Artificial Bee Colony	RCCRO	Real Coded Chemical Reaction Optimization
BSA	Bird Swarm Algorithm	HSA	Harmony Search Algorithm
CSO	Chicken Swarm Optimization		
ICSO	Improved Chicken Swarm Optimization		
DE	Differential Evolution		
BA	Bat algorithm		
IRRO	Improved Raven Rooster Optimization		
NFL	No Free Lunch		
TLBO	Teaching Learning Based Optimization		
mTLBO	Modified Teaching Learning Based Optimization		
ICSOTLBO	Improved Chicken Swarm Optimization Teaching Learning Based Optimization		
SaDE	Self Adaptive Differential Evolution		
jDE	New Self Adaptive Differential Evolution		
EPSDE	Differential Evolution with Ensemble of Parameter		

1. Introduction

In recent years, nature inspired optimization methods have gathered attention of the masses. Most of the nature inspired optimization approaches mimic a natural phenomenon or social behaviour of a group of animals. For example, Genetic Algorithm (GA) mimics the natural phenomenon of survival of the fittest (Goldberg & Holland 1988), Simulated Annealing (SA) is inspired from the process of annealing of solids (Laarhoven et al. 1987), Gravitational Search Algorithm (GSA) is inspired by the gravitational laws, and interaction between the masses (Rashedi et al. 2009), Particle Swarm Optimization (PSO) mimics the phenomenon of bird flocking (Poli et al. 2007), Cuckoo Search (CS) mimics the brood parasitism of cuckoo (Yang & Deb 2009; Yang & Deb 2014), Elephant Herding Optimization (EHO) uses the herding behaviour of elephants (Wang et al. 2016; Wang et al. 2015), Earthworm Optimization (EWA) inspired by the burrowing action of earthworms in the soil (Wang et al. 2015) , Grey Wolf Optimization (GWO) mimics the hunting of grey wolf (Mirjalili et al. 2014; Faris et al. 2018), Whale Optimization Algorithm (WOA) mimics the social behaviour of whales (Mirjalili & Lewis 2016), Artificial Bee Colony (ABC) imitates the foraging behaviour of honeybee (Karaboga & Basturk 2008), Bird Swarm Algorithm (BSA) utilizes the social interaction in a bird swarm(Meng et al. 2016), Bat Algorithm (BA) mimics the echolocation behaviour of bats (Cai et al. 2016), Harmony Search Algorithm (HSA) mimics the natural phenomenon of musicians improvisation of the harmony (Gao et al. 2015) etc.

One such nature inspired algorithm that has gained popularity in the recent years is CSO (Meng et al. 2014). CSO efficientlyexploits the hierarchal order in the chicken swarm and the food-searching process of the chicken swarm. In the aforementioned algorithm, the positions of the members of the chicken swarm are regarded as the candidate solutions of the optimization problem to be solved. The chicken swarm is divided into rooster, hens, and chicks depending upon the food searching capability. The competition between different chickens under a specific hierarchal order and mother-child relationship is also taken into account in this algorithm. A number of variants of the CSO algorithm are also available in the existing literature. Deb et al. (2019) presented a comprehensive overview of different variants of CSO algorithm and concluded that there is still scope of improving the algorithm. Chen et al. (2015) proposed an improved version of CSO with modified update equation of the hen. Wang et al. (2017) introduced the mutation strategy in the update phenomenon of chicks to enhance their food searching ability. Han et al. (2017) also proposed an improved binary version of CSO where the mutation operator is applied to the population with the worst fitness value.Ahmed et al. (2017) combined chaos tent map and logistic map with CSO and used the algorithm to solve feature selection problem.Liang et al. (2016) replaced the update mechanism of roosters with the update mechanism of Bat Algorithm (BA) and proposed hybrid Bat CSO. Kumar &Veni(2018) hybridized CSO with Differential Evolution (DE) and applied the proposed algorithm for solving routing problem. Experimental results showed that the proposed algorithm performed better than the standalone algorithms as the solutions obtained by CSO were further fine-tuned by DE to avoid premature convergence. Torabi&Esfahani (2018) hybridized Improved Raven Rooster Optimization (IRRO) with CSO and utilized the proposed algorithm for solving task scheduling problems.

It is observed that in the existing literature a number of nature inspired algorithms are available. Despite the availability of such a wide range of nature inspired algorithms researchers are still trying to develop new more efficient algorithms, improve the existing algorithms by hybridization or modifying some algorithmic components of the methods. The main motivation behind this lies in the No Free Lunch (NFL) theorem (Wolpert& Macready1997). NFL theorem concludes that a single algorithm cannot perform well on all the optimization problems. Hence, there is necessity of developing new more efficient algorithms and improving the existing algorithms. The present work is also concerned with improving CSO and its hybridization with TLBO. CSO has good utilization rate of population. However, the algorithm may sometimes get trapped in local optima. Researchers have tried to overcome this inherent drawback of CSO by variety of ways (Chen et al. 2015; Wang et al. 2017; Han et al. 2017; Liang et al. 2016; Kumar &Veni 2018; Torabi & Esfahani 2018). Some of the variants of CSO are listed in Table 1.The present work also makes an attempt to improve the basic CSO by modifying the update equation of roosters and introducing a novel constraint handling mechanism. Further, the work also proposes synergy of the improved version of CSO (ICSO) with Teaching Learning Based Optimization (TLBO) algorithm. The salient features of the proposed algorithm in comparison with the existing improvements of CSO are modified update equation of rooster, novel constraint handling mechanism and hybridization with TLBO.The contributions of the work as compared to the existing works on CSO are summarized as follows:

1. Improvement of CSO by modifying the update equation of rooster and introduction of a novel constraint handling mechanism.

2. Hybridization of ICSO with TLBO. It is expected that synergy of ICSO and TLBO will enhance the utilization rate of population and overcome the premature convergence of the algorithm.

3. The proposed algorithm is used for solving eight basic benchmark functions, fifteen computationally expensive benchmark functions as well as two real- world problems.

4. The performance of the proposed algorithm is statistically compared with other state of art algorithms like PSO and its variants, DE and its variants, GA, TLBO and its variants.

Table 1- Variants of CSO

Author	Improvement
Chen et al. (2015)	Modification in the update equation of the hen
Wang et al. (2017)	Introduction of mutation strategy in the update equation of chicks
Han et al. (2017)	Development of binary version of CSO
Ahmed et al. (2017)	Development of chaotic CSO
Liang et al. (2016)	Hybridization of Bat algorithm with CSO
Kumar &Veni (2018)	Hybridization of DE with CSO
Torabi&Esfahani (2018)	Hybridization of IRRO with CSO

The remainder of the paper is organized as follows. Section 2, Section 3 and Section 4 illustrate the fundamentals of CSO, ICSO and TLBO respectively. Section 5 elaborates the hybridization of ICSO with TLBO. Section 6 and section 7 illustrates the results related to the performance of the proposed algorithm on basic benchmark functions and computationally expensive benchmark functions respectively. Section 8 discusses the computational complexity of the proposed algorithm. Section 9illustrates the performance of the proposed algorithm on real- world problems like Charging Station Placement and Economic Load Dispatch problem. Section 10 presents the future direction of research on CSO. Finally, section 11 concludes the work.

2. CSO

CSO is one of the latest swarm intelligence based algorithms developed by Meng et al. in the year 2014. The hierarchal order prevalent in the chicken swarm and the collective food searching mechanism of the swarm are mimicked by the algorithm. The entire populace of chicken in the group is segregated into dominant rooster, hens, and chicks depending upon the fitness values of the chickens. The chickens with highest food searching ability or fitness are designated as roosters, chickens with least food searching ability or fitness are designated as chicks, and the chickens with intermediate food searching ability or fitness are assigned as hens. The mother-child relationship is also established randomly. The hierarchal order and mother-child relationship are updated after every *G* time steps. The natal behavior of hens to go behind their group mate rooster and chicks to go behind their mother in the quest for food is utilized effectively in the algorithm. It is also presumed that the chickens would try to scratch the food found by others thereby giving rise to a competition for food in the group. The algorithm is divided into two steps- Initialization and Update.

In Initialization, the population size and other related parameters of CSO like number of roosters, number of hens, number of chicks, number of mother hens, *G* is first defined. The fitness values of the randomly generated initial population of chicken are evaluated and a hierarchal order is established based on this fitness value. The algorithm is based on the following assumptions-

- The number of hens is highest in the group
- All the hens are not mother hens
- The mother hens are selected randomly from the set of hens
- The number of chicks is less than hen

There is variation in the food searching capacity of roosters, hens, and chicks. In the update step, the fitness values of the initial population are updated depending on the food searching capacity of the different members of the group.

Food searching capacity of rooster depends on their fitness value and their update formula is as follows:

$$x_{i,j}^{t+1} = x_{i,j}^{t} \times (1 + \mathrm{randn}(0, \sigma^2)) \tag{1}$$

If $f_i \leq f_k$

$$\sigma^2 = 1 \tag{2}$$

Else,

$$\sigma^2 = \exp\left(\frac{(f_k - f_i)}{|f_i| + \varepsilon}\right) \tag{3}$$

where randn$(0,\sigma^2)$ is a Gaussian distribution function with mean 0 and standard deviation σ^2. f is the fitness value of corresponding x, k is randomly selected rooster's index.ϵ is a small constant value which is used to avoid zero division error.

Hens follow their group mate roosters in their quest for food. Moreover, there is also a tendency among the chickens to steal the food found by other chickens. The mathematical representation of their update formula is as follows-

$$x_{i,j}^{t+1} = x_{i,j}^{t} + S1 \times \text{rand} \times (x_{r1,j}^{t} - x_{i,j}^{t}) + S2 \times \text{rand}(x_{r2,j}^{t} - x_{i,j}^{t}) \quad (4)$$

$$S1 = \exp\left(\frac{f_i - f_{r1}}{abs(f_i) + \varepsilon}\right) \quad (5)$$

$$S2 = \exp(f_{r2} - f_i) \quad (6)$$

where rand is a randomly generated number between 0 and 1. $r1 \in [1, N]$ is an index of rooster which is i^{th} hen's group mate. And $r2 \in [1, N]$ is an index of rooster or hen which is randomly chosen such that *r1* is not equal to *r2*.

The natural tendency of chicks to follow their mother is mathematically formulated as follows-

$$x_{i,j}^{t+1} = x_{i,j}^{t} + FL \times (x_{m,j}^{t} - x_{i,j}^{t}) \quad (7)$$

where $x_{m,j}^{t}$ represents the position of i^{th} chick's mother. *FL* is a parameter which signifies that the chick would follow its mother. *FL* is generally chosen in between 0 and 2.

The pseudo code of CSO is as shown in Algorithm 1

Algorithm 1-Pseudo code of CSO(Meng et al. 2014)

```
Initialize the population of chicken having size N and define other algorithm specific parameters like G, size of RN, HN,CN, and MN;
Evaluate the fitness value of N chicken, t=0 , establish the hierarchal order in the swarm as well as mother child relationship;
While (t<gen)
t=t+1;
If(t%G==0)
Establish the hierarchal order in the swarm as well as mother child relationship;
Else
For i=1:PN
If i==rooster
Update its solution by Eq.(1);
End if
If i==hen
Update its solution by Eq.(4);
End if
If i==chick
Update its solution by Eq.(7);
End if
Evaluate the new solutions;
Update the new solutions if they are better than the previous one;
End for
End if else
End while
```

3. ICSO

The key features of ICSO are modification in the update mechanism of roosters and a novel constraint handling mechanism. In the basic CSO, hens follow their group mate rooster in the food searching process. And the chicks follow their mother hen. Thus, it is obvious that the performance of the algorithm is very much dependent on roosters. If the roosters get struck in local optima then there is possibility of premature convergence. Hence, in order of overcome this drawback authors have modified the update equation of roosters. The algorithm considers that the roosters would utilize its previous experience in the food searching process. In the quest for food, each rooster can record and update its best experience from the past and the swarms' previous best experience about food availability. Social information is shared instantaneously among the roosters. Thus, the update equation of the roosters is modified as:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + C \times rand \times (p_{i,j} - x_{i,j}^{t}) + S \times rand(g_j - x_{i,j}^{t}) \tag{8}$$

where $p_{i,j}$ represents the best previous position of i^{th} rooster, g_j represents the best previous position of the swarm, C represents cognitive co-efficient and S represents social co-efficient.

Another salient feature of ICSO is a novel constraint handling mechanism. In the basic CSO, whenever the updated value of the decision variable exceeds the upper or lower limit of the decision variable then it is fixed to upper or lower limit respectively. In ICSO an improved an efficient methodology of constraint handling is used to improve the convergence speed. This improved methodology of constraint handling is as shown in Algorithm 2. The pseudo code of ICSO is as shown in Algorithm 3.

Algorithm 2-Pseudo code of improved constraint handling (Chen et al. 2016)

if $x_{i,j}^{t} < lb \,||\, x_{i,j}^{t} < ub$ *lb and ub represents lower and upper bound of the decision variable*

$w = const \times \left| g_j - p_{i,j} \right|$

$temp = g_j + w \times rand$

if $lb \le temp \le ub$

$x_{i,j}^{t} = temp$

else

$x_{i,j}^{t} = p_{i,j}$

end if

end if

Algorithm 3-Pseudo code of ICSO

Initialize the population of chicken having size N and define other algorithm specific parameters like G, size of RN, HN,CN, and MN;
Evaluate the fitness value of N chicken, t=0 , establish the hierarchal order in the swarm as well as mother child relationship;
While (t<gen)
t=t+1;
If(t%G==0)
Establish the hierarchal order in the swarm as well as mother child relationship;
Else
For i=1:PN
If i==rooster
Update its solution by Eq. (8);
End if
Perform constraint handling by Algorithm 2
If i==hen
Update its solution by Eq.(4);
End if
Perform constraint handling by Algorithm 2
If i==chick
Update its solution by Eq.(7);
End if
Perform constraint handling by Algorithm 2
Evaluate the new solutions;
Update the new solutions if they are better than the previous one;
End for
Perform constraint handling by Algorithm 2
End if else
End while

4. TLBO

TLBO is a latest evolutionary algorithm introduced by Rao et al. in the year 2011. TLBO is a population-based evolutionary algorithm which mimics the interactive process of teaching and learning. A class of learners constitutes the population here. The teacher transfers his/her knowledge to the learners. The performance of the learners depends on the knowledge and capability of the teacher. The students can learn from the teacher as well as learn from each other through mutual interaction. Thus, the algorithm is divided into two parts- Teacher phase and Learner phase (Rao et al. 2011; Rao et al. 2016).

In the teacher phase, the students learn from the teacher who is an erudite scholar with profound knowledge and skill. The learner having the best fitness in a randomly generated population of teachers is generally assigned the role of teacher. Each learner learns from the teacher and is modified as follows-

$$Z_{diff} = \text{rand} \times (T_k - R_t m_k) \tag{9}$$

$$Z_{new} = Z_{old} + Z_{diff} \tag{10}$$

And the objective function value for each learner set modified by transfer of knowledge by the teacher is recalculated. If the new value of the objective function for any learner is better than the previous one then it is replaced by the new value. Else, the old learner is kept as it is.

In the learner phase, the learner learns by mutual interaction among themselves. For each learner Z_i any learner Z_j is chosen arbitrarily from the learner matrix. The objective function values are compared arbitrarily for the two aforementioned learners. If the value of the objective function of Z_i is lower than the objective function of Z_j then the i^{th} learner is modified as follows-

$$Z_{new}=Z_{old}+\text{rand}\times(Z_i-Z_j) \tag{11}$$

Else it is modified as-

$$Z_{new}=Z_{old}+\text{rand}\times(Z_j-Z_i) \tag{12}$$

The pseudo code and flowchart of TLBO is as shown in Algorithm 4

Algorithm 4- Pseudo code of TLBO(Rao et al. 2011; Rao et al. 2016)

Set k=1;
Initialize the population size and generate the initial population of students randomly;
Compute the objective function for all the individuals of the population;
while(k<gen)
{Teacher Phase}
Assign the teacher based on the fitness value;
for i=1:pop
Modify each learner by Eq.(9), Eq.(10);
Evaluate the new solutions;
Update the new solutions if they are better than the previous one;
{End of teacher phase}
{Learner Phase}
Choose two learners Z_i and Z_j, $i\neq j$;
if(fitness of Z_i better than Z_j)
Replace i^{th} learner by Eq.(11);
Else
Replace i^{th} learner by Eq. (12);
End if else
End for
k=k+1
End while

5. ICSOTLBO

Standalone Nature Inspired Optimization (NIO) algorithms are sometimes not efficient enough to handle the uncertainty of the practical optimization problems. Hybridization of NIO algorithms offers competitive solutions than standalone NIO algorithms in case of practical problems. Also, the hybrid algorithms inherit the advantages of two standalone algorithms, eliminate the limitations of the standalone algorithms, and perform better than the standalone algorithms. A good balance between exploration and exploitation is maintained in the hybrid algorithms.Hybridization of CSO and TLBO is also presented in the work. It is expected that the grading mechanism of ICSO when introduced in TLBO the utilization rate of population will increase.Hence, in ICSOTLBO, TLBO is performed in all the generations and ICSO is periodically invoked in some generations. The salient features of the proposed ICSOTLBO algorithm are

1. In the hybridization scheme, TLBO is performed in all generations and CSO is periodically invoked.
2. The algorithm is expected to have good utilization rate of population due to the grading mechanism of CSO

3. The premature convergence that is a drawback of CSO can be avoided in ICSOTLBO because of amalgamation of CSO with TLBO
4. The modified update equation of CSO utilizing the best experience from the past and the swarms' previous best experience about food availability will also enhance the performance of the algorithm
5. The new constraint handling mechanism will improve the convergence speed of the algorithm

The scheme for hybridizing ICSO and TLBO is elaborated by Algorithm 5.

Algorithm 5- Pseudo code of hybridizing scheme utilized in ICSOTLBO

Initialize the population size, gen and the other algorithm specific parameters of ICSOTLBO
Set t=1
While (t<gen)
Activate TLBO
If (t mod INV)>0
Activate ICSO
End if
t=t+1
End while

6. Performance of ICSOTLBO on basic benchmark functions

The proposed algorithm was first tested on six basic benchmark functions as shown in Table 2. In Table 2, f1, f2, f3, f4, f6 are unimodal functions and f5, f7, and f8 are a multimodal function. The detailed formulations of these benchmark functions can be found in the reference (Mirjalili 2016). The different algorithm-specific parameters of ICSOTLBO were tuned as in Table 3. The performance of ICSOTLBO on these basic benchmark functions was compared with a number of state of art algorithms like different variants of PSO, DE, GA etc. The results related to these comparisons are presented in the subsequent subsections.

Table 2- List of basic benchmark functions

Function no	Function name	Bounds	f_min
f1	Sphere	[-100 100]	0
f2	Schwefel 2.22	[-10 10]	0
f3	Schwefel 1.2	[-100 100]	0
f4	Step	[-100 100]	0
f5	Rastrigin	[-5.12 5.12]	0
f6	Schwefel 2.21	[-100 100]	0
f7	Ackley	[-32 32]	0
f8	Griewank	[-600 600]	0

Table 3- Algorithm specific parameters of ICSOTLBO

Parameter	Value
RN	0.2PN
HN	0.6PN
CN	PN-RN-HN
MN	0.1PN
S	2
C	2
G	10
INV	25

6.1. Comparison of ICSOTLBO with different variants of DE

The performance of ICSOTLBO was compared with SaDE, jDE and EPSDE on 8 benchmark functions shown in Table 2. The results of SaDE, jDE and EPSDE were directly taken from the reference (Satapathy & Naik 2014). For fair comparison, the population size and number of function evaluations of ICSOTLBO were kept same as in the reference (Satapathy & Naik 2014). The mean and standard deviations of the errors are reported in Table 4 for each of the basic benchmark functions shown in Table 2. Further, Wilcoxon's rank sum test was conducted at 0.05 significance level between ICSOTLBO and each of SaDE, jDE and EPSDE. The results of the Wilcoxon's rank sum test are reported in the last three rows of Table 4. It was observed that ICSOTLBO was always better than SaDE and jDE. And, ICSOTLBO was better than EPSDE for 5 benchmark functions and equivalent to EPSDE for 3 benchmark functions. For comparing the performance of the proposed algorithm with the variants of DE, Friedman test was performed. The ranks of the different algorithms obtained by Friedman test is as shown in Fig.1. It was observed that ICSOTLBO had obtained the best rank in comparison to the different variants of DE.

Table 4- Comparison of ICSOTLBO with different variants of DE (D=30, PN=20)

Function	FE	SaDE			jDE			EPSDE			ICSOTLBO	
		Mean	SD	+-=	Mean	SD	+-=	Mean	SD	+-=	Mean	SD
f1	1.5e+05	4.5e-20	1.9e-14	-	2.5e-28	3.5e-28	-	1.53e-85	9.01e-86	-	0	0
f2	2e+05	1.9e-14	1.1e-14	-	1.5e-23	1.0e-23	-	3.18e-54	3.11e-54	-	0	0
f3	5e+05	9e-37	5.4e-36	-	5.2e-14	1.1e-13	-	4.81e-76	1.9e-76	-	0	0
f4	1e+04	9.3e+02	1.8e+02	-	1e+03	2.2e+02	-	0	0	=	0	0
f5	1e+05	1.2e-03	6.5e-04	-	1.5e-04	2e-04	-	0	0	=	0	0
f6	5e+05	7.4e-11	1.82e-10	-	1.4e-15	1e-15	-	1.94e-02	8.90e-4	-	0	0
f7	5e+04	2.7e-03	5.1e-04	-	3.5e-04	1e-04	-	5.36e-13	4.77e-14	-	0	0
f8	5e+04	7.8e-04	1.2e-03	-	1.9e-05	5.8e-05	-	0	0	-	0	0
-	8				8			5			+ indicates better	
+	0				0			0			- indicates worse	
=	0				0			3			= indicates equivalent	

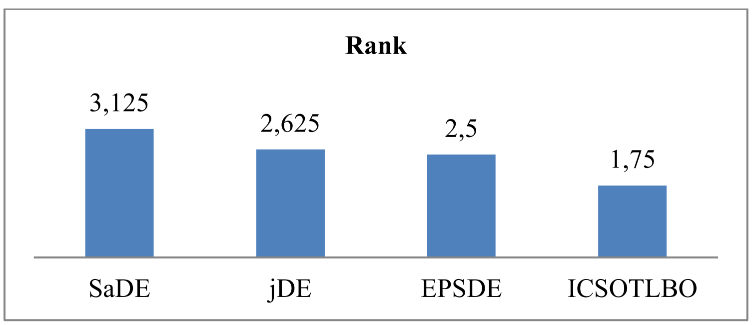

Fig.1-Comparison of Friedman ranks of ICSOTLBO with different variants of DE for basic benchmark functions

6.2. Comparison of ICSOTLBO with different variants of PSO

The performance of ICSOTLBO was compared with APSO, OLPSO, and CLPSO on the benchmark functions shown in Table 2. The results of APSO, OLPSO, and CLPSO were directly taken from the reference (Satapathy & Naik 2014). For fair comparison, the population size and number of function evaluations of ICSOTLBO were kept same as in the reference (Satapathy & Naik 2014). The mean and standard deviations of the errors are reported in Table 5 for each of the basic benchmark functions shown in Table 2. Further, Wilcoxon's rank sum test was conducted at 0.05 significance level between ICSOTLBO and each of APSO, OLPSO, and CLPSO. The results of the Wilcoxon's rank sum test are reported in the last three rows of Table 5. It was observed that ICSOTLBO was always better than APSO, OLPSO, and CLPSO on 6,3, and 6 benchmark functions respectively. ICSOTLBO performed equivalent to APSO, OLPSO, and CLPSO on 1,2, and 1 benchmark functions respectively. For comparing the performance of the proposed algorithm with the variants of PSO, Friedman test was also performed. The ranks of the different algorithms obtained by Friedman test is as shown in Fig.2. It was observed that ICSOTLBO had obtained the best rank in comparison to the different variants of PSO.

Table 5- Comparison of ICSOTLBO with different variants of PSO (D=30, PN=20, FE=2e+05)

Function	APSO			OLPSO			CLPSO			ICSOTLBO	
	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD
f1	1.5e-150	5.73e-150	-	1.11e-38	1.28e-128	-	1.89e-19	1.49e-19	-	0	0
f2	5.15e-84	1.44e-83	-	7.67e-22	5.63e-22	-	1.01e-13	6.54e-14	-	0	0
f3	1.1e-10	2.13e-10	-	NA	NA	NA	3.97e+02	1.42e+02	-	0	0
f4	0	0	=	NA	NA	NA	0	0	=	0	0
f5	5.8e-15	1.01e-14	-	0	0	=	2.57e-01	6.64e-11	-	0	0
f7	1.11e-14	3.55e-15	-	4.14e-05	0	-	2.01e-12	9.22e-13	-	0	0
f8	1.67e-02	2.41e-02	-	0	0	=	6.45e-13	2.07e-12	-	0	0
-	6			3			6			+ indicates better	
+	0			0			0			- indicates worse	
=	1			2			1			= indicates equivalent	

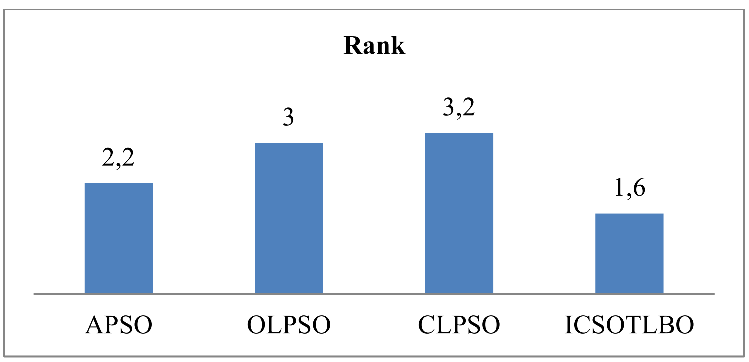

Fig.2.Comparison of Friedman ranks of ICSOTLBO with different variants of PSO for basic benchmark functions

6.3. Comparison of ICSOTLBO with BPSOGSA BGSA and GA

The performance of ICSOTLBO was compared with BPSOGSA, BGSA, and GA on the benchmark functions shown in Table 2. The results of BPSOGSA, BGSA, and GA were directly taken from the reference (Mirjalili et al. 2014). For fair comparison, the population size and number of function evaluations of ICSOTLBO were kept same as in the reference (Mirjalili et al. 2014). The mean and standard deviations of the errors are reported in Table 6 for each of the basic benchmark functions shown in Table 2. Further, Wilcoxon's rank sum test was conducted at 0.05 significance level between ICSOTLBO and each of BPSOGSA, BGSA, and GA. The results of the Wilcoxon's rank sum test are reported in the last three rows of Table 6. It was observed that ICSOTLBO was always better than BPSOGSA, BGSA, and GA.Further, for comparing the performance of the proposed algorithm with the BPSOGSA, BGSA, and GA, Friedman test was performed. The ranks of the different algorithms obtained by Friedman test is as shown in Fig.3. It was observed that ICSOTLBO had obtained the best rank in comparison to BPSOGSA, BGSA, and GA.

Table 6- Comparison of ICSOTLBO with BPSOGSA, BGSA, and GA(D=5, PN=30, FE=500)

Function	BPSOGSA			BGSA			GA			ICSOTLBO	
	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD
f1	0.753881836	0.744054218	-	2052.005	41.45277	-	10.0750	24.9445	-	3.1296e-61	1.3963e-60
f2	0.158447266	0.121911192	-	1.32569	0.67277	-	0.226948	0.23788	-	1.2360e-32	5.4264e-32
f3	45.2867	94.45	-	509.0988	266.3714	-	555.9039	250.693	-	1.8293e-71	8.1809e-71
f4	2.464062500	2.429516395	-	7.999	3.45	-	1.59375	1.21348	-	7.8294e-15	3.5014e-14
f5	1.875194	1.271683	-	5.999694	2.963102	-	2.1896	0.8330273	-	0.9950	0.757
f6	2.464062500	2.429516395	-	7.999	3.45	-	1.59375	1.21348	-	9.7504e-25	4.3473e-24
f7	0.541234	0.800463	-	2.947044	1.481999	-	1.399853	1.338105	-	0.1646	0.5067
f8	0.179551	0.092974	-	0.647846	0.228547	-	0.7067	0.3223	-	0.0666	0.0506
-	8			8			8			+ indicates better - indicates worse = indicates equivalent	
+	0			0			0				
=	0			0			0				

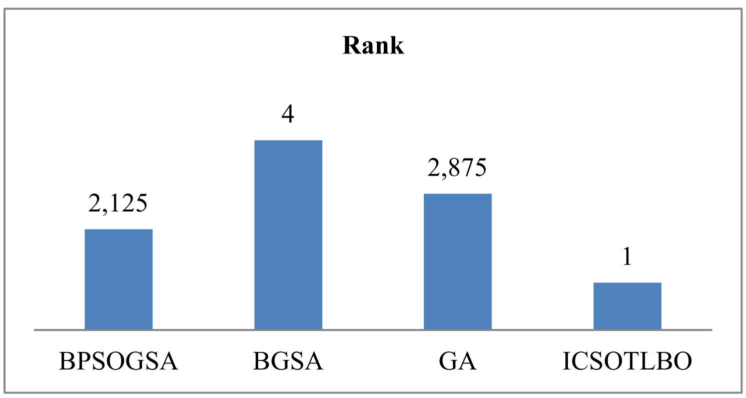

Fig.3.Comparison of Friedman ranks of ICSOTLBO with BPSOGSA, BGSA, and GA for basic benchmark functions

7. Performance of ICSOTLBO on computationally expensive benchmark functions

The proposed algorithm wasfurther tested on 15 computationally expensive benchmark functions as shown in Table 7.The benchmark functions reported in Table 7 were taken from the set of computationally expensive benchmark functions of various years of Congress on Evolutionary Competition (CEC). Most of the test functions reported in Table 7are complex functions representing shifted, rotated, expanded versions of basic benchmark functions. In Table 7, F1-F5 are unimodal functions, F6-F13 are multimodal functions and F14, F15 are hybrid functions. The detailed formulations of these benchmark functions can be found in the reference (Suganthan et al. 2005). The different algorithm specific parameters of ICSOTLBO were tuned as in Table 3. The performance of ICSOTLBO on these computationally expensive benchmark functions was compared with a number of state of art algorithms like different variants of PSO, DE, GA, TLBO and its variants etc. The results related to these comparisons are presented in the subsequent subsections.

Table 7- List of computationally expensive benchmark functions

Function no	Functionname	Bounds	f_bias
F1	Shifted sphere	[-100 100]	-450
F2	Shifted Schwefel's Problem 1.2	[-100 100]	-450
F3	Shifted Rotated High Conditioned Elliptic	[-100 100]	-450
F4	Shifted Schwefel's Problem 1.2 with Noise in Fitness	[-100 100]	-450
F5	Schwefel's Problem 2.6 with Global Optimum on Bounds	[-100 100]	310
F6	Shifted Rosenbrock's Function	[-100 100]	390
F7	Shifted Rotated Ackley's Function with Global Optimum on Bounds	[-32 32]	-140
F8	Shifted Rastrigin Function	[-5 5]	-330
F9	Shifted Rotated Rastrigin's Function	[-5 5]	-330
F10	Shifted Rotated Weierstrass Function	[0.5 -0.5]	90
F11	Schwefel's Problem 2.13	[-100 100]	-460
F12	Expanded Extended Griewank's plus Rosenbrock's Function	[-3 1]	-130
F13	Expanded Rotated Extended Scaffe's Function	[-100 100]	-300
F14	Hybrid Composition Function 1	[-5 5]	120
F15	Rotated Hybrid Composition Function 3	[-5 5]	360

7.1. Comparison of ICSOTLBO with TLBO and its variants

The performance of ICSOTLBO was compared with TLBO and mTLBO on the benchmark functions shown in Table 7. The results of TLBO were directly taken from the reference (Zhai et al. 2015; Rao &Waghmare 2013) and the results of mTLBO were directly taken from the reference (Satapathy & Naik 2014). For fair comparison, the population size and number of function evaluations of ICSOTLBO were kept same as in the references(Satapathy & Naik 2014; Zhai et al. 2015; Rao &Waghmare 2013).The mean and standard deviations of the errors are reported in Table 8 for each of the basic benchmark functions shown in Table 7. Further, Wilcoxon’s rank sum test was conducted at 0.05 significance level between ICSOTLBO and each of TLBO and mTLBO. The results of the Wilcoxon’s rank sum test are reported in the last three rows of Table 8. It was observed that ICSOTLBO performed better than TLBO on 9 benchmark functions, worse than TLBO on 5 benchmark functions and equivalent to TLBO on 1 benchmark function. And, ICSOTLBO performed better than mTLBO on 6 benchmark functions, worse than TLBO on 5 benchmark functions and equivalent to TLBO on 4 benchmark functions.For comparing the performance of the proposed algorithm with TLBO and its variants, Friedman test was also performed. The ranks of the different algorithms obtained by Friedman test is as shown in Fig.4. It was observed that ICSOTLBO was the second best performing algorithm in comparison to TLBO and mTLBO.

Table 8- Comparison of ICSOTLBO with TLBO and its variants (D=30, FE=3e+05)

Function	TLBO			mTLBO			ICSOTLBO	
	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD
F1	3.39 e-27	1.49 e-27	-	0.00e+00	0.00e+00	=	0.00e+00	0.00e+00
F2	1.56 e-09	4.20 e-09	+	1.79 e-08	3.46 e-08	-	1.717 e-08	3.41 e-08
F3	6.81 e+05	4.08 e+04	-	2.02 e+05	1.72 e+05	-	1.8507 e+05	6.2989e+04
F4	7.35 e+01	9.78 e+01	-	1.92 e+02	1.47 e+02	-	6.87e+01	5.98e+01
F5	3.16 e+03	6.77 e+02	-	4.21 e+03	1.13 e+03	-	1.7156e+03	5.19e+02
F6	5.36 e+01	4.12 e+01	-	1.82 e+01	5.79 e+00	-	9.5287e+00	5.4702e+00
F7	2.09 e+01	3.52 e-02	-	2.07 e+01	3.92e-02	=	2.07 e+01	3.92e-02
F8	8.59 e+01	1.92 e+01	+	6.34e+01	1.76e+01	+	3.02e+02	9.327e+01
F9	1.23e+02	3.30e+01	-	6.14e+01	6.13e+00	+	1.09e+02	2.6e+01
F10	3.09e+01	3.39e+00	+	3.15e+01	1.11e+00	+	3.7e+01	1.2747e+00
F11	9.93e+03	1.17e+04	+	1.67e+03	3.61e+03	+	3.864e+05	1.35e+05
F12	4.33e+00	9.27e-01	+	3.19e+00	3.4e-01	+	6.91e+01	4.4e+01
F13	1.29e+01	1.87e-01	-	1.20e+01	2.11e-01	=	1.20e+01	2.01e-01
F14	2.80e+02	7.48e+01	+	3.05e+02	6.46e+01	+	3.76e+02	5.04e+02
F15	5.002+02	1.92e+00	=	5.002+02	2.08e-13	=	5.002+02	0.00e+00
+	6			6			+ indicates better	
-	9			5			- indicates worse	
=	1			4			= indicates equivalent	

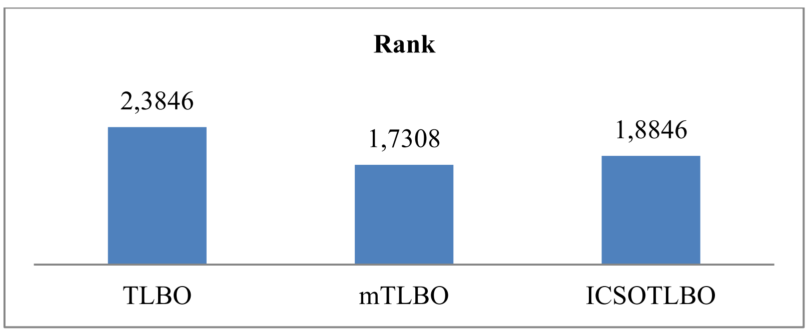

Fig.4-Comparison of Friedman ranks of ICSOTLBO with TLBO and its variants for unimodal and multimodal computationally expensive benchmark functions

7.2. Comparison of ICSOTLBO with different variants of PSO

The performance of ICSOTLBO was compared with APSO, OLPSO, and CLPSO on the benchmark functions shown in Table 7. The results of APSO, OLPSO, and CLPSO were directly taken from the reference (Li et al. 2015). For fair comparison, the population size and number of function evaluations of ICSOTLBO were kept same as in the reference (Li et al. 2015). The mean and standard deviations of the errors are reported in Table 9 for each of the benchmark functions shown in Table 7. Further, Wilcoxon's rank sum test was conducted at 0.05 significance level between ICSOTLBO and each of APSO, OLPSO, and CLPSO. The results of the Wilcoxon's rank sum test are reported in the last three rows of Table 9. ICSOTLBO performed better than APSO on 7 benchmark functions, worse than APSO on 7 benchmark functions and equivalent to APSO on 1 benchmark function. ICSOTLBO performed better than OPSO on 8 benchmark functions, worse than OPSO on 5 benchmark functions and equivalent to OPSO on 2 benchmark functions. And, ICSOTLBO performed better than CLPSO on 8 benchmark functions, worse than CLPSO on 6benchmark functions and equivalent to CLPSO on 1 benchmark function. For comparing the performance of the proposed algorithm with different variants of PSO, Friedman test was also performed. The ranks of the different algorithms obtained by Friedman test is as shown in Fig.5. It was observed that ICSOTLBO was the best performing algorithm in comparison to different variants of PSO.

Table 9- Comparison of ICSOTLBO with different variants of PSO (D=30, FE=3e+05)

Function	APSO			OLPSO			CLPSO			ICSOTLBO	
	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD
F1	7.01e-14	2.45e-14	-	0.00e+00	0.00e+00	=	5.68E-14	0.00e+00	-	0.00e+00	0.00e+00
F2	9.97e-13	1.79e-12	+	1.50e+01	1.23e+01	-	8.79e+02	1.79e+02	-	1.717 e-08	3.41 e-08
F3	3.96e+05	1.59e+05	-	1.46e+07	5.33e+06	-	1.67e+07	4.66e+06	-	1.8507 e+05	6.2989e+04
F4	7.23e+01	6.02e+01	-	2.26e+03	9.70e+02	-	6.61e+03	1.14e+03	-	6.87e+01	5.98e+01
F5	5.85e+03	1.45e+03	-	3.28e+03	5.54e+02	-	3.86e+03	5.32e+02	-	1.7156e+03	5.19e+02
F6	6.94e+00	1.68e+01	+	2.63e+01	2.50e+01	-	5.10e+00	5.43e+00	-	9.5287e+00	5.4702e+00
F7	2.07e+01	2.97e-02	=	2.09e+01	6.90e-02	-	2.09e+01	5.46e-02	-	2.07 e+01	3.92e-02
F8	1.48e-13	5.90e-14	+	0.00e+00	0.00e+00	+	1.08e-11	1.02e-11	+	3.02e+02	9.327e+01
F9	1.50e+02	6.25e+01	-	1.10e+02	3.12e+01	-	1.14e+02	1.50e+01	-	1.09e+02	2.6e+01
F10	2.78e+01	3.16e+00	+	2.55e+01	2.95e+00	+	2.7 e+01	1.71e+00	+	3.7e+01	1.2747e+00
F11	1.27e+04	1.70e+04	+	1.33e+04	6.95e+03	+	2.81e+04	6.59e+03	+	3.864e+05	1.35e+05
F12	1.54e+00	4.05e-01	+	1.92e+00	3.28e-01	+	1.66e+00	5.68e-01	+	6.91e+01	4.4e+01
F13	1.30e+01	5.24 e-01	-	1.31e+01	2.57 e-01	-	1.29e+01	1.72 e-01	-	1.20e+01	2.01e-01
F14	3.48e+02	1.50e+02	+	2.5e+02	9.21e+01	+	1.06e+02	5.34e+01	+	3.76e+02	5.04e+02
F15	7.66e+02	3.23e+02	-	5.002+02	2.86e-13	=	5.002+02	4.14e-13	=	5.002+02	0.00e+00
+	7			5			6			+ indicates better	
-	7			8			8			- indicates worse	
=	1			2			1			= indicates equivalent	

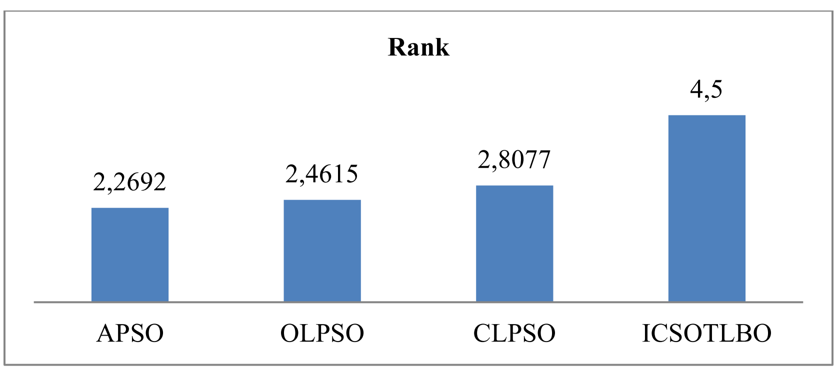

Fig.5-Comparison of Friedman ranks of ICSOTLBO with different variants of PSO for unimodal and multimodal computationally expensive benchmark functions

7.3. Comparison of ICSOTLBO with different variants of DE

The performance of ICSOTLBO was compared with SaDE, jDE and EPSDE on 15 benchmark functions shown in Table 7. The results of SaDE, jDE and EPSDE were directly taken from the reference (Satapathy & Naik 2014). For fair comparison, the population size and number of function evaluations of ICSOTLBO were kept same as in the reference (Satapathy & Naik 2014). The mean and standard deviations of the errors are reported in Table 10 for each of the basic benchmark functions shown in Table 7. Further, Wilcoxon's rank sum test was conducted at 0.05 significance level between ICSOTLBO and each of SaDE, jDE and EPSDE. The results of the Wilcoxon's rank sum test are reported in the last three rows of Table 10. ICSOTLBO performed better than SaDE on 5 benchmark functions, worse than SaDE on 7 benchmark functions and equivalent to SaDE on 3 benchmark functions. ICSOTLBO performed better than jDE on 8 benchmark functions, worse than jDE on 6 benchmark functions and equivalent to jDE on 1benchmark function. And, ICSOTLBO performed better than EPSDE on 6 benchmark functions, worse than EPSDE on 7 benchmark functions and equivalent to EPSDE on 2 benchmark functions. For comparing the performance of the proposed algorithm with different variants of DE, Friedman test was also performed. The ranks of the different algorithms obtained by Friedman test is as shown in Fig.6. It was observed that jDE was the best performing algorithm followed by SaDE. And, the rank of ICSOTLBO was equivalent to EPSDE.

Table 10- Comparison of ICSOTLBO with different variants of DE (D=30, FE=3e+05)

Function	jDE			SaDE			EPSDE			ICSOTLBO	
	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD
F1	0.00e+00	0.00e+00	=	0.00e+00	0.00e+00	=	0.00e+00	0.00e+00	=	0.00e+00	0.00e+00
F2	1.11e-06	1.10e-06	-	8.26e-06	1.65e-06	-	4.23e-26	4.07e-26	+	1.717 e-08	3.41 e-08
F3	1.98e+05	1.10e+05	-	4.27e+05	2.08e+05	-	8.74e+05	3.28e+06	-	1.8507 e+05	6.2989e+04
F4	4.40e-02	1.26e-01	+	1.77e+02	2.67e+02	-	3.49e+02	2.23e+03	-	6.87e+01	5.98e+01
F5	5.11e+02	4.40e+02	+	3.25e+03	5.90e+02	-	1.40e+03	7.12e+02	+	1.7156e+03	5.19e+02
F6	2.35e+01	2.50e+01	-	5.31e+01	3.25e+01	-	6.38e+01	1.49e+00	-	9.5287e+00	5.4702e+00
F7	2.09e+01	4.86e-02	-	2.09e+01	4.95e-02	-	2.09e+01	5.81e-02	-	2.07 e+01	3.92e-02
F8	0.00e+00	0.00e+00	+	2.39e-01	4.33e-01	+	3.98e-02	1.99e-01	+	3.02e+02	9.327e+01
F9	5.54e+01	8.46e+00	+	4.72e+01	1.01e+01	+	5.36e+01	3.03e+01	+	1.09e+02	2.6e+01
F10	2.79e+01	1.61e+00	+	1.65e+01	2.42e+00	+	3.76e+01	3.88e+00	=	3.7e+01	1.2747e+00
F11	8.63e+03	8.31e+03	+	3.02e+03	2.33e+03	+	3.58e+04	7.05e+03	+	3.864e+05	1.35e+05
F12	1.66e+00	1.35e-01	+	3.94e+00	2.81e-01	+	1.94e+00	1.46e_01	+	6.91e+01	4.4e+01
F13	1.30e+01	2.00e-01	-	1.26e+01	2.38e-01	-	1.35e+01	2.09e-01	-	1.20e+01	2.01e-01
F14	3.77e+02	8.02e+01	=	3.76e+02	7.83e+01	+	2.12e+02	1.98e+01	+	3.76e+02	5.04e+02
F15	5.002+02	4.80e-13	=	5.52e+02	1.82e+02	-	8.33e+02	1.00e+02	-	5.002+02	0.00e+00
+	7			6			7			+ indicates better	
-	5			8			6			- indicates worse	
=	3			1			2			= indicates equivalent	

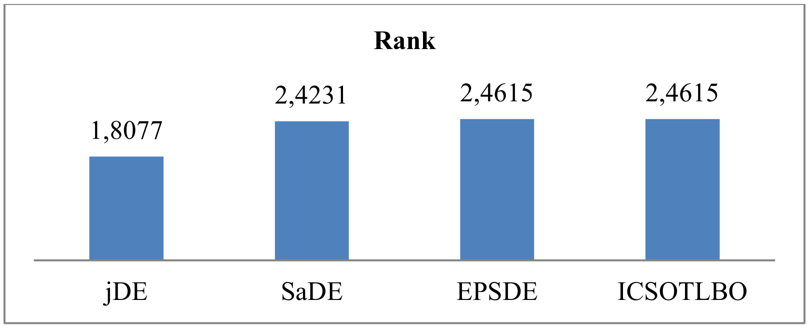

Fig.6-Comparison of Friedman ranks of ICSOTLBO with different variants of DE for unimodal and multimodal computationally expensive benchmark functions

7.4. Comparison of ICSOTLBO with SPC-PNX and CMA-ES

The performance of ICSOTLBO was compared with SPC-PNX and CMA-ES on 15benchmark functions shown in Table 7. The results of SPC-PNX and CMA-ES were directly taken from the references(Ballester et al. 2005; Satapathy & Naik 2014). For fair comparison, the population size and number of function evaluations of ICSOTLBO were kept same as in the reference (Ballester et al. 2005; Satapathy & Naik 2014). The mean and standard deviations of the errors are reported in Table 11 for each of the basic benchmark functions shown in Table 7. Further, Wilcoxon's rank sum test was conducted at 0.05 significance level between ICSOTLBO and each of SPC-PNX and CMA-ES. The results of the Wilcoxon's rank sum test are reported in the last three rows of Table 11. It was observed that ICSOTLBO performed better than SPC-PNX on 7 benchmark functions, worse than SPC-NX on 7 benchmark functions and equivalent to SPC-NX on 1 benchmark function. And, ICSOTLBO performed better than CMA-ES on 6 benchmark functions, worse than CMA-ES on 7 benchmark functions and equivalent to CMA-ES on 2 benchmark functions.For comparing the performance of the proposed algorithm with SPC-PNX and CMA-ES, Friedman test was also performed. The ranks of the different algorithms obtained by Friedman test is as shown in Fig.7. It was observed that CMA-ES was the best performing algorithm. And, the rank of ICSOTLBO was equivalent to SPC-PNX.

Table 11- Comparison of ICSOTLBO with SPC-PNX and CMA-ES (D=30, FE=3e+05)

Function	SPC-PNX			CMA-ES			ICSOTLBO	
	Mean	SD	+,-,=	Mean	SD	+,-,=	Mean	SD
F1	9.3524e-9	4.6327e-10	-	1.58e-25	3.35e-26	-	0.00e+00	0.00e+00
F2	6.9482e-7	1.4911e-6	-	1.12e-24	2.93e-25	+	1.717 e-08	3.41 e-08
F3	1.1020e+6	4.2081e+5	-	5.54e-21	1.69e-21	+	1.8507 e+05	6.2989e+04
F4	8.1320e-7	1.7457e-6	+	9.15e+05	2.16e+06	-	6.87e+01	5.98e+01
F5	4.2374e+3	1.3752e+3	-	2.77e-10	5.04e-11	+	1.7156e+03	5.19e+02
F6	1.5197e+1	1.4903e+1	-	4.78e-01	1.32e+00	+	9.5287e+00	5.4702e+00
F7	2.0932e+1	4.5876e-2	-	2.07e+01	5.72e-01	=	2.07 e+01	3.92e-02
F8	2.3934e+1	6.2477e+0	+	4.45e+02	7.12e+01	-	3.02e+02	9.327e+01
F9	6.0297e+1	4.0576e+1	+	4.63e+01	1.16e+01	+	1.09e+02	2.6e+01
F10	1 1255e+1	3.2979e+00	+	7.11e+01	2.14e+00	-	3.7e+01	1.2747e+00
F11	1.31e+04	1.3346e+04	+	1.26e+04	1.74e+04	+	3.864e+05	1.35e+05
F12	3.5881e+00	1.0857e+00	+	3.43e+00	7.60e-01	+	6.91e+01	4.4e+01
F13	1.3131e+1	2.6887e-1	-	1.47e+01	3.31e-01	-	1.20e+01	2.01e-01
F14	3.6822e+02	9.45e+01	+	5.55e+02	3.32e+02	-	3.76e+02	5.04e+02
F15	5.002e+02	0.00e+00	=	5.002+02	2.68e-12	=	5.002+02	0.00e+00
+	7			7			+ indicates better	
-	7			6			- indicates worse	
=	1			2			= indicates equivalent	

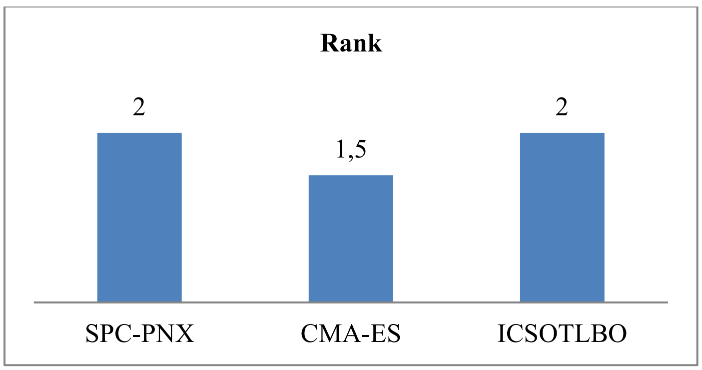

Fig.7-Comparison of Friedman ranks of ICSOTLBO with SPC-PNX and CMA-ES for unimodal and multimodal computationally expensive benchmark functions

8. Computational Complexity of ICSOTLBO

The complexity of the proposed ICSOTLBO algorithm was compared with other state of art algorithms like basic PSO, TLBO, GA, and DE. The details related to the evaluation criterion of computational complexity of algorithms used in the present work can be found in the reference (Suganthan et al. 2005). The proposed algorithms were tested in MATLAB 2016a software installed on a computer with the processor of 64 bit Intel i7 CPU. The results related to the computational complexity of the aforesaid algorithms are presented in Table 12. In Table 12, T0 represents the computing time of the base program given by CEC.T1 represents the computing time of F3 for 2e+05 function evaluations. T2 represents the mean of the computing time of F3 for 2e+05 function evaluations obtained by running the program 5 times. From Table 12, it can beobserved that the computational complexity of ICSOTLBO is less than

TLBO and GA but more than PSO and DE. It must be noted that the computational complexity of the algorithm increase with the increasing size of the data set. Thus, the computational complexity of all the algorithms may increase if we increase the value of D.

Table 11- Computational complexity of ICSOTLBO, PSO, TLBO and GA(D=30, FE=2e+05)

Algorithm	T0 (sec)	T1(sec)	T2(sec)	|(T2-T1)/T0|
ICSOTLBO	0.2081	61.7467	62.5996	4.098
PSO		29.89	29.87	0.0961
TLBO		55.45	59.98	21.768
GA		57.87	54.22	17.539
DE		32.76	32.54	1.057

9. Performance of ICSOTLBO on real-worldproblems

The performance of the proposed ICSOTLBO algorithm wasfurther tested on real- world problems like Charging Station Placement problem and Economic Load Dispatch problem. The performance of ICSOTLBO on these real-world problems is illustrated in this section.

9.1. Performance of ICSOTLBO on Charging Station Placement Problem

The performance of the proposed ICSOTLBO algorithm was appraised by applying it on solving the complex and demanding problem of charging station placement for Electric Vehicles (EVs). EVs are an environment friendly alternative to gasoline fuelled vehicles. However, the limited driving range is one of the drawbacks of EVs. The EVs need to recharge their batteries in the charging stations after travelling certain distance. These charging stations augment the load of the power grid(Deb et al. 2018a; Deb et al. 2019). Thus, the charging station placement must take into consideration security of the power distribution network as well as EV user's convenience. There are different formulations of charging station placement present in the existing literature (Deb et al. 2018b). In this work the charging station placement problem present in the reference (Deb et al. 2017) was solved by ICSOTLBO. The decision variables of the charging station placement problem were-

- Position where charging stations will be placed, b
- Number of fast charging stations placed at b, N_{Fb}
- Number of slow charging stations placed at b, N_{Sb}

It was considered that the charging stations would be placed at the superimposed nodes (TS) of the road and distribution network. Also, it was assumed that the charging stations would only be placed at the strong nodes (S) of the distribution network that were not prone to voltage instability.

The optimization aimed at minimization of the overall cost associated with charging stations. Moreover, the cost was divided into the direct and indirect cost. Direct cost considered the installation and operation cost associated with charging stations. Indirect cost considered the penalty paid by the utilities for violating the safe limits of distribution network parameters like voltage profile, reliability and the travelling distance cost from point of charging station to point of placement of charging station.

The objective function is

$$J = Min\ \ (C_{installation} + C_{operation} + C_{penalty} + C_{travel}) \tag{13}$$

where $C_{installation}$ represents installation cost of chargers, $C_{operation}$ represents operating cost of the charging stations, $C_{penalty}$ represents the penalty paid by utility for violating safe limits of voltage profile and Average Energy

Not Served (AENS), C_{travel} represents the travelling distance cost from point of charging station to point of placement of charging station.

Subject to

$$0 < N_{Fb} \leq n_{fastCS} \tag{14}$$

$$0 < N_{Sb} \leq n_{slowCS} \tag{15}$$

$$S_{\min} \leq S_i \leq S_{\max} \tag{16}$$

$$L \leq L_{\max} \tag{17}$$

where n_{fastCS} and n_{slowCS} represent the maximum number of fast and slow charging stations that can be placed, S_{min} and S_{max} represent lower and upper limit of reactive power flow of each bus, L_{max} represents the loading margin of the network.

Apart from the aforementioned constraints the power balance equation must also be considered as an equality constraint while solving the charging station placement problem (Deb et al. 2017).

The charging station placement problem was solved for superimposed IEEE 33 bus distribution network and 25 node road network. The details of the test system and the input parameters of the charging station placement problem can be found in the reference (Deb et al. 2017).

The performance of ICSOTLBO on solving charging station placement problem was also compared with other state of art algorithms like GA, DE, PSO, CSO, TLBO, and CSOTLBO. The different algorithm specific parameters of the aforesaid algorithms are listed in Table 13. For fair comparison the population and iteration are fixed to 10 and 50 respectively for all the aforesaid algorithms. The optimal value of the decision variables for minimization of the overall cost and the best value of the fitness function is as reported in Table 14. It was observed that the best fitness value obtained by ICSOTLBO was 1.3605 that was better than CSOTLBO, TLBO, CSO, GA, DE, and PSO. A statistical comparison of the quality of solution was performed for all the algorithms, the results of which are reported in Table 15.The results reported in Table 15 demonstrate the superiority of ICSOTLBO over CSO, TLBO, CSOTLBO, GA, PSO and DE in solving the complex charging station placement problem. The convergence curve of ICSOTLBO and the aforesaid algorithms for the best fitness value is as shown in Fig.8.

Table 13-Algorithm specific parameters of different state of art algorithms for charging station placement problem

Algorithm	Parameters
PSO	c1=c2=2, w=0.1
DE	CR=0.6, F=1.5
CSO	RN=0.2PN, HN=0.5PN, CN=PN-RN-HN, MN=0.3PN, G=5
CSO TLBO	RN=0.3PN, HN=0.4PN, CN=PN-RN-HN, MN=0.3PN, G=3, INV=5
ICSOTLBO	RN=0.3PN, HN=0.4PN, CN=PN-RN-HN, MN=0.3PN, G=3, INV=5, C=2, S=2

Table 14-Optimal placement of charging stations by ICSOTLBO and other state of art algorithms

Optimization technique	Fitness value (best)	b	N_{Fb}	N_{Sb}
ICSOTLBO	1.3605	6	1	2
		3	1	2
		23	1	3
CSOTLBO	1.4841	6	1	2
		3	1	3
		23	1	3
CSO	1.4870	6	1	3
		23	1	3
		3	1	2
TLBO	1.4878	3	1	3
		23	1	3
		28	1	2
PSO	1.4898	23	1	2
		6	1	3
		3	1	3
DE	1.4898	23	1	2
		6	1	3
		3	1	3
GA	1.5075	23	1	2
		3	1	3
		28	1	3

Table 15-Statistical comparison of ICSOTLBO with other algorithms in solving Charging Station Placement problem

Algorithm	Mean fitness
ICSOTLBO	1.4268
CSOTLBO	1.5241
CSO	1.5430
TLBO	1.5413
PSO	1.5413
DE	1.5497
GA	1.5584

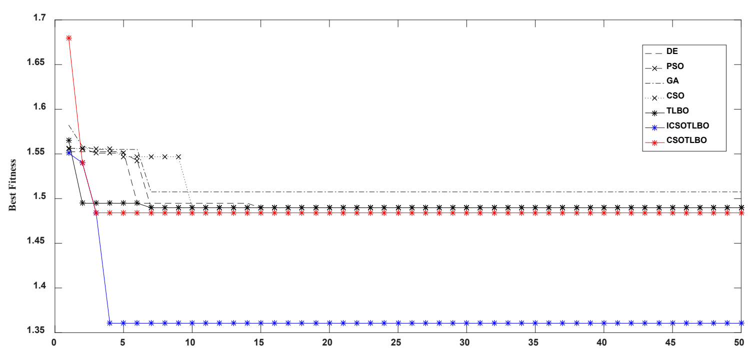

Fig.8-Convergence curve of different algorithms for the best fitness value of Charging Station Placement problem

9.2. Performance of ICSOTLBO on Economic Load Dispatch Problem

Economic Load Dispatch is considered as one of the complex power system optimization problems. The main objective of Economic Load Dispatch is to minimize the net cost of generation under a set of operating constraints.

Both convex and non convex formulations of Economic Load Dispatch problem are available in the existing literature (Bhattacharjee et al. 2014a; Bhattacharjee et al. 2014b; Bhattacharjee et al. 2014c). In the present work, Economic Load Dispatch problem with quadratic fuel cost function along with operating limits was solved by ICSOTLBO. The objective function is expressed as-

$$J = Min\sum_{i=1}^{N}(a_i + b_i P_i + c_i P_i^2) \tag{18}$$

where N is the total number of generators in the system, a_i, b_i , c_i are the cost coefficients of the i^{th} generator, P_i is the output power of i^{th} generator.

Subject to-

$$\sum_{i=1}^{N} P_i - P_D = 0 \tag{19}$$

$$P_i^{\min} \le P_i \le P_i^{\max} \tag{20}$$

where P_D represents net power demand of the system, P_i^{min} and P_i^{max} represent lower and upper power limit of i^{th} generator.

The Economic Load Dispatch problem with quadratic cost function and operating constraints elaborated by Eq. (18) - Eq. (20) was solved by ICSOTLBO. The test system considered was a 38 generator test system. The details of the test system and the input parameters were same as reference (Bhattacharjee et al. 2014a; Bhattacharjee et al. 2014b; Bhattacharjee et al. 2014c). The performance of ICSOTLBO algorithm in solving Economic Load Dispatch problem was compared with other algorithms like RCCRO, TLBO, and DE. The results of RRCO, DE were taken from (Bhattacharjee et al. 2014c) and the results of TLBO were taken from (Bhattacharjee et al. 2014b). For fair comparison, the population size and number of function evaluations of ICSOTLBO were kept same as in the reference (Bhattacharjee et al. 2014b; Bhattacharjee et al. 2014c).The different algorithm specific parameters of ICSOTLBO were same as in Table 13 only the value of *INV* is changed to 10.The mean of the cost function over 50 trials obtained by the aforesaid algorithms are reported in Table 16.The results reported in Table 16 demonstrate the superiority of ICSOTLBO over TLBO, RRCRO and DE in solving the Economic Load Dispatch problem. The convergence curve of different algorithms for the best fitness value is as shown in Fig.9.

Table 16- Statistical comparison of ICSOTLBO with other algorithms in solving Economic Load Dispatch problem

Algorithm	Mean Fitness ($/hr)
TLBO	9411938.55723
RCCRO	9412404.277425
DE	9417237.290970
ICSOTLBO	9411938.54700

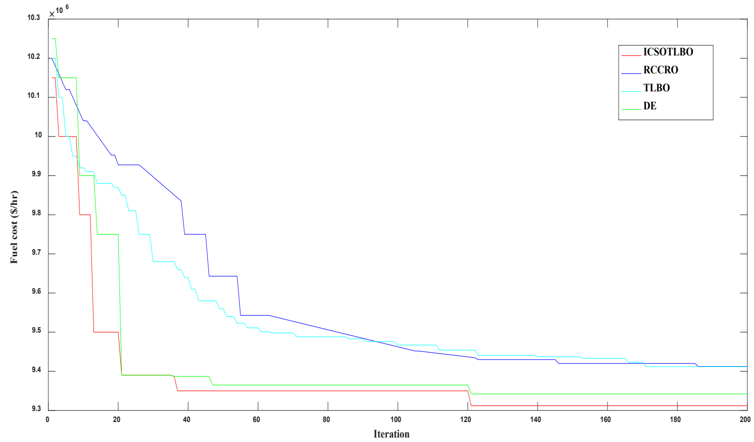

Fig.9-Convergence curve of different algorithms for the best fitness value of Economic Load Dispatch problem

10. Future Directions of Work

The work proposes improvement of basic CSO algorithm and its hybridization with TLBO. The proposed algorithm performs satisfactorily on basic, computationally expensive benchmark problems as well as real-world problems. However, still there is scope of improving the algorithm and using the algorithm for complex optimization problems. Future works that can be undertaken on the proposed algorithm are listed below:

1. Development of adaptive CSO- CSO has a number of algorithm-specific parameters. Improper tuning of these parameters sometimes causes slow convergence of the algorithm. Also, the tuning of these parameters is done by trial and error method that is very much time consuming. Hence, development of an adaptive version of CSO is a promising area of research.

2. Solution of complex real-world optimization problems by ICSOTLBO-There are many complex real-world optimization problems that are difficult to solve by conventional algorithms. The proposed ICSOTLBO algorithm can be utilized to solve real-world complex problems such as optimization of vehicle-to-vehicle frontal crash model (Bououden et al. 2017), predictive control of non-linear processes (Bououden et al. 2015), microgrid control (Goodarzi and Kazemi 2017), optimal configuration of microgrid (Deb et al. 2016; Ghosh et al. 2017).

3. Improvement of the algorithm with information feedback models-The proposed ICSOTLBO algorithm does not fully utilize the information available from previous iterations. If the information from the previous iterations can be utilized properly then it is expected that the quality of the solutions will significantly improve. Thus, introduction of the feedback models suggested by Wang et al. 2019 in the proposed ICSOTLBO algorithm is a promising area of research.

4. Hybridization of CSO with other Nature Inspired algorithms-Standalone Nature Inspired Optimization (NIO) algorithms are sometimes not efficient enough to handle the uncertainty of the practical optimization problems. Hybridization of NIO algorithms offers competitive solutions than standalone NIO algorithms in case of practical problems. Also, the hybrid algorithms inherit the advantages of two standalone algorithms, eliminate the limitations of the standalone algorithms, and perform better than the standalone algorithms. A good balance between exploration and exploitation is maintained in the hybrid algorithms.There is scope of hybridizing CSO with other

NIO algorithms. Hybridization of CSO with other NIO algorithms such as Jaya algorithm, Sine Cosine algorithm is a promising area of research.

11. Conclusions

The work proposes improvement of basic CSO algorithm and its hybridization with TLBO. The performance of the proposed algorithm is investigated on basic benchmark functions as well as computationally expensive functions. It is observed that the proposed algorithm outperforms many of the state of art algorithms like PSO, DE, GA. Further, the proposed algorithm is used for solving the complex problem of Charging Station Placement and Economic Load Dispatch problem. The superiority of the proposed algorithm in solving complex real-world problems likeCharging Station Placement and Economic Load Dispatch problem is also clearly revealed in the work. Our future work will focus on further improvement of CSO, development of adaptive CSO, hybridization of CSO with other evolutionary algorithms, solution of complex power system optimization problems by CSO.

Acknowledgements

Xiao-Zhi Gao's research work was partially supported by the National Natural Science Foundation of China (NSFC) under Grant 51875113.

Compliance with ethical standards

Conflict of interest

We have no conflict of interest with this research article

Human and animal rights

We use no animal in this research

REFERENCES

Ahmed, K., Hassanien, A. E., & Bhattacharyya, S. (2017, November).A novel chaotic chicken swarm optimization algorithm for feature selection. In *Research in Computational Intelligence and Communication Networks (ICRCICN), 2017 Third International Conference on* (pp. 259-264). IEEE.

Ballester, P. J., Stephenson, J., Carter, J. N., & Gallagher, K. (2005, September).Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX.In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*(Vol. 1, pp. 498-505). IEEE.

Bhattacharjee, K., Bhattacharya, A., & nee Dey, S. H. (2014). Oppositional real coded chemical reaction optimization for different economic dispatch problems. *International Journal of Electrical Power & Energy Systems*, *55*, 378-391.

Bhattacharjee, K., Bhattacharya, A., &Dey, S. H. N. (2014).Teaching-learning-based optimization for different economic dispatch problems. *Scientia Iranica. Transaction D, Computer Science & Engineering, Electrical*, *21*(3), 870.

Bhattacharjee, K., Bhattacharya, A., & nee Dey, S. H. (2014).Chemical reaction optimisation for different economic dispatch problems. *IET Generation, Transmission & Distribution*, *8*(3), 530-541.

Cai, X., Gao, X. Z., & Xue, Y. (2016). Improved bat algorithm with optimal forage strategy and random disturbance strategy. *International Journal of Bio-Inspired Computation*, *8*(4), 205-214.

Chen, Y. L., He, P. L., & Zhang, Y. H. (2015). Combining penalty function with modified chicken swarm optimization for constrained optimization.*Advances in Intelligent Systems Research*, *126*, 1899-1907.

Chen, S., Yang, R., Yang, R., Yang, L., Yang, X., Xu, C., & Liu, W. (2016). A Parameter Estimation Method for Nonlinear Systems Based on Improved Boundary Chicken Swarm Optimization. *Discrete Dynamics in Nature and Society*, *2016*.

Deb, S., Gao, X. Z., Tammi, K., Kalita, K., & Mahanta, P. (2019). Recent Studies on Chicken Swarm Optimization algorithm: a review (2014–2018). *Artificial Intelligence Review*, 1-29.

Deb, S., Tammi, K., Kalita, K., & Mahanta, P. (2018a).Impact of Electric Vehicle Charging Station Load on Distribution Network. *Energies*, *11*(1), 178.

Deb S, Tammi K, Kalita K, Mahanta P.(2018b) Review of recent trends in charging infrastructure planning for electric vehicles. *WIREs Energy Environment*. 2018;e306. https://doi.org/10.1002/wene.306

Deb, S., Kalita, K., Gao, X. Z., Tammi, K., & Mahanta, P. (2017, November). Optimal placement of charging stations using CSO-TLBO algorithm. In *Research in Computational Intelligence and Communication Networks (ICRCICN), 2017 Third International Conference on* (pp. 84-89). IEEE.

Deb, S., Ghosh, D., & Mohanta, D. K. (2016, October).Optimal configuration of stand-alone hybrid microgrid considering cost, reliability and environmental factors. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*(pp. 48-53). IEEE.

Deb, S., Kalita, K., &Mahanta, P. (2019).Distribution Network planning considering the impact of Electric Vehicle charging station load.In *Smart Power Distribution Systems* (pp. 529-553).Academic Press.

Faris, H., Aljarah, I., Al-Betar, M. A., &Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications*, 1-23.

Gao, X. Z., Govindasamy, V., Xu, H., Wang, X., & Zenger, K. (2015). Harmony search method: theory and applications. *Computational intelligence and neuroscience*, *2015*, 39.

Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, *3*(2), 95-99.

Ghosh, D., Deb, S., &Mohanta, D. K. (2017). Reliability Evaluation and Enhancement of Microgrid Incorporating the Effect of Distributed Generation.In *Handbook of Distributed Generation* (pp. 685-730).Springer, Cham.

Han, M., & Liu, S. (2017, December).An Improved Binary Chicken Swarm Optimization Algorithm for Solving 0-1 Knapsack Problem. In *Computational Intelligence and Security (CIS), 2017 13th International Conference on* (pp. 207-210). IEEE.

Karaboga, D., &Basturk, B. (2008).On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing*, *8*(1), 687-697.

Kumar, D.S &Veni,S. "Enhanced Energy Steady Clustering UsingConvergence Node Based Path Optimizationwith Hybrid Chicken Swarm Algorithm inMANET.*International Journal of Pure and Applied Mathematic,*118,767-788.

Li, Y. F., Zhan, Z. H., Lin, Y., & Zhang, J. (2015, May). Comparisons study of APSO OLPSO and CLPSO on CEC2005 and CEC2014 test suits. In *Evolutionary Computation (CEC), 2015 IEEE Congress on* (pp. 3179-3185).IEEE.

Liang, S., Feng, T., Sun, G., Zhang, J., & Zhang, H. (2016, October). Transmission power optimization for reducing sidelobe via bat-chicken swarm optimization in distributed collaborative beamforming. In *Computer and Communications (ICCC), 2016 2nd IEEE International Conference on* (pp. 2164-2168). IEEE.

Meng, X.B., Gao, X. Z., Lu, L., Liu, Y., & Zhang, H. (2016). A new bio-inspired optimisation algorithm: Bird Swarm Algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, *28*(4), 673-687.

Meng, X.B., Liu, Y., Gao, X., & Zhang, H. (2014, October). A new bio-inspired algorithm: chicken swarm optimization. In *International conference in swarm intelligence* (pp. 86-94).Springer, Cham.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61.

Mirjalili, S., & Lewis, A. (2016).The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51-67.

Poli, R., Kennedy, J., & Blackwell, T. (2007).Particle swarm optimization. *Swarm intelligence*, *1*(1), 33-57.

Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, *96*, 120-133.

Mirjalili, S., Wang, G. G., & Coelho, L. D. S. (2014).Binary optimization using hybrid particle swarm optimization and gravitational search algorithm.*Neural Computing and Applications*, *25*(6), 1423-1435.

Rashedi, E., Nezamabadi-Pour, H., &Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences*, *179*(13), 2232-2248.

Rao, R. (2016). Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decision science letters*, *5*(1), 1-30.

Rao, R. V., & Kalyankar, V. D. (2011). Parameters Optimization of Advanced Machining Processes using TLBO Algorithm, vol. 20.*EPPM, Singapore*.

Rao, R. V., & Waghmare, G. G. (2013).Solving composite test functions using teaching-learning-based optimization algorithm. In *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)* (pp. 395-403). Springer, Berlin, Heidelberg.

Satapathy, S. C., &Naik, A. (2014). Modified Teaching–Learning-Based Optimization algorithm for global numerical optimization—A comparative study. *Swarm and Evolutionary Computation*, *16*, 28-37.

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization.*KanGAL report*, *2005005*, 2005.

Torabi, S., & Safi-Esfahani, F. (2018). A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing. *The Journal of Supercomputing*, 1-46.

Van Laarhoven, P. J., &Aarts, E. H. (1987).Simulated annealing. In *Simulated annealing: Theory and applications*(pp. 7-15). Springer, Dordrecht.

Wang, G. G., Deb, S., Gao, X. Z., & Coelho, L. D. S. (2016). A new metaheuristic optimisation algorithm motivated by elephant herding behaviour. *International Journal of Bio-Inspired Computation*, *8*(6), 394-409.

Wang, G. G., Deb, S., & Coelho, L. D. S. (2015, December). Elephant herding optimization. In *Computational and Business Intelligence (ISCBI), 2015 3rd International Symposium on*(pp. 1-5). IEEE.

Wang, G. G., Deb, S., & Coelho, L. D. S. (2015). Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *International Journal of Bio-Inspired Computation*, *7*, 1-23.

Wang, K., Li, Z., Cheng, H., & Zhang, K. (2017, December). Mutation chicken swarm optimization based on nonlinear inertia weight. In *Computer and Communications (ICCC), 2017 3rd IEEE International Conference on* (pp. 2206-2211). IEEE.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67-82.

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.

Yang, X. S., & Deb, S. (2014). Cuckoo search: recent advances and applications. *Neural Computing and Applications*, *24*(1), 169-174.

Zhai, Z., Li, S., Liu, Y., & Li, Z. (2015). Teaching-learning-based optimization with a fuzzy grouping learning strategy for global numerical optimization. *Journal of Intelligent & Fuzzy Systems*, *29*(6), 2345-2356.