Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

An ensemble symbiosis organisms search algorithm and its application to real world problems

Sukanta Nama[*] and Apu Kumar Saha

Department of Mathematics, National Institute of Technology Agartala, Barjala, Jirania-799046, West Tripura, India

C H R O N I C L E

Article history:
Received January 16, 2017
Received in revised format: May 22, 2017
Accepted June 23, 2017
Available online June 25, 2017

Keywords:
Symbiosis organisms search (SOS) algorithm
Quasi-opposition based learning (QOBL)
Ensemble algorithm
Unconstrained global optimization problem

A B S T R A C T

In this study, an ensemble algorithm has been proposed, called Quasi-Oppositional Symbiosis Organisms Search (QOSOS) algorithms, by incorporating the quasi-oppositional based learning (QOBL) strategy into the newly proposed Symbiosis Organisms Search (SOS) algorithm for solving unconstrained global optimization problems. The QOBL is incorporated into the basic SOS algorithm due to the balance of the exploration capability of QOBL and the exploitation potential of SOS algorithm. To validate the efficiency and robustness of the proposed Quasi-Oppositional Symbiosis Organisms Search (QOSOS) algorithms, it is applied to solve unconstrained global optimization problems. Also, the proposed QOSOS algorithm is applied to solve two real world global optimization problems. One is gas transmission compressor design optimization problem and another is optimal capacity of the gas production facilities optimization problem. The performance of the QOSOS algorithm is extensively evaluated and compares favorably with many progressive algorithms.

1. Introduction

In solving the non-linear problem, the optimization algorithm is most significant and sturdy. Since some of the problems have discontinuous and gradient based algorithms are not suitable for that sort of problem (Lee & Geem, 2004; Osman & Laporte, 1996), to overcome this difficulity, recently, several optimization algorithms have been introduced within the past decade. Some of them are given in the references (Holland, 1975; Kennedy & Eberhart, 1995; Storn & Price, 1997; Dorigo & Stützle, 2004; Geem et al., 2001; Karaboga & Basturk, 2007; Yang & Deb, 2009; Sadollah et al., 2012; Eskandar et al., 2012; Pham et al., 2006; Yang, 2009; Nama et al., 2015a, 2016b, 2017). The idea of Opposition-based learning (OBL) was initially developed by Tizhoosh (Tizhoosh, 2005). The inventor of OBL claims that a number's opposite is probably nearer than a random number to a solution. Integrating OBL in conventional evolutionary algorithms, one can increase the coverage of the solution space leading to increases accuracy and faster convergence. Some of the applications of OBL in the soft

* Corresponding author.
E-mail address: sukanta1122@gmail.com (S. Nama)

doi: 10.5267/j.dsl.2017.6.006

computing field include oppositional particle swarm optimization (OPSO) (Wang et al., 2011), oppositional biogeography based optimization (OBBO) (Roy and Mandal 2014), oppositional differential evolution (ODE) (Qingzheng et al. 2011). The idea of QOBL is employed in population initialization and generation jumping. A number of the application of QOBL within the soft computing field includes Quasi-oppositional TLBO (QOTLBO) (Mandal & Roy, 2013), quasi-oppositional BBO (QOBBO) (Roy & Mandal, 2011) and Multi-objective quasi-oppositional TLBO (MOQOTLBO) (Sultana & Roy, 2014). In this paper, quasi-opposition based learning (QOBL) strategy is introduced in the basic SOS algorithm which provides the additional likelihood of finding solutions nearer to the global optimum. In the proposed QOSOS algorithm, first, all the steps of SOS algorithm executed and then the concept of quasi-opposition based jumping is executed based on the jumping rate.

Therefore, the main contributions of this study are summarized as follows:

i) A new ensemble algorithm (called QOSOS) is proposed by the combination of QOBL strategy and SOS algorithm.
ii) The proposed QOSOS algorithm have been applied to solve some well known benchmark functions.
iii) Finally, the proposed QOSOS algorithm has been applied to solve two real world optimization problems.

The remaining part of the paper is organized as follows: Section 2 discusses the overview of the basic Symbiosis Organisms Search algorithm. The Quasi-opposition-based learning (QOBL) strategy is presented in Section 3. Section 4 presents the details description of the newly proposed Quasi-Oppositional Symbiosis Organisms Search (QOSOS) algorithm. Section 5 and 6 demonstrate the efficiency and accuracy of the QOSOS algorithm for solving some well known unconstrained optimization problems and two real world optimization problems. Finally, Section 7 presents the summary of the contribution of this paper along with some future research directions.

2. Symbiosis Organism Search Algorithm

The Symbiosis Organisms Search algorithm, proposed by Cheng and Prayogo (2014), simulates the interactive behavior seen among organisms in nature. Symbiosis is derived from the Greek word for ‘‘living together’’. De Bary first used the term in 1878 to explain the cohabitation behavior of unlike organisms (Sapp 1994). Today, symbiosis is used to describe a relationship between any two distinct species. Symbiotic relationships may be either obligate, meaning the two organisms depend on each other for survival or facultative, i.e., the two organisms choose to cohabitate in a mutually beneficial but non-essential relationship.

In an ecosystem, the most common symbiotic relationships are mutualism, commensalism, and parasitism. Mutualism relationship may be between two or more organisms and in this symbiosis relationship, both organisms get benefit from each other. In Commensalism relationships, one organism is benefited from the interaction and the other is not harmed or helped. A parasitism relationship is a symbiosis relationship in which one organism gets the benefit and the other is harmed but not always to its death.

SOS begins with an initial population referred to as the ecosystem. In the initial ecosystem, a bunch of organisms is generated randomly within the search area. Each organism represents one candidate solution to the corresponding problem. Every organism within the ecosystem is associated with an explicit fitness value that reflects the degree of adaptation to the required objective. In SOS, new solution generation is governed by imitating the biological interaction between two organisms in the ecosystem. The detailed description of SOS algorithm can be seen in Cheng and Prayogo (2014). Also some applications and improve vesion of SOS algorithm are presented in the literature (Nama et al., 20016c, 2016d; Prasad & Mukherjee, 2015; Abdullahi & Ngadi, 2016; Cheng et al., 2015; Verma et al., 2015).

3. Quasi-oppositional strategy

3.1. Quasi-opposition-based learning (QOBL)

We recall some definitions related to Quasi-opposition-based learning (QOBL) for the convenience of the proposed algorithm.

3.1.1. Definition (Tizhoosh, 2005)

Let $x \in [a,b]$ be a real number. The opposite number x^o of x is defined by $x^o = a + b - x$.

3.1.2. Definition (Tizhoosh, 2005)

Let $P = (x_1, x_2,, \dots \dots \dots.., x_D)$ be a point in D-dimensional space, where $x_1, x_2,, \dots \dots \dots.., x_D \in R$ and $x_i \in [a, b] \; \forall \, i \in \{1, 2, 3, \dots.., D\}$. The oppositional point $OP = (x_1^o, x_2^o,, \dots \dots.., x_D^o)$ of P is defined by $x_i^o = a + b - x_i$.

3.1.3. Definition (Roy & Mandal, 2011)

Let x be any real number between (a, b). Its quasi-opposite number x_{qo} is defined as $x_{qo} = rand(c, x^o)$ where *c* is given by: $c = \frac{a+b}{2}$.

3.1.4. Definition (Roy & Mandal, 2011)

Let x be any real number between (a, b). Then quasi-opposite point x_i^{qo} is defined as $x_i^{qo} = rand(c_i, x_i^o)$ where $c_i = \frac{a_i+b_i}{2}$, $x_i^o \in [a_i, b_i]; i \in \{1, 2, 3, \dots \dots.., D\}$.

3.2. Quasi-opposition based optimization

The quasi-oppositional based optimization is based on quasi-opposition based initialization and quasi-opposition based generation jumping which are described below:

3.2.1. Quasi-opposition initialization

By utilizing opposite points, better initial candidate solutions, namely opposite population (OP) may be obtained even when there is no prior knowledge about the solutions. Initialization of quasi-opposite population (QOP) may be described as follows (Roy & Mandal, 2011):

For i = 1: eco-size (Number of organism in the ecosystem)
    For j = 1: D (Dimension of the variable)
        $OP_{i,j} = a_j + b_j - x_{i,j}$;
        $c_j = \frac{a_j+b_j}{2}$;
        $x_{i,j}^{qo} = rand(c_j, OP_{i,j})$;
    End
End

3.3.2. Quasi-opposition generation jumping

By applying a similar approach to the current population, the evolutionary process can be forced to jump to a new solution candidate, which may be better than the current one. Based on a jumping rate Jr, after generating new population by using all the steps of SOS, the opposite population is generated. Quasi-Opposite population jumping based on jumping rate is described as follows:

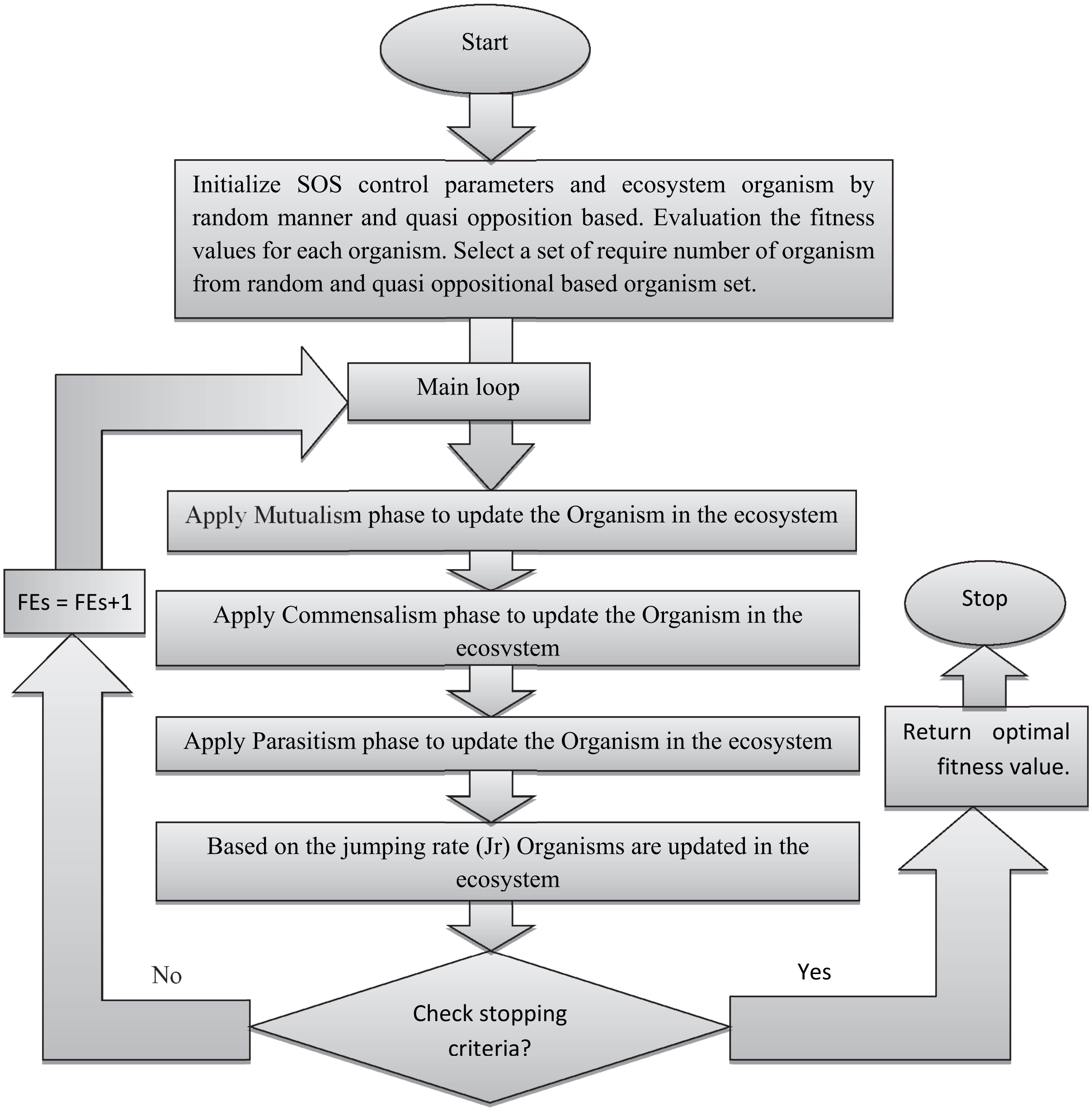


Fig. 1. Flowchart of the proposed QOSOS algorithm.

If rand (0, 1) < Jr (Jumping rate)
  For i = 1: eco-size (Number of organism in the ecosystem)
    For j = 1: D (Dimension of the variable)
      $OP_{i,j} = a_j + b_j - x_{i,j}$;
      $c_j = \frac{a_j+b_j}{2}$;
      $x_{i,j}^{qo} = rand(c_j, OP_{i,j})$;
    End
  End
End

In this paper jumping rate (Jr) is chosen as 0.3.

4. The Procedure of Proposed QOSOS Algorithm

In the development of heuristic global optimization algorithm, the balance between exploration and exploitation capability plays a major role (Crepinšek et al., 2013). According to Crepinšek et al. (2013) "Exploration is the process of visiting entirely new regions of a search space, whilst exploitation is the process of visiting those regions of a search space within the neighborhood of previously visited points". As discussed above the QOP method may be used for better exploration when executing the optimization process, on the other hand, Cheng and Prayoga (2014) elaborately discussed the better exploitation ability of SOS algorithm for global optimization. By incorporating the exploration capability of QOP into the SOS algorithm, the quasi oppositional symbiosis organisms search (QOSOS) algorithm has been proposed. The QOSOS is able to explore the new search region with the SOS algorithm and to exploit the population information with the QOP. This ensemble method can increase the robustness of the algorithm as well as searching capability of the algorithm for attaining the global optimum. During the execution of the optimization process of the proposed QOSOS, two variants of organisms are evaluated with the same size. One is randomly initialized and another is quasi-oppositional based initialize organisms. From these, eco-size number organisms have been selected as initial organisms. If an organism violates boundary constraints, the organism is reflected back inside the boundary using the following rule (Gong et al., 2006):

$$\mathrm{Org_i} = \begin{cases} \mathrm{l_i} + \mathrm{rand}(0,1) * (\mathrm{u_i} - \mathrm{l_i}) & \text{if } \mathrm{Org_i} < \mathrm{l_i} \\ \mathrm{u_i} - \mathrm{rand}(0,1) * (\mathrm{u_i} - \mathrm{l_i}) & \text{if } \mathrm{Org_i} > \mathrm{u_i} \end{cases} \tag{1}$$

where l_i *and* u_i are the lower and upper bound of the ith organism.

The flowchart of the proposed QOSOS algorithm is given in Fig.1 and the procedure for implementing the QOSOS algorithm can be summarized in the following steps:
Step 1: Initialize the algorithm parameter.
Step 2: Randomly generate the organisms and evaluate the fitness value for each organism.
Step 3: Generate quasi-oppositional population i.e. organism (QOP) set and evaluates the fitness value using the procedure given in Section 3.2.1.
Step 4: Select eco-size number of fittest organisms from {Org, QOP} as initial organisms set. Org is the set of the organism. Eco-size is the number of the organism in the ecosystem.

Step 5: Main Loop

Step 5.1: Update organisms by mutualism phase:

In this phase, an organism $\mathrm{Org_j}$ is selected randomly from the ecosystem to interact with organism $\mathrm{Org_i}$, the ith member in the ecosystem. Both organisms engage in a mutualistic relationship with the goal of increasing mutual survival advantage in the ecosystem. The new organism in the ecosystem for $\mathrm{Org_i}$ and $\mathrm{Org_j}$ are calculated based on the mutualistic symbiosis between organism $\mathrm{Org_i}$ and $\mathrm{Org_j}$, by Eqs. (2-3).

$$\mathrm{Org_i^{new}} = \mathrm{Org_i} + \mathrm{rand}(0,1) \times (\mathrm{Org_{best}} - \mathrm{Mutual_{Vector}} \times \mathrm{BF1}) \tag{2}$$

$$\mathrm{Org_j^{new}} = \mathrm{Org_j} + \mathrm{rand}(0,1) \times (\mathrm{Org_{best}} - \mathrm{Murual_{Vector}} \times \mathrm{BF2}) \tag{3}$$

where,

$$\mathrm{Mutual_{Vector}} = \frac{\mathrm{Org_i} + \mathrm{Org_j}}{2} \tag{4}$$

Here, benefit factors (BF1 and BF2) are determined randomly as either 1 or 2. These factors represent the level of benefit to each organism, i.e., whether an organism partially or fully benefits from the interaction. Eq. (4) shows a vector called ‘‘Mutual_Vector’’ which represents the relationship characteristic between organism Org_i and Org_j. Org_{best} is the best organism in the ecosystem.

Step 5.2: Update organisms by commensalism phase:

In the commensalism phase, organism Org_i benefits by organism Org_j and organism Org_i increased the beneficial advantage in the ecosystem to the higher degree of adaption. New organism Org_i is calculated by Eq.(5) and updated in the ecosystem only if its new fitness is better than its pre-interaction fitness.

$$Org_i^{new} = Org_i + rand(-1,1) * (Org_{best} - Org_j) \quad (5)$$

Step 5.3: Update organisms by parasitism phase:

In SOS, by duplicating organism Org_i an artificial parasite called ‘‘Parasite_Vector’’ is created in the search space. Select another organism Org_j randomly from the ecosystem and serves as a host to the parasite vector. If Parasite_Vector has a better fitness value, it will kill organism Org_j and assume its position in the ecosystem. If the fitness value of Org_j is better, Org_j will have immunity from the parasite and the Parasite_Vector will no longer be able to live in that ecosystem.

Step 5.4: Update Organisms by Quasi-opposition generation jumping:

Based on a jumping rate Jr (i.e. Jumping probability), QOP is generated and fitness value of the QOP is calculated as given in Section 3.3.2. Select require numbers of the fittest organisms from {Org, QOP} as current Organisms.

Step 6: If the stopping criterion is not satisfied, go to Step 5, else return the organism with the best fitness value as the optimum solution.

5. Performance Evaluation of Well Known Test Functions

To validate a new optimization technique, it is a common practice to compare different algorithms using different benchmark problems. In this work, also, the convergence graph, three experiments, and two real life problems are considered for verifying the performance of the proposed algorithm. The statistical measures are obtained with different independent runs. Details of benchmark functions are given in the Appendices. The proposed algorithm coded in MATLAB7.10.0 (R2010a).

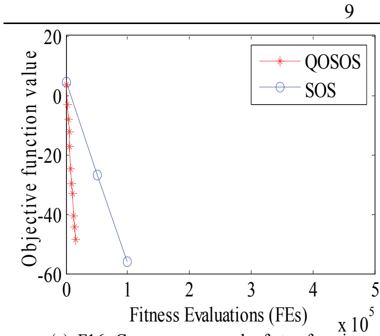
5.1. Performance Evaluation on Twenty Six Well Known Benchmark Functions

Table 1 shows the statistical measures of 26 benchmark functions which are given in Appendix-1. The performance results are compared with another algorithm like GA, DE, PSO, BA, PBA and SOS (Cheng & Prayogo, 2014). To compare the performance results of the proposed algorithm on twenty six benchmark functions, the algorithm runs 30 times with 500000 function evaluations. In Table 1, results except QOSOS are taken from Cheng and Prayogo (2014). From Table 1, it can be seen that the performance of proposed QOSOS algorithm is better than other compared algorithm on 22 benchmark functions out of 26 benchmark functions. The convergence rate of the QOSOS algorithm is also examined. The comparison of QOSOS is performed with SOS algorithm. Fig.2 shows the convergence graph of ten different benchmark functions which are given in Appendix-1 (Fig 2.(a) for Step function, (b) Sphere function, (c) for Sum Squares function, (d) for Quartic function, (e) for Schwefel2.22, (f) for Schwefel1.2, (g) for Rosenbrock, (h) for Dixon-Price, (i) for Rastrigin function, and (j) for Ackley function). From Fig. 2 it can be easily seen that QOSOS converges faster than SOS algorithm. Above all, these experimental results demonstrate that the proposed QOSOS algorithm performs better than other compared algorithms.

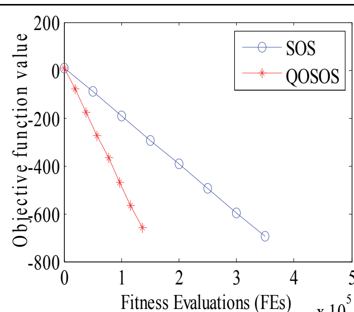
Table 1

Performance results of GA, DE, PSO, BA, PBA, SOS and QOSOS after reaching 500000 FEs of twenty six well known test functions over 30 runs. ''Mean '' and ''SD'' show the average and standard deviation of the test function respectively. Boldface represents the best result among the compared algorithms

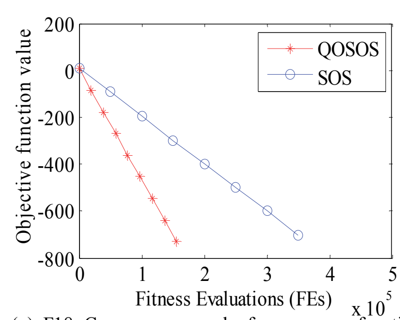
Function		Min	GA	DE	PSO	BA	PBA	SOS	QOSOS
1	Mean	0	**0**	**0**	**0**	1.88e-05	**0**	**0**	**0**
	SD		0	0	0	1.94e-05	0	0	0
2	Mean	-1	**-1**	**-1**	**-1**	-0.99994	**-1**	**-1**	**-1**
	SD		0	0	0	4.50e-05	0	0	0
3	Mean	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	SD		0	0	0	0	0	0	0
4	Mean	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	SD		0	0	0	0	0	0	0
5	Mean	0	**0**	**0**	**0**	0.00053	**0**	0.03382	**0**
	SD		0	0	0	0.00074	0	0.12870	0
6	Mean	-1.8013	**-1.8013**	**-1.8013**	-1.57287	**-1.8013**	**-1.8013**	**-1.8013**	**-1.8013**
	SD		0	0	0.11986	0	0	0	1.0299e-006
7	Mean	0	0.00424	**0**	**0**	**0**	**0**	**0**	**0**
	SD		0.00476	0	0	0	0	0	0
8	Mean	-1.03163	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**	**-1.0316**
	SD		0	0	0	0	0	0	0
9	Mean	0	0.06829	**0**	**0**	**0**	**0**	**0**	**0**
	SD		0.07822	0	0	0	0	0	0
10	Mean	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	SD		0	0	0	0	0	0	0
11	Mean	-186.73	**-186.73**	**-183.73**	**-186.73**	**-186.73**	**-186.73**	**-186.73**	**-186.7305**
	SD		0	0	0	0	0	0	0.00026809
12	Mean	0	0.01494	0.04091	**0**	1.11760	**0**	**0**	**0**
	SD		0.00736	0.08198	0	0.46623	0	0	0
13	Mean	-4.6877	-4.64483	-4.68348	-2.49087	**-4.6877**	**-4.6877**	**-4.6877**	**-4.6877**
	SD		0.09785	0.01253	0.25695	0	0	0	4.5274e-008
14	Mean	0	0.01336	**0**	**0**	**0**	**0**	**0**	**0**
	SD		0.00453	0	0	0	0	0	0
15	Mean	-9.6602	-9.49683	-9.59115	-4.00718	**-9.6602**	**-9.6602**	-9.65982	-9.6595
	SD		0.14112	0.06421	0.50263	0	0	0.00125	0.001698
16	Mean	0	1.17e+03	**0**	**0**	5.12370	**0**	**0**	**0**
	SD		76.56145	0	0	0.39209	0	0	0
17	Mean	0	1.11e+03	**0**	**0**	**0**	**0**	**0**	**0**
	SD		74.21447	0	0	0	0	0	0
18	Mean	0	1.48e+02	**0**	**0**	**0**	**0**	**0**	**0**
	SD		12.40929	0	0	0	0	0	0
19	Mean	0	0.18070	0.00136	0.00116	**1.72e-06**	0.00678	9.13e-05	3.2708e-005
	SD		0.02712	0.00042	0.00028	1.85e-06	0.00133	3.71e-05	1.7536e-005
20	Mean	0	11.0214	**0**	**0**	**0**	7.59e-10	**0**	**0**
	SD		1.38686	0	0	0	7.10e-10	0	0
21	Mean	0	7.40e+03	**0**	**0**	**0**	**0**	**0**	**0**
	SD		1.14e+03	0	0	0	0	0	0
22	Mean	0	1.96e+05	18.20394	15.088617	28.834	4.2831	**1.04e-07**	1.0354
	SD		3.85e+04	5.03619	24.170196	0.10597	5.7877	2.95e-07	1.3959
23	Mean	0	1.22e+03	0.66667	0.66667	0.66667	0.66667	**0**	**0**
	SD		2.66e+02	1e-9	1e-8	1.16e-09	5.65e-10	0	0
24	Mean	0	52.92259	11.71673	43.97714	**0**	0	**0**	**0**
	SD		4.56486	2.53817	11.72868	0	0	0	0
25	Mean	0	10.63346	0.00148	0.01739	**0**	0.00468	**0**	**0**
	SD		1.16146	0.00296	0.02081	0	0.00672	0	0
26	Mean	0	14.67178	**0**	0.16462	**0**	3.12e-08	**0**	**0**
	SD		0.17814	0	0.49387	0	3.98e-08	0	0
			9	18	17	18	20	22	22



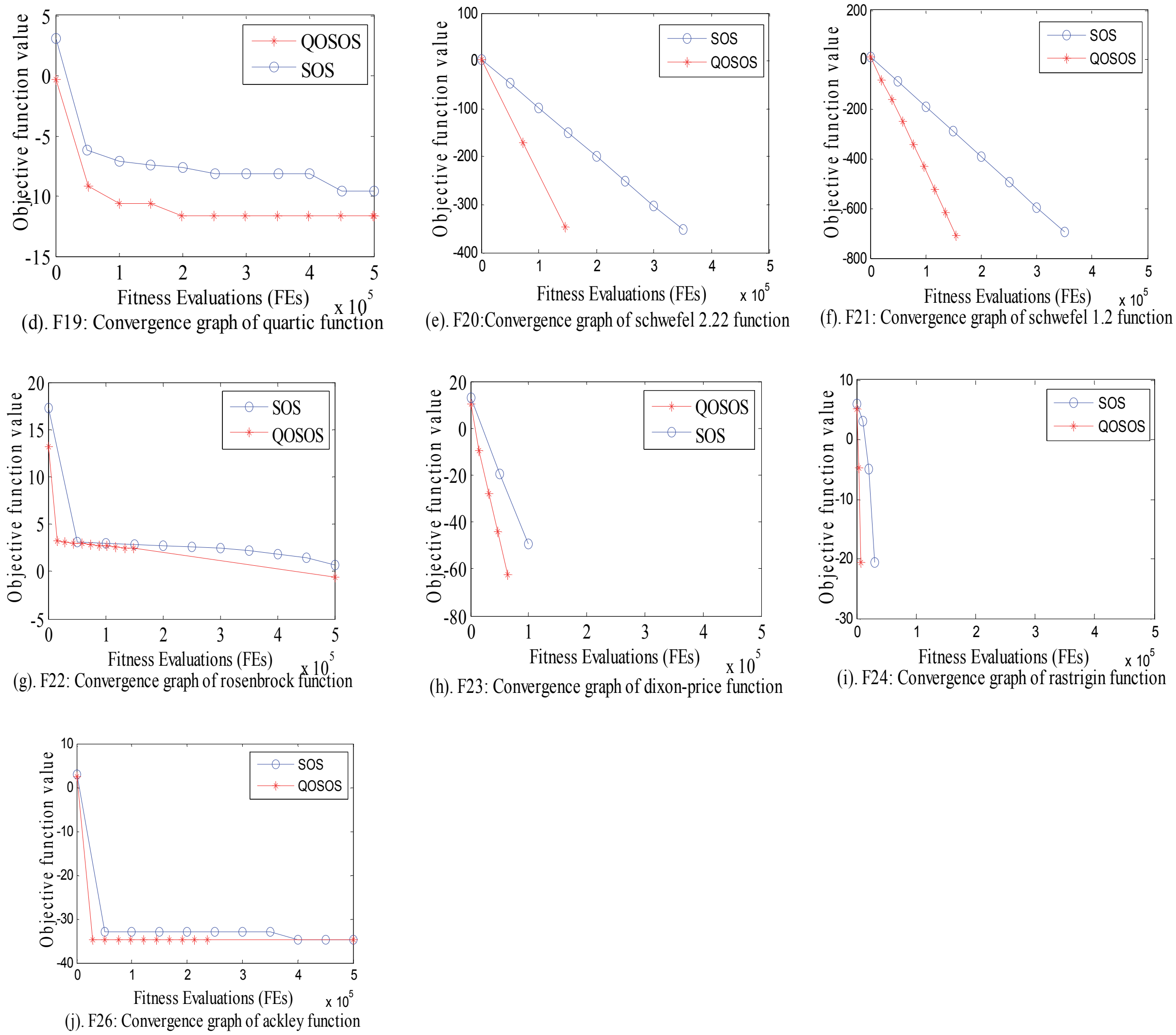
(a). F16: Convergence graph of step function



(b). F17:Convergence graph of sphere function



(c). F18: Convergence graph of sum squares function



(d). F19: Convergence graph of quartic function

(e). F20:Convergence graph of schwefel 2.22 function

(f). F21: Convergence graph of schwefel 1.2 function

(g). F22: Convergence graph of rosenbrock function

(h). F23: Convergence graph of dixon-price function

(i). F24: Convergence graph of rastrigin function

(j). F26: Convergence graph of ackley function

Fig. 2. Convergence graph of ten unconstrained functions which are given in Appendix 1 ((a) F16, (b) F17, (c) F18, (d) F19, (e) F20, (f) F21, (g) F22, (h) F23, (i) F24 and (j) F26).

5.2. Performance Evaluation on Eight Well Known Benchmark Functions

Considering the problem given in Appendix-2, the performance results and maximum function evaluation of all eight well known test functions are reported in Table 2 and Table 3 along with the results as quoted in (Gao et al., 2012) for 30, 50, 100, 200 and 300 dimensions, respectively with 20 independent runs. The authors in Gao et al. (2012) compares CSPSO with several state of the art algorithms, namely SDE, SPSO, MPSO-TVSE, DPSO with 5000×D (D = dimension of the problem) function evaluations and 40 eco-size. In Table 2, the bold face results represent the best performance results among the compared algorithms. From the results in Table 2, it is observed that QOSOS performs better than all other algorithms for solving lower to high dimensional problems with less number of function evaluations.

Table 2
Performance results of SDE, SPSO, MPSO-TVAC, DPSO, CSPSO and QOSOS at dimensions (D) 30, 50, 100, 200 and 300 after reaching D*5000 FEs of four (F1-F4) well known benchmark functions over 20 runs with 40 eco-size. ''Min '' and ''std.'' show the best and standard deviation of the benchmark functions respectively. Boldface represents the best result among the compared algorithms

F	D	Result	SDE	SPSO	MPSO-TVAC	DPSO	CSPSO	QOSOS
1	30	Min	-5.31e+003	-6.72e+003	-6.62e+003	-8.58e+003	-1.25e+004	**-1.18e+004**
		Std.	4.90e+002	1.02e+003	6.15e+002	4.63e+002	3.78e-012	5.74e+002
	50	Min	-7.04e+003	-1.01e+004	-9.77e+003	-1.38e+004	-2.09e+004	**-1.88e+004**
		Std.	3.03e+002	1.32e+003	7.92e+002	7.45e-012	3.63e-012	1.18e+003
	100	Min	-1.13e+004	-1.81e+004	-1.79e+004	-2.72e+004	-4.18e+004	**-3.39e+004**
		Std.	1.07e+003	2.20e+003	1.51e+003	1.19e+003	5.31e-012	1.74e+003
	200	Min	-2.48e+004	-3.13e+004	-4.02e+004	-5.51e+004	-8.37e+004	**-6.41e+004**
		Std.	1.83e+003	4.21e+003	4.36e+003	1.99e+003	1.63e-011	3.16e+003
	300	Min	-3.69e+004	-4.34e+004	-5.69e+004	-7.99e+004	-1.25e+005	**-8.98e+004**
		Std.	4.18e+003	7.01e+003	3.51e+003	4.49e+003	3.16e-011	4.03e+003
2	30	Min	1.72e+002	1.99e+001	1.60e+001	6.40e+000	**0**	**0.00e+000**
		Std.	1.14e+001	5.17e+000	4.06e+000	5.07e+000	0	0.00e+000
	50	Min	3.24e+002	3.99e+001	3.64e+001	1.53e+001	**0**	**0.00e+000**
		Std.	2.13e+001	7.93e+000	6.52e+000	5.58e+000	0	0.00e+000
	100	Min	5.27e+002	9.37e+001	8.81e+001	4.14e+001	**0**	**0.00e+000**
		Std.	9.79e+001	9.96e+000	9.12e+000	7.33e+000	**0**	0.00e+000
	200	Min	1.46e+002	2.23e+002	1.94e+002	9.98e+001	**0**	**0.00e+000**
		Std.	1.81e+001	1.74e+001	3.08e+001	1.14e+001	0	0.00e+000
	300	Min	2.48e+002	3.63e+002	2.66e+002	2.12e+002	**0**	**0.00e+000**
		Std.	2.08e+001	1.76e+001	2.92e+001	3.71e+001	0	0.00e+000
3	30	Min	1.38e+002	1.02e+001	1.45e+001	5.12e+000	**0**	**0.00e+000**
		Std.	9.01e+000	3.42e+000	2.89e+000	4.1e+000	**0**	0.00e+000
	50	Min	4.13e+002	2.43e+001	2.31e+001	1.26e+001	**0**	**0.00e+000**
		Std.	8.09e+001	6.34e+000	5.17e+000	3.89e+000	0	0.00e+000
	100	Min	8.97e+002	6.35e+001	6.62e+001	3.38e+001	**0**	**0.00e+000**
		Std.	9.79e+001	7.38e+000	7.58e+000	5.79e+000	0	0.00e+000
	200	Min	3.28e+002	1.97e+002	1.57e+002	7.45e+001	**0**	**0.00e+000**
		Std.	8.18e+001	5.79e+001	2.63e+001	1.00e+001	0	0.00e+000
	300	Min	5.73e+002	3.21e+002	2.42e+002	1.89e+002	**0**	**0.00e+000**
		Std.	4.28e+001	5.06e+001	2.02e+001	2.82e+001	0	0.00e+000
4	30	Min	7.52e-016	9.72e-003	9.45e-003	7.08e-003	**0**	**0.00e+000**
		Std.	4.25e-016	1.11e-002	1.63e-002	8.64e-003	0	0.00e+000
	50	Min	5.36e-004	5.81e-003	7.08e-002	5.82e-002	**0**	**0.00e+000**
		Std.	1.90e-003	8.46e-004	9.45e-003	1.46e-003	0	0.00e+000
	100	Min	1.10e-002	2.12e-003	2.25e-002	2.10e-002	**0**	**0.00e+000**
		Std.	2.45e-002	4.30e-003	3.11e-002	1.71e-002	0	0.00e+000
	200	Min	5.36e-002	5.37e-002	9.01e-002	4.79e-002	1.11e-016	**0.00e+000**
		Std.	7.51e-002	1.07e-001	1.02e-001	9.40e-002	0	0.00e+000
	300	Min	8.01e-001	5.58e-001	3.99e-001	4.52e-001	3.33e-016	**0.00e+000**
		Std.	5.17e-001	2.11e-001	2.40e-001	1.91e-001	0	0.00e+000

Table 3

Performance results of SDE, SPSO, MPSO-TVAC, DPSO, CSPSO and QOSOS at dimensions (D) 30, 50, 100, 200 and 300 after reaching D*5000 FEs of four (F5-F8) well known benchmark functions over 20 runs with 40 eco-size. ''Min '' and ''std.'' show the best and standard deviation of the benchmark functions respectively. Boldface represents the best result among the compared algorithms

F	D	Result	SDE	SPSO	MPSO-TVAC	DPSO	CSPSO	QOSOS
5	30	Min	3.86e-009	7.59e-006	6.51e-014	4.78e-011	2.57e-014	**8.88e-016**
		Std.	1.80e-009	1.37e-005	8.53e-014	9.15e-011	1.77e-015	0.00e+000
	50	Min	9.92e-010	1.70e-004	9.95e-005	1.58e-008	5.70e-014	**8.88e-016**
		Std.	3.65e-010	1.28e-004	1.73e-004	1.78e-008	3.82e-015	0.00e+000
	100	Min	3.18e-002	3.31e-001	4.69e-001	3.68e-007	1.33e-013	**1.07e-015**
		Std.	1.72e-001	5.01e-001	1.91e-001	1.63e-007	6.36e-015	7.94e-016
	200	Min	1.58e-000	2.13e+000	6.94e-001	9.49e-007	2.89e-013	**1.60e-015**
		Std.	2.59e-001	2.19e-001	4.08e-001	4.07e-007	1.28e-014	1.46e-015
	300	Min	3.05e+000	3.15e+000	7.66e-001	1.59e-006	4.51e-013	**2.31e-015**
		Std.	5.37e-001	5.60e-001	3.16e-001	7.38e-007	1.03e-014	1.79e-015
6	30	Min	3.37e-016	7.38e-004	7.93e-023	5.16e-023	**1.57e-032**	**1.57e-032**
		Std.	2.89e-016	2.79e-003	2.50e-022	1.74e-022	2.73e-048	2.81e-048
	50	Min	1.80e-017	6.74e-003	1.19e-002	1.62e-017	**9.42e-033**	**9.42e-033**
		Std.	1.60e-017	5.44e-003	3.03e-002	3.93e-017	1.36e-048	2.81e-048
	100	Min	1.20e-003	2.90e+001	3.77e-001	8.24e-011	**4.71e-033**	**4.71e-033**
		Std.	5.98e-003	1.53e+001	6.13e-001	1.79e-010	6.84e-049	1.40e-048
	200	Min	1.18e-002	1.81e+003	2.17e+000	1.74e-010	4.22e-033	**2.36e-033**
		Std.	2.90e-002	1.74e+003	1.61e+000	1.07e-010	8.22e-034	7.02e-049
	300	Min	7.26e-002	1.47e+004	3.73e+000	4.03e-011	4.56e-033	**7.02e-049**
		Std.	9.76e-002	9.03e-003	2.68e+000	3.31e-010	6.50e-034	1.57e-033
7	30	Min	4.46e-015	5.49e-004	9.36e-027	6.31e-027	**1.35e-032**	**1.35e-032**
		Std.	3.62e-015	2.45e-003	4.17e-026	5.72e-027	0	2.81e-048
	50	Min	7.06e-016	5.92e-003	1.19e-002	8.25e-017	**1.35e-032**	**1.35e-032**
		Std.	9.22e-016	4.54e-003	3.03e-002	4.17e-017	0	2.81e-048
	100	Min	9.81e-003	3.80e+001	4.28e-001	6.73e-011	3.84e-032	**1.35e-032**
		Std.	7.24e-003	1.82e+001	2.01e-001	4.12e-010	9.78e-033	2.81e-048
	200	Min	8.34e-002	1.36e+003	4.21e+000	7.54e-010	2.58e-031	**1.35e-032**
		Std.	3.71e-002	1.23e+003	2.31e+000	6.27e-010	9.02e-032	2.81e-048
	300	Min	6.21e-002	3.27e+004	4.98e+000	1.29e-011	4.76e-031	**1.35e-032**
		Std.	3.72e-002	4.44e+004	3.28e+000	2.48e010	1.71e-032	2.81e-048
8	30	Min	3.40e-017	1.25e-004	5.26e-024	1.93e-024	**1.35e-031**	1.52e-031
		Std.	2.46e-017	3.49e-003	7.58e-024	4.38e-024	0	2.86e-032
	50	Min	4.69e-019	8.19e-003	5.36e-002	8.29e-017	**1.35e-031**	2.42e-031
		Std.	1.58e-019	2.57e-003	4.89e-002	4.27e-017	0	9.23e-032
	100	Min	1.83e-003	8.41e+001	1.53e-001	5.34e-011	2.99e-031	**2.55e-031**
		Std.	4.09e-003	4.27e+001	8.49e-001	2.87e-010	1.65e-031	2.97e-031
	200	Min	5.71e-003	5.01e+003	5.19e+000	3.52e-010	8.28e-031	**1.35e-031**
		Std.	8.21e-003	2.71e+003	1.24e+000	2.73e-010	7.44e-031	0.00e+000
	300	Min	9.70e-002	2.48e+004	5.72e+000	4.03e-011	1.40e-030	**1.35e-031**
		Std.	8.04e-002	4.20e+003	1.83e+000	3.31e-010	5.13e-031	0.00e+000

5.3. Performance Evaluation on Six Well Known Benchmark Functions

Table 4 shows the performance results of six well known test functions which are given in Appendix-3. The algorithm runs 20 times with D×5000 fitness evaluation and 40 eco-size. In Table 4, the results except QOSOS are taken from (Gao et al., 2012). The performance results are compared with GPSO (Shi & Eberhart, 1998), LPSO (Kennedy & Mendes, 2002), VPSO (Kennedy & Mendes, 2006), FIPS (Mendes et al., 2004), DWS-PSO (Liang & Suganthan, 2005), CLPSO (Liang et al., 2006), APSO (Zhan et al., 2009) and CSPSO (Gao et al., 2012). From Table 4, it is seen that overall performance of

QOSOS on benchmark functions Schwefel, Rastrigin, Non-continuous Rastrigin, Griewank and Penalized are better.

Table 4
Performance results of GPSO (27), LPSO (28), VPSO (29), FIPS (30), DWS-PSO (31), CLPSO (32), APSO (33), CSPSO (21) and QOSOS at dimensions (D) 30 after reaching D*5000 FEs of eight well known test functions over 20 runs with 40 eco-size. ''Mean '' and ''std.'' show the average and standard deviation of the test functions respectively. Boldface represents the best result among the compared algorithms

Algorithms	Function→	Schwefel	Rastrigin	NCRastrigin	Ackley	Griewank	Penalized
GPSO	Mean	-10090.16	30.7	15.5	1.31e-014	2.37e-002	1.04e-002
	Std.	495	8.68	7.4	2.08e-015	2.57e-002	3.16e-002
LPSO	Mean	-9628.35	34.90	30.40	8.20e-008	2.10e-002	2.18e-030
	Std.	456.54	7.25	9.23	8.73e-008	1.60e-002	5.14e-030
VPSO	Mean	-9845.27	34.09	21.33	1.4e-014	1.31e-002	3.46e-003
	Std.	588.87	8.07	9.46	3.48e-015	1.35e-002	1.89e-002
FIPS	Mean	-10133.8	29.98	35.97	2.33e-014	9.04-004	1.22e-031
	Std.	889.58	10.92	9.49	7.19e-016	2.78e-003	4.85e-032
DWS-PSO	Mean	9593.33	28.1	32.8	1. 84-014	1. 31e-002	2. 05e-032
	Std.	441	6.42	6.49	4.35e- 015	1.73e-002	8.12e-033
CLPSO	Mean	-12557.65	2.57e-011	0.167	3.66e-007	6.45e-013	1.59e-021
	Std.	36.2	6.64e-011	0.379	7.57e-008	2.07e-012	1.93e-021
APSO	Mean	-12569.5	5.8e-015	4.14e-016	1.11e-014	1.67e-002	3.76e-031
	Std.	5.22e-011	1.01e-014	1.45e-015	3.55e-015	2.41e-002	1.2e-030
CSPSO	Mean	-12569.5	0	0	2.57e-014	0	**1.57e-032**
	Std.	3.78e-012	0	0	1.77e-015	0	2.73e-048
QOSOS	Mean	**-1.18e+004**	**0.00e+000**	**0.00e+000**	**8.88e-016**	**0.00e+000**	**1.57e-032**
	Std.	5.74e+002	0.00e+000	0.00e+000	0.00e+000	0.00e+000	2.81e-048

6. Performance Results on Two Real World Problems

Table 5 and Table 6 show the performance results on two real life optimization problems. The mathematical models of these two problems are given in Appendix-3. The minimum objective function values for RP1 and RP2 are reported in Tables 5 and Table 6, respectively, along with results quoted in (Das & Singh, 2014). The performance results are compared with DE, GSA, DE-DSA, DFO, DFO (QA) and Beightler and Phillips (1976). In Table 5 and Table 6, values in the bold print show best values for QOSOS that are superior to DE, GSA, DE-DSA, and Beightler and Phillips (25). From these two tables, it is seen that QOSOS gives better performance than the other algorithms.

Table 5
Performance results of DE, GSA, DE-GSA, DFO and QOSOS on gas transmission compressor design optimization problem. Boldface represents the best result among the compared algorithms

Item	DE	GSA	DE-DSA	DFO	DFO(QA)	Beightler and Phillips (25)	QOSOS
X_1	52.3966	53.0547	53.5080	53.4831	53.4961	55	**53.4467**
X_2	1.1875	1.1919	1.1901	1.19011	1.1901	1.195	**1.1901**
X_3	24.6697	24.5070	24.7624	24.715	24.7149	25.026	**24.7186**
f(X)	2.96443E+06	2.96449E+06	2.96437E+06	2.96291E+06	2.9631E+06	2.96455E+06	**2.96438e+006**

Table 6
Performance results of DE, GSA, DE-GSA, DFO and QOSOS on Optimal Capacity of the Gas production facilities optimization problem. Boldface represents the best result among the compared algorithms

Item	DE	GSA	DE-DSA	DFO	DFO(QA)	Beightler and Phillips (25)	QOSOS
X_1	17.5	17.5	17.5	17.5	17.5	17.5	**17.5**
X_2	600	600	600	600	600	465	**600**
f(X)	169.844	169.844	169.844	169.844	169.844	173.76	**169.8437**

7. Conclusion

This paper has developed an ensemble algorithm combining quasi-oppositional based learning with Symbiosis Organisms Search algorithm, called quasi-oppositional Symbiosis Organisms Search (QOSOS) algorithm. The proposed algorithm solved several unconstrained benchmark function to enrich the searching behavior and to avoid being trapped into a local optimum. The comparison results show that the proposed QOSOS method has higher efficiency in terms of the numerical results than the other reported optimization techniques to solve the benchmark problems. Also, the proposed algorithm has been applied to solve two real world optimization problems and the results obtained by proposed algorithm were compared with other algorithms available in the literature. The comparative study shows the satisfactory performance of the proposed QOSOS algorithm.

References

Abdullahi, M., & Ngadi, M. A. (2016). Symbiotic Organism Search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems, 56*, 640-650.

Beighter, C.S., & Phillips, D.T. (1976). *Applied Geometric Programming*. John Wiley and Sons.

Cheng, M. Y., Prayogo, D., & Tran, D. H. (2015). Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search. *Journal of Computing in Civil Engineering*, *30*(3), 04015036.

Cheng, M. Y., & Prayogo, D. (2014). Symbiotic organisms search: a new metaheuristic optimization algorithm. *Computers & Structures*, *139*, 98-112.

Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys (CSUR)*, *45*(3), 35.

Das, K. N., & Singh, T. K. (2014). Drosophila food-search optimization. *Applied mathematics and Computation*, *231*, 566-580.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Bradford Company.

Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, *110*, 151-166.

Gao, W. F., Liu, S. Y., & Huang, L. L. (2012). Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique. *Communications in Nonlinear Science and Numerical Simulation*, *17*(11), 4316-4327.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *Simulation*, *76*(2), 60-68.

Gong, W., Cai, Z., & Ling, C. X. (2010). DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Computing*, *15*(4), 645-665.

Holland, J.H. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, USA.

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

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *In: Proceedings of the IEEE international conference on neural networks. Perth, Australia*;. p. 1942– 1948.

Kennedy, J., & Mendes, R. (2002), Population structure and particle swarm performance, *In: IEEE international conference evolutionary computation, Honolulu, HI,* p. 1671–1676.

Kennedy, J., & Mendes, R., (2006), Neighborhood topologies in fully informed and best-of neighborhood particle swarms. *IEEE Transactions Systems, Man, and Cybernetics, Part C*, *36*(4), 515–519.

Lee, K. S., & Geem, Z. W. (2004). A new structural optimization method based on the harmony search algorithm. *Computers & structures*, *82*(9), 781-798.

Liang, J.J., Qin, A.K., Suganthan, P.N., & Baskar, S., (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions Evolutionary Computation, 10*(3), 281–295.

Liang, J.J., & Suganthan, P.N. (2005). Dynamic multi-swarm particle swarm optimizer, *In: IEEE swarm intelligence symposium,* p. 124–129.

Lee, K. S., & Geem, Z. W. (2004). A new structural optimization method based on the harmony search algorithm. *Computers & structures*, *82*(9), 781-798.

Mendes, R., Kennedy, J., Neves, J., (2004), The fully informed particle swarm: simpler maybe better. IEEE Transactions Evolutionary Computation, 8(3), 204–210.

Nama, S., Saha, A. K., & Ghosh, S. (2015a). Parameters optimization of geotechnical problem using different optimization algorithm. *Geotechnical and Geological Engineering*, *33*(5), 1235-1253.

Nama, S., Saha, A.K., & Ghosh, S. (2016b). A new ensemble algorithm of differential evolution and backtracking search optimization algorithm with adaptive control parameter for function optimization, *International Journal of Industrial Engineering Computations*,*7*(2), 323-338.

Nama, S., Saha, A., & Ghosh, S. (2016c). Improved symbiotic organisms search algorithm for solving unconstrained function optimization. *Decision Science Letters*, *5*(3), 361-380.

Nama, S., Saha, A.K., and Ghosh, S. (2016d). A hybrid symbiosis organisms search algorithm and its application to real world problems. *Memetic Computing* (DOI: 10.1007/s12293-016-0194-1)

Nama, S., Saha, A. K., & Ghosh, S. (2017). Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c-Φ backfill. *Applied Soft Computing*, *52*, 885-897.

Osman, I.H., & Laporte, G. (1996). Metaheuristics: a bibliography. *Annals of Operations Research*, *63*,511-623

Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., & Zaidi, M. (2006), The Bees algorithm, a novel tool for complex optimization problems. *In: Proceedings of the 2nd international virtual conference on intelligent production machines and systems (IPROMS 2006). Elsevier: Oxford*; p. 454–9.

Prasad, D., & Mukherjee, V. (2015). A novel symbiotic organisms search algorithm for optimal power flow of power system with FACTS devices. *Engineering Science and Technology, an International Journal, 19*(1), 79-89.

Xu, Q., Wang, L., He, B. M., & Wang, N. (2011). Modified opposition-based differential evolution for function optimization. *Journal of Computational Information Systems*, *7*(5), 1582-1591.

Roy, P.K., & Mandal, D. (2011). Quasi-oppositional biogeography-based optimization for multi-objective optimal power flow. *Electric Power Components and Systems, 40*(2), 236–56.

Roy, P.K., & Mandal, D. (2014). Oppositional biogeography-based optimization for optimal power flow. *International Journal of Power Energy Conversion, 5*(1), 47–69.

Sadollah, A., Bahreininejad, A., Eskandar, H., Hamdi, M., (2012), Mine blast algorithm for optimization of truss structures with discrete variables. *Computers and Structures, 102*,49–63

Sapp, J. (1994). *Evolution by association: a history of symbiosis: a history of symbiosis*. New York: Oxford University Press, USA.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer, *In: IEEE international conference on computational intelligence,* p. 69–73.

Storn, R., Price , K., (1997), Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*, 341–59.

Sultana, S., & Roy, P.K. (2014), Multi-objective quasi-oppositional teaching learning based optimization for optimal location of distributed generator in radial distribution systems, *Electrical Power and Energy Systems, 63*, 534–545.

Tizhoosh, H.R. (2005). Opposition-based learning: a new scheme for machine intelligence. *In: International conference on computational intelligence for modelling control and automation*; *Austria:* p. 695–01.

Verma, S., Saha, S., & Mukherjee, V. (2017). A novel symbiotic organisms search algorithm for congestion management in deregulated environment. *Journal of Experimental & Theoretical Artificial Intelligence*, *29*(1), 59-79.

Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., & Ventresca, M. (2011). Enhancing particle swarm optimization using generalized opposition-based learning. *Information Sciences*, *181*(20), 4699-4714.

Yang, X-S. (2009). Firefly algorithms for multimodal optimization. In: Proceedings of the 5th international conference on stochastic algorithms: foundations and applications. *Sapporo, Japan: Springer-Verlag*, 169–78.

Yang, X-S., & Deb, S. (2009), Cuckoo search via levy flights. *In: Proceedings of the world congress on nature & biologically inspired computing (NaBIC-2009). Coimbatore, India.* p. 210–214.

Zhan, Z.H., Zhang, J., Li, Y., & Chung, H.H. (2009), Adaptive particle swarm optimization. *IEEE Transactions B, 39*(6), 1362–1381.

Appendix-1

F1: Beale, F2: Eason, F3: Mathyas, F4:, F5: Booth, F6: Michalewicz2, F7: Schaffer, F8: Six Hump Camel back, F9: Bohachevsky2, F10: Bohachevsky3, F11: Shubert, F12: Colville, F13: Michalewicz5, F14: Zakharov, F15: Michalewicz10, F16: Step, F17: Sphere, F18: SumSquares, F19: Quartic, F20: Schwefel2.22, F21: Schwefel1.2, F22: Rosenbrock, F23: Dixon-Price, F24: Rastrigin, F25: Griewank, F26: Ackley

F	Formulation	D	S	fmin
F1	$f(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2^{\ 2})^2+(2.625-x_1+x_1x_2^{\ 3})^2$	2	(-4.5,4.5)	0
F2	$f(x)=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$	2	(-100,100)	-1
F3	$f(x)=0.26(x_1^{\ 2}+x_2^{\ 2})-0.48x_1x_2$	2	(-10,10)	0
F4	$f(x)=x_1^{\ 2}+2x_2^{\ 2}-0.3\cos(3\pi x_1)-0.48\cos(4\pi x_2)+0.7$	2	(-100,100)	0
F5	$f(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	2	(-10, 10)	0
F6	$f(x)=-\sum_{i=1}^{D}\sin(x_i)(\sin(ix_i^{\ 2}/\pi))^{20}$	2	(0, π)	-1.8013
F7	$0.5+\dfrac{\sin^2\sqrt{\sum_{i=1}^{D}x_i^{\ 2}}-0.5}{1+0.001(\sum_{i=1}^{D-1}x_i^{\ 2}+x_{i+1}^{\ 2})}$	2	(-100, 100)	0
F8	$f(x)=4x_1^{\ 2}-2.1x_1^{\ 4}+\frac{1}{3}x_1^{\ 6}+x_1x_2-4x_2^{\ 2}+4x_2^{\ 4}$	2	(-5, 5)	-1.03163
F9	$f(x)=x_1^{\ 2}+2x_2^{\ 2}-0.3\cos(3\pi x_1)(4\pi x_2)+0.3$	2	(-100,100)	0
F10	$f(x)=x_1^{\ 2}+2x_2^{\ 2}-0.3\cos(3\pi x_1+4\pi x_2)+0.3$	2	(-100,100)	0
F11	$f(x)=(\sum_{i=1}^{5}i\cos((i+1)x_1+i))\times(\sum i\cos((i+1)x_2+i))$	2	(-10, 10)	-186.73
F12	$f(x)=100(x_1^{\ 2}-x_1^{\ 2})^2+(x_1-1)^2+(x_3-1)^2+(x_4^{\ 2}-1)^2+90(x_3^{\ 2}-x_4)^2$ $+10.1(x_2-1)^2+19.8(x_2-1)(x_4-1)$	4	(-10, 10)	0
F13	$f(x)=-\sum_{i=1}^{D}\sin(x_i)(\sin(ix_i^{\ 2}/\pi))^{20}$	5	(0, π)	-4.6877
F14	$f(x)=\sum_{i=1}^{D}x_i^{\ 2}+(\sum_{i=1}^{D}0.5ix_i)^2+(\sum 0.5ix_i)^4$	10	(-5, 10)	0

F15	$f(x) = -\sum_{i=1}^{D} \sin(x_i)(\sin(ix_i^2/\pi))^{20}$	10	(0, π)	-9.6602
F16	$f(x) = \sum_{i=1}^{D} (x_i + 0.5)^2$	30	(5.12, 5.12)	0
F17	$f(x) = \sum_{i=1}^{D} x_i^2$	30	(-100,100)	0
F18	$f(x) = \sum_{i=1}^{D} ix_i^2$	30	(-10, 10)	0
F19	$f(x) = \sum_{i=1}^{D} ix_i^4 + random(0,1)$	30	(-1.28, 1.28)	0
F20	$f(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30	(-10, 10)	0
F21	$f(x) = \sum_{i=1}^{D} \sum_{j=1}^{i} x_i^2$	30	(-100, 100)	0
F22	$f(x) = \sum_{i=1}^{D} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	(30, 30)	0
F23	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{D} i(2x_i^2 - x_i - 1)^2$	30	(-10, 10)	0
F24	$f(x) = 10D + \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)]$	30	(-5.12, 5.12)	0
F25	$f(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) - 1$	30	(-600, 600)	0
F26	$f(x) = 20 + e - 20e^{\frac{1}{D}(\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2})} - e^{\frac{1}{D}(\sum \cos(2\pi x_i))}$	30	(-32,32)	0

Appendix-2

F1: Schwefel, F2: Rastrigin, F3: Non continuous Rastrigin, F4: Griewank, F5: Ackley, F6: Penalized, F7: Penalized1, F8: Levy

F	Formulation	S	fmin
F1	$f(x) = -\sum x_i \sin(\sqrt{\lvert x_i \rvert})$	(-500,500)	-12569.5
F2	$f(x) = 10D + \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)]$	(-5.12,5.12)	0
F3	$f(x) = 10D + \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i)]$ Where $y_i = \begin{cases} x_i & \lvert x_i \rvert \le \frac{1}{2} \\ \frac{round(2x_i)}{2} & \lvert x_i \rvert \ge \frac{1}{2} \end{cases}$	(-5.12,5.12)	0

F4	$f(x)=\sum_{i=1}^{D}\frac{x_i^{\,2}}{4000}-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})-1$	(-600, 600)	0
F5	$f(x)=20+e-20e^{\frac{1}{D}(\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^{\,2}})}-e^{\frac{1}{D}(\sum\cos(2\pi x_i))}$	(-32, 32)	0
F6	$f(x)=\frac{\pi}{D}\left\{10\sin^2(\pi y_i)+\sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2(3\pi y_{i+1}) + (y_D-1)^2]\right\} + \sum_{i=1}^{D}u(x_i,10,100,4)$ Where $y_i=1+\frac{1}{4}(x_i+1)$, $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<a\end{cases}$	(-50, 50)	0
F7	$f(x)=0.1\left\{10\sin^2(\pi x_i)+\sum_{i=1}^{D-1}(x_i-1)^2[1+10\sin^2(3\pi x_{i+1}) + (x_D-1)^2[1+\sin^2(2\pi x_D)]]\right\} + \sum_{i=1}^{D}u(x_i,5,100,4)$	(-50, 50)	0
F8	$\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+\sin^2(3\pi x_1)$	(-10, 10)	0

Appendix-3

RP1.Gas transmission compressor design problem (Beightler and Phillips, 1976):

$$Min\ \ f(x)=8.61\times10^5\times x_1^{\frac{1}{2}}\times x_2\times x_3^{\frac{-2}{3}}\times(x_2^2-1)^{-\frac{1}{2}}+3.69\times10^4\times x_3+7.72\times10^8\times x_1^{-1}\times x_2^{0.219}-765.43\times10^6\times x_1^{-1}$$

S.t $10\ \le x_1\ \le\ 55,\ 1.1\ \le\ x_2\le\ 2, 10\ \le x_3\ \le\ 40;$

RP2. Optimal capacity of gas production facilities (Beightler and Phillips, 1976):

$$Min\ f(x)=61.8+5.72\times x_1\times0.2623\times\left[(40-x_1)\times\ln\left(\frac{x_2}{200}\right)\right]^{-0.85}+0.087\times(40-x_1)\times\ln\left(\frac{x_2}{200}\right)+700.23\times x_2^{-0.75}$$

S.t $x_1\ \ge\ 17.5,\ x_2\ \ge\ \ 200,\ \ 17.5\ \le\ x_1\le\ 40, 300\le\ x_2\ \le\ 600;$