

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl**A novel hybrid backtracking search optimization algorithm for continuous function optimization**Sukanta Nama^{a*} and Apu Kumar Saha^b^aDepartment of Mathematics, Ram Thakur College Agartala, A.D. Nagar-799003, West Tripura, India^bDepartment of Mathematics, National Institute of Technology Agartala, Barjala, Jirania, Tripura-799046, India**CHRONICLE***Article history:*

Received April 3, 2018

Received in revised format:

May 10, 2018

Accepted July 9, 2018

Available online

July 9, 2018

*Keywords:**Backtracking Search Optimization Algorithm (BSA)**Quadratic approximation (QA)**Hybrid Algorithm**Unconstrained non-linear**function optimization***ABSTRACT**

Stochastic optimization algorithm provides a robust and efficient approach for solving complex real world problems. Backtracking Search Optimization Algorithm (BSA) is a new stochastic evolutionary algorithm and the aim of this paper is to introduce a hybrid approach combining the BSA and Quadratic approximation (QA), called HBSA for solving unconstrained non-linear, non-differentiable optimization problems. For the validity of the proposed method the results are compared with five state-of-the-art particle swarm optimization (PSO) variant approaches in terms of the numerical result of the solutions. The sensitivity analysis of the BSA control parameter (F) is also performed.

© 2018 by the authors; licensee Growing Science, Canada.

1. Introduction

Stochastic Optimization algorithms are effective and powerful tool for solving nonlinear complex optimization problem. Many nature based stochastic algorithms have been introduced and studied by many authors (e.g. Yang & Press, 2010). The success of an optimization algorithm depends on its significant development of exploration and exploitation abilities. The first attempt to start these types of studies is genetic algorithm (GA) (Holland, 1992) which actually employs the natural process of genetic evolution. After that various nature-inspired meta-heuristic approaches have been proposed such as differential evolution (DE) (Storn & Price, 1997), evolutionary strategy (ES) (Back, 1996; Beyer, 2001), particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), ant colony optimization (ACO) (Dorigo, 2004), cuckoo search (CS) (Gandomi et al., 2013), firefly algorithm (FA) (Gandomi, 2011), biogeography-based optimization (BBO) (Simon, 2008) big bang–big crunch algorithm (Erol & Eksin, 2006), charged system search (CSS) (Kaveh & Talatahari, 2010) animal migration optimization (AMO) (Li et al., 2013), water cycle algorithm (WCA) (Eskandar et al., 2012), mine blast Algorithm (MBA) (Sadollaha et al., 2013), harmony search algorithm (Mahdavi et al., 2007), improvements of Symbiosis Organisms Search Algorithm (Nama et al. 2016b, 2016b; Nama & Saha, 2018). Recently Civicioglu (2013) proposed a novel algorithm called backtracking search algorithm (BSA) which is

* Corresponding author.

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

based on the return of a social group at random intervals to hunting areas that were previously found fruitful for obtaining nourishment (Civicioglu, 2013; Nama et al., 2016d) BSA's strategy for generating a trial population includes two new crossover and mutation operators which are different from other evolutionary algorithm like DE and GA. BSA uses random mutation strategy with one direction individual for each target individual and a non-uniform crossover strategy. BSA's strategies is very powerful to control the magnitude of the search-direction matrix.

In recent years, many authors have examined that the combination of two algorithms can give better results compared with a single optimization algorithm. The combination of one meta-heuristic algorithm with another meta-heuristics algorithm is called a hybrid metaheuristic algorithm, growing interest in the field of hybrid meta-heuristics algorithm can be seen in the literature and some of its latest applications to a wide range of problems can be seen in the references (Kundra & Sood, 2010; Yildiz, 2013; Zhang et al., 2009; Nemati et al., 2009; Kao & Zahara, 2008; Shahla et al., 2011; Xiaoxia & Lixin, 2009; Nama et al., 2016a, 2016c; Nama & Saha, 2018a,b) for the last decade.

Since the success of an optimization algorithm depends on its significant development of exploration and exploitation abilities (Wolpert & Macready, 1997), in the proposed HBSA BSA is used to enhance the algorithm's exploitation ability and QA is used for exploration ability of the algorithm. Here, the simplified quadratic approximation (QA) with the three best points in the current population is used to reduce the computational burden and to improve the local search ability as well as the solution accuracy of the algorithm. These can maintain the algorithm's exploitation and exploration ability and at the same time can expedite its convergence. The remaining part is arranged in the following way: In Section 2 basic concept of the BSA and QA is described, in Section 3 the new method HBSA is presented. Section 4 empirically demonstrates the efficiency and accuracy of the hybrid approach in solving unconstrained optimization problems. For the validity of the proposed method, the obtained results are compared with five state-of-the-arts particle swarm optimization (PSO) variant approaches in terms of the numerical result of the solutions. However, utilizing experiences may make BSA converge slowly and prejudice exploitation on later iteration stage. The mutation operation in BSA introduces occasional changes of a random individual position with a specified mutation probability. However, the significance of amplitude control factor F in controlling BSA performance has not been acknowledged in BSA research. So the results are investigated for different value of the BSA control parameter (F). Finally, Section 5 summarizes the contribution of this paper along with some future research directions.

2. Overview of BSA and QA

In this section, the discussion of basic BSA and QA has been presented.

2.1. Basics of BSA

The stochastic evolutionary algorithm BSA is a population-based iterative evolutionary algorithm. BSA executes the search space into five major components: initialization, selection-I, mutation, crossover and selection-II.

Initialization:

At the initial stage, the initial population generates randomly within the uniform search space. The initial population is calculated according to the Eq. (1)

```

for i=1:PS
  for j=1:D
    Popi,j = Popmin + rand (0, 1)*(Popmax - Popmin);
  end
end

```

(1)

Here, PS is the population size, D is the dimension of the optimization problem, Pop is the initial population, Popmax and Popmin is the lower and upper bound of the population.

Selection-1:

BSA determines the historical population OldPop for calculating the search direction. The initial historical population is determined within the search boundary according to the following:

```
for i=1:PS
  for j=1:D
     $OldPop_{i,j} = Pop_{min} + rand(0, 1) \times (Pop_{max} - Pop_{min});$ 
  end
end
```

(2)

BSA has the option of redefining OldPop at the beginning of each iteration through the Eq. (3):

If rand < rand

```
 $OldPop = Pop;$ 
end
```

(3)

After OldPop is determined, Eq. (4) is used to randomly change the order of the individuals in OldPop:

```
 $OldPop = permuting(OldPop)$ 
```

(4)

Mutation:

During each generation BSA's mutation process generates the initial form of the trial population (Mutant) using Eq. (5)

```
 $Mutant = Pop + CF \times (OldPop - Pop)$ 
```

(5)

Here CF controls the amplitude of the search-direction matrix (OldPop - Pop). As the historical population is employed within the calculation of the search-direction matrix, BSA generates a trial population, taking partial advantage of its experiences from previous generations. The control parameter $F = 3 \times r_{ndn}$, where $r_{ndn} \in N(0, 1)$.

Crossover:

After the new mutation operation is finished, the crossover process generates the final form of the trial population T. The initial value of the trial population is Mutant, which has been set in the mutation process. Individuals with better fitness values for the optimization problem are used to evolve the target population. The first step of the crossover process calculates a binary integer-valued matrix (H) of size $N \times D$ that indicates the individuals of T to be manipulated using the relevant individuals of P. Then, the trial population T is updated as given by Algorithm 1.

Algorithm 1. The Crossover Strategy Algorithm.

```
If  $a < b$ ;  $(a, b) \in U(0, 1)$ 
for i from 1 to HP
  U= permuting (D);
  Map (i, mixrate×rand×D) = 0
else
for i from 1 to HP
```

```

    Map (i, randi (D)) = 0
  end
  for i from 1 to HP
    for j from 1 to D
  If map (i, j) = 1;
  T (i, j) = Pop (i, j);
  end
    end
  end
end
end
end

```

In Algorithm 1 two predefined strategies are randomly used in defining the integer-valued matrix, which is more complex than the processes used in DE. The first strategy uses mix rate M , and the other allows only one randomly chosen individual to mutate in each trial.

Selection-II:

In Selection-II stage, BSA updates the population (Pop) by comparing the trial population (T) with the corresponding population (Pop) based on a greedy selection. If the best individual of Pop (Pbest) has a better fitness value than the global minimum value obtained so far by BSA, the global minimizer is updated to be Pbest and the global minimum value is updated to be the fitness value of Pbest. The implementation step of the basic BSA algorithm is as follows:

- Step 1. Generalizes initial population, algorithm parameter;
- Step.2 Evaluate the fitness of each individual in the population;
- Step 3. Generate mutant population by Eq. (5)
- Step 4. By using Crossover Strategy given in Algorithm 1, generates the trial population.
- Step 5. Evaluate fitness vector of the trial population (offspring)
- Step 6. Select individual between target population and trial population (offspring) and update individual.
- Step 7. If the stopping criterion is not satisfied go to Step 3, else return the individual with the best fitness as the solution.

2.2. Quadratic Approximation (QA)

In this section, we discuss about three points Quadratic Approximations (Deep & Das, 2008). We choose randomly two distinct individuals $Pop_j = (pop_j^1, pop_j^2, pop_j^3, \dots, pop_j^D)$ and

$Pop_k = (pop_k^1, pop_k^2, pop_k^3, \dots, pop_k^D)$. Then another individual

$Pop_i = (pop_i^1, pop_i^2, pop_i^3, \dots, pop_i^D)$ is updated by approximate minimal point which is

calculated by using pop_j and pop_k . The approximate minimal point

$\overline{Pop}_i = (\overline{pop}_i^1, \overline{pop}_i^2, \overline{pop}_i^3, \dots, \overline{pop}_i^D)$ is calculated according to following Eq. (6)

$$\overline{Pop}_i^m = (0.5 \frac{((Pop_i^m)^2 - (Pop_j^m)^2)f_k + ((Pop_j^m)^2 - (Pop_k^m)^2)f_i + ((Pop_k^m)^2 - (Pop_i^m)^2)f_j}{(Pop_i^m - Pop_j^m)f_k + (Pop_j^m - Pop_k^m)f_i + (Pop_k^m - Pop_i^m)f_j}). \quad (6)$$

This has been successfully used in RST of Mohan and Shanker (1994). Here, the three-point quadratic approximation in the current population is used to reduce the computational burden and to improve the local search ability as well as the accuracy of the minimum function value of the algorithm.

3. Proposed Hybrid Backtracking Search Optimization Algorithm (HBSA)

In this section, a new hybrid method is introduced in detail. It has been already mention that for success of an optimization algorithm,, the significant development of the intensification i.e. exploitation and diversification i.e., exploration abilities of the algorithm is the most common factor (Wolpert & Macready, 1997). Diversification means the diverse solutions is generated within the search boundary so as to explore the search space of optimization problem on the global scale, while intensification indicates that the solution search in a local area by exploiting the information of current best solution in the region. The BSA mutation operator produce a final form of the trial population through their exploration ability. After producing the new population by BSA, QA is used to each population for reproducing the population. This can balance the exploration of the proposed algorithm. If one population violates the boundary condition the population is reflected back from the violated boundary using the following rule (Gong et al., 2010):

$$NewPop_{i,j} = \begin{cases} Pop_{min,j} + \text{rand}(0,1) * (Pop_{max,j} - Pop_{min,j}) & \text{if } Pop_{i,j} < Pop_{min,j} \\ Pop_{min,j} - \text{rand}(0,1) * (Pop_{max,j} - Pop_{min,j}) & \text{if } Pop_{i,j} > Pop_{max,j} \end{cases} \quad (8)$$

Here, $i = 1, 2, 3, \dots, PS$; Popmin represents the lower limit and Popmax represents the the upper limit in the search space of the i th population. So, combining the exploration of QA with the exploitation of BSA effectively, a hybrid BSA approach, called HBSA has been proposed.. In the proposed approach, the algorithm initialized with a population of random solutions and searches for optima by traveling into the search space. During this travel an evolution of this solution is performed by integrating BSA and QA. The description diagram of the proposed algorithm is shown in Fig. 1 and it is described as follows:

Step 1. Initialization:

Initialize algorithm parameter, population set and OldPop with random positions on D-dimensions in the problem space using Eqn. (1). Evaluate the fitness value for each individual of the population set.

Step 2. Setting OldPop:

Set the final form of OldPop for each target individual using Eq. (3) and Eq (4).

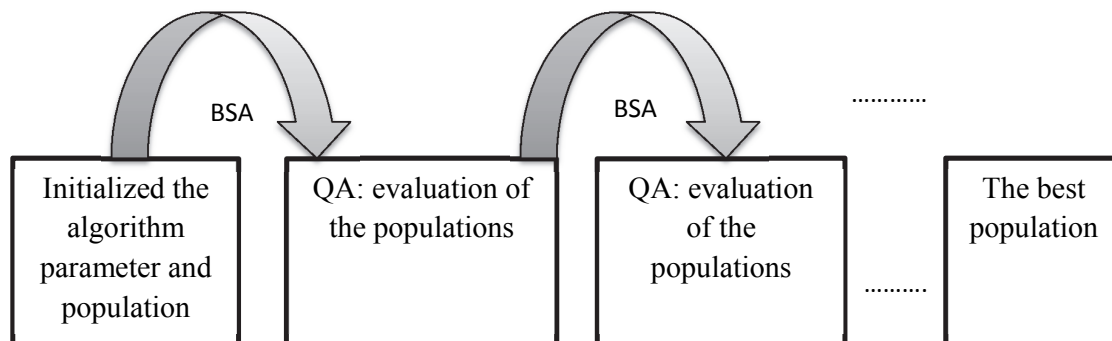


Fig.1. Diagram model of population's update

Step 3. Mutation:

Generate the trial population by Mutation operator using Eq. (5).

Step 4. Crossover:

By using Crossover strategy given in Algorithm 1, generate the final form of the trial population. Repair each infeasible the trial population to be feasible using Eqn. (8) and evaluate the fitness value for each trial population.

Step 5. Selection:

Select individual between trial population and target population. The fitness value of each trial population is utilized for selecting target population. In the selection, if the trial population has less or equal objective function value (in a minimization problem) than the corresponding target population, the trial population will replace the target population and enter the population of the next generation. Otherwise, the target population will remain in the population for the next generation.

Step 6. Update each individual by QA:

Update each individual by QA using eqn. (6) and repair the infeasible individuals of the population to be feasible using eqn. (8).

Step 7. Stopping Criteria:

This process is repeated from Step 3 until some specific stopping criterion is met. The pseudo code of the proposed Algorithm is shown in Fig. 2.

<p>Initialization: Initialize population size, stopping criterion (here in this paper function evaluations), mix rate parameter (mixrate), Control parameter (F) and range of design variables.</p> <p>Initial population: Generate the initial population and evaluate the fitness for each individual.</p> <p>Main loop: While stopping criteria is not meet Evaluation Set OldPop using Eq. (3) and Eq. (4). Update each individual of the population by Mutation operator using Eq. (5). By using Crossover strategy given in Algorithm 1, generate the final form of each individual of the trial population Repair the infeasible individuals of the population to be feasible using Eq. (8) Updated individual by QA using Eq. (6) Repair the infeasible individuals of the population to be feasible using Eq. (8)</p> <p>End While</p>
--

Fig. 2. The pseudo code of the proposed HBSA

4. Experimental Results and Discussion

4.1. Benchmark functions used

In order to compare the performance of the proposed HBSA, a set of 20 benchmark scalable test problems is selected from the literature. The problems include Unimodal and multimodal functions which are scalable and given in Table 1. The HBSA is implemented in MATLAB R2010a and the experiments are carried out on an Acer, 2.00 GHz machine with 500 MB RAM under Windows 7 platform. A total of 30 runs is conducted for the proposed HBSA, a different seed for the generation of random numbers to start a run. Since the significance of amplitude control factor F in controlling BSA performance has not been acknowledged in BSA research, the proposed HBSA experiment of various possible value of control parameter (F) of HBSA with keeping the population size fixed at 50 and the stopping criterion is a maximum of 150000 function evaluations.

4.2. Algorithms compared

In order to evaluate the merits of the proposed new mutation strategy, HBSA has been compared with five state-of-the-art PSO variants listed below:

- Unified particle swarm optimization scheme (UPSO) (Parsopoulos, 2004);
- Fully informed particle swarm (FIPS) (Mendes et al., 2004);

Table 1

Function used for checking the validity of the proposed method, SS: Search Space, D: Dimension=50, fmin=0

SL. No.	Function	Formulation	SS
F1	Sphere	$f(x) = \sum_{i=1}^D x_i^2$	[-100, 100]
F2	Schwefel2.22	$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	[-10, 10]
F3	Schwefel1.2	$f(x) = \sum_{i=1}^D \sum_{j=1}^i x_j^2$	[-100, 100]
F4	Schwefel2.21	$f(x) = \max\{ x_i , 1 \leq i \leq D\}$	[-100,100]

F5	Rosenbrock	$f(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30]
F6	Step	$f(x) = \sum_{i=1}^D (x_i + 0.5)^2$	[-100, 100]
F7	Quartic	$f(x) = \sum_{i=1}^D ix_i^4 + \text{random}(0,1)$	[-1.28, 1.28]
F8	Schwefel	$f(x) = 418 : 9829n - \sum x_i \sin(\sqrt{ x_i })$	[-500, 500]
F9	Rastrigin	$f(x) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$	[-5.12, 5.12]
F10	Ackley	$f(x) = 20 + e - 20e^{\frac{1}{D}(\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2})} - e^{\frac{1}{D}(\sum \cos(2\pi x_i))}$	[-32, 32]
F11	Griewank	$f(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) - 1$	[-600, 600]
F12	Penalized1	$f(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(3\pi y_{i+1})] \right\}$ $+ \sum_{i=1}^D u(x_i, 10, 100, 4)$ <p>where $y_i = 1 + \frac{1}{4}(x_i + 1)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$</p>	[-50, 50]
F13	Penalized2	$f(x) = 0.1 \left\{ 10 \sin^2(\pi x_i) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + 10 \sin^2(3\pi x_{i+1})] \right\}$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$	[-50, 50]
F14	Salomon	$f(x) = 1 - \cos(2\pi \sqrt{\sum_{i=1}^D x_i^2}) + 0.1 \sqrt{\sum_{i=1}^D x_i^2}$	[-100, 100]
F15	Zakharov	$f(x) = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D \frac{ix_i}{2}\right)^2 + \left(\sum_{i=1}^D \frac{ix_i}{2}\right)^2$	[-5.12, 5.12]
F16	Axis parallel hyper ellipsoid	$f(x) = \sum_{i=1}^D ix_i^2$	[-5.12, 5.12]
F17	Ellipsoidal	$f(x) = \sum_{i=1}^D (x_i - i)^2$	[-100, 100]
F18	Cigar	$f(x) = x_1^2 + 100000 \sum_{i=2}^D x_i^2$	[-10, 10]
F19	Exponential	$f(x) = 1 - \exp(-0.5 \sum_{i=1}^D x_i^2)$	[-1, 1]

$$f(x) = 00.1D + \sum_{i=1}^D x_i^2 - 0.1 \sum_{i=1}^D \cos(5\pi x_i)$$

- Fitness-distance-ratio based particle swarm optimization (FDR-PSO) (Peram et al., 2003);
- Cooperative approach to particle swarm optimization (CPSO-H) (van den Bergh and Engelbrecht 2004);
- Comprehensive Learning Particle Swarm Optimizer (CLPSO) (Liang et al., 2006).

4.3. Results on numerical benchmarks

Table 2 shows the experimental result of the mean and the standard deviation of the best-of-run errors for 30 independent runs of each of the five algorithms on twenty numerical benchmark problems for dimension $D = 50$. Note that the experimental results have been performed and best-of-the-run error corresponds to the absolute difference between the best-of-the-run value $f(X_{best})$ and the actual optimum $f(X^*)$ of a particular objective function i.e. $abs(f(X_{best}) - f(X^*))$. The best results are marked in bold for all problems. Table 2 indicates that out of 20 in 13 cases HBSA could beat all other competitor algorithms. It is seen that for function F8 HBSA superior than FDR-PSO, FIPS, CPSO-H, for F9 HBSA superior than FDR-PSO, FIPS, UPSO, for F10 HBSA superior than FDR-PSO, FIPS, CPSO-H, for F11 HBSA superior than FDR-PSO, FIPS, UPSO, CLPSO and for function F17 HBSA superior than FIPS, UPSO, CLPSO and CPSO-H respectively. As the comparison result given in Table 2 may be concluded that the proposed method HBSA perform better than other algorithms.

Table 2

Comparison of statistical result of HBSA of error function value with different state of the art PSO variants

F	FDR-PSO	FIPS	UPSO	CLPSO	CPSO-H	HBSA
F1	2.99e-031(1.53e-030)	5.65e-001(1.85e-001)	4.42e-031(4.17e-031)	1.17e-004(3.15e-005)	4.99e-004(2.29e-004)	1.78e-045(2.77e-045)
F2	1.31e-012(4.82e-012)	1.51e-001(2.21e-002)	4.52e-019(2.62e-019)	1.74e-003(2.75e-004)	7.85e-003(1.59e-003)	7.39e-026(6.64e-026)
F3	1.82e-030(7.20e-030)	1.00e+001(1.71e+000)	6.81e-030(5.67e-030)	2.57e-003(6.84e-004)	1.41e-002(1.10e-002)	2.52e-043(5.96e-043)
F4	3.11e+000(8.62e-001)	1.04e+001(8.35e-001)	2.38e+000(8.17e-001)	2.21e+001(1.42e+000)	2.22e+001(5.00e+000)	1.31e-005(7.69e-006)
F5	5.32e+001(3.04e+001)	2.88e+002(5.96e+001)	5.47e+001(3.14e+001)	1.77e+002(4.65e+001)	5.29e+001(5.09e+001)	6.25e+001(2.90e+001)
F6	3.20e+000(1.94e+000)	0.00e+000(0.00e+000)	0.00e+000(0.00e+000)	0.00e+000(0.00e+000)	0.00e+000(0.00e+000)	4.27e+000(1.80e+000)
F7	1.56e-002(5.46e-003)	4.68e-002(7.85e-003)	3.93e-002(1.18e-002)	2.81e-002(4.74e-003)	4.33e-002(1.64e-002)	8.09e-003(2.48e-003)
F8	5.69e+003(1.33e+003)	3.81e+003(1.48e+003)	3.47e-004(6.21e-004)	1.15e-002(1.38e-002)	9.08e+001(1.19e+002)	1.03e+000(8.42e-001)
F9	7.70e+001(1.36e+001)	2.56e+002(1.75e+001)	1.29e+002(2.02e+001)	3.77e+000(1.41e+000)	2.01e-004(1.11e-004)	9.92e+000(3.55e+000)
F10	1.22e-001(3.76e-001)	1.82e-001(3.51e-002)	4.77e-004(2.61e-003)	5.94e-003(1.31e-003)	5.15e-003(1.50e-003)	5.67e-014(1.43e-014)
F11	5.42e-003(8.41e-003)	5.12e-001(7.73e-002)	2.47e-004(1.35e-003)	4.60e-004(1.45e-004)	1.13e-002(1.65e-002)	4.02e-003(6.45e-003)
F12	1.02e-001(1.82e-001)	3.59e-001(1.13e-001)	1.17e-001(3.49e-001)	2.09e-002(7.42e-003)	3.93e-006(3.69e-006)	1.09e-002(2.68e-002)
F13	4.03e-003(5.39e-003)	3.19e-001(7.96e-002)	7.32e-004(2.79e-003)	6.73e-005(2.13e-005)	2.89e-005(3.24e-005)	7.10e-018(1.96e-017)
F14	6.53e-001(1.11e-001)	8.60e-001(7.34e-002)	6.60e-001(1.43e-001)	7.54e-001(1.43e-001)	2.45e+000(5.38e-001)	4.50e-001(7.31e-002)
F15	6.67e-002(7.23e-002)	8.48e+001(1.37e+001)	7.13e+000(2.78e+000)	6.84e+001(1.16e+001)	2.23e+002(3.27e+001)	9.02e-004(1.87e-003)
F16	9.84e-030(5.39e-029)	2.70e-002(5.99e-003)	2.34e-032(2.93e-032)	7.06e-006(1.61e-006)	3.94e-005(1.81e-005)	4.02e-046(1.14e-045)
F17	1.47e-028(2.04e-028)	5.16e-001(1.25e-001)	1.50e-031(4.03e-031)	1.43e-004(3.71e-005)	4.37e-004(2.84e-004)	1.19e-018(8.03e-019)
F18	4.04e-028(1.25e-027)	4.14e+002(8.62e+001)	4.05e-028(5.40e-028)	1.06e-001(2.03e-002)	4.98e-001(2.70e-001)	3.11e-042(7.85e-042)
F19	5.92e-016(2.71e-016)	2.73e-005(9.58e-006)	1.07e-016(2.03e-017)	5.69e-009(1.18e-009)	2.95e-008(1.85e-008)	4.29e-016(1.99e-016)
F20	1.23e-001(1.51e-001)	6.76e-004(1.53e-004)	1.48e-001(2.74e-001)	1.62e-007(5.31e-008)	5.53e-007(2.83e-007)	2.16e-015(2.21e-015)

4.4. A parametric study on HBSA

Table 3 shows the mean and the standard deviation of the error function value of different problem for 30 independent runs and for $D = 50$. In Table 3 the results have been executed with different values of control parameter F which are taken as 0.5, 0.6, 0.7, 0.8 and 0.9, respectively. From this table it is seen that, for F2, F7, F13, and F17 HBSA give the best result for control parameter $F=0.8$, respectively. The best result is obtained for F8 with the value of control parameter $F = 0.5$ and for others function the value of control parameter $F = 0.9$. So, it is clear that the performance of the algorithm gradually becomes better with increase of F up to 0.9. From the parametric study it is clear that the algorithm performs best when $F = 0.9$ for all the numerical benchmark functions. Table 4 shows the comparative study of HBSA with the basic BSA for the value of control parameter $F = 3*\text{rndn}$ and 0.9, from this

table it is seen that, except for function F6, F11 and F12 HBSA perform better than the BSA. Fig. (a) - (f), shows the convergence graph for different value of F with the proposed algorithm. From the figures (Fig. (a) - (f)) it is clear that the converges speed vary with the variation of F. So, F is an important parameter of the HBSA and $F = 0.9$ is the better choice in this case. We do not claim that these parameter settings are the best for any problem in general, but these values are recommended since they are found to be repeat giving good results for most of the problems and hence they are an appropriate setting to choose if we talk about the overall performance of the algorithm.

Table 3

Comparison of statistical result of HBSA of error function value for different value of control parameter(F)

F	HBSA(F=0.5)	HBSA(F=0.6)	HBSA(F=0.7)	HBSA(F=0.8)	HBSA(F=0.9)
F1	3.33e-017(3.93e-017)	4.14e-024(7.77e-024)	4.91e-033(7.32e-033)	2.26e-041(3.23e-041)	1.78e-045(2.77e-045)
F2	3.74e-012(3.92e-012)	3.63e-017(2.05e-017)	9.15e-023(9.16e-023)	1.72e-026(1.46e-026)	7.39e-026(6.64e-026)
F3	7.64e-016(9.01e-016)	9.97e-023(1.88e-022)	1.17e-031(1.98e-031)	1.61e-040(2.69e-040)	2.52e-043(5.96e-043)
F4	5.09e-002(2.79e-001)	2.25e-001(3.00e-001)	8.23e-004(3.96e-004)	5.51e-005(2.58e-005)	1.31e-005(7.69e-006)
F5	1.06e+002(5.76e+001)	8.78e+001(3.38e+001)	8.97e+001(4.63e+001)	6.82e+001(3.56e+001)	6.25e+001(2.90e+001)
F6	1.00e+002(5.18e+001)	3.94e+001(1.35e+001)	1.87e+001(5.35e+000)	9.73e+000(4.50e+000)	4.27e+000(1.80e+000)
F7	5.28e-002(1.53e-002)	3.39e-002(9.41e-003)	1.80e-002(4.92e-003)	1.03e-002(2.93e-003)	8.09e-003(2.48e-003)
F8	1.03e+000(1.04e+000)	1.14e+000(1.17e+000)	1.13e+000(9.59e-001)	1.22e+000(1.03e+000)	1.03e+000(8.42e-001)
F9	4.04e+001(3.99e+001)	2.93e+001(9.17e+000)	2.40e+001(3.77e+000)	2.92e+001(5.99e+000)	9.92e+000(3.55e+000)
F10	2.88e+000(7.84e-001)	1.69e+000(5.86e-001)	2.03e-001(5.41e-001)	9.41e-014(1.83e-014)	5.67e-014(1.43e-014)
F11	2.63e-002(4.52e-002)	5.90e-003(9.50e-003)	5.17e-003(8.77e-003)	6.73e-003(9.06e-003)	4.02e-003(6.45e-003)
F12	2.82e-001(7.54e-001)	5.42e-002(1.00e-001)	3.15e-002(6.48e-002)	2.91e-002(5.49e-002)	1.09e-002(2.68e-002)
F13	1.38e-001(7.28e-001)	2.93e-003(1.12e-002)	1.34e-014(3.33e-014)	8.38e-022(4.58e-021)	7.10e-018(1.96e-017)
F14	2.84e+000(3.91e-001)	1.90e+000(3.13e-001)	8.60e-001(1.45e-001)	4.90e-001(8.45e-002)	4.50e-001(7.31e-002)
F15	1.67e+000(9.40e-001)	9.86e-001(5.21e-001)	7.22e-002(4.67e-002)	1.10e-003(5.94e-004)	9.02e-004(1.87e-003)
F16	1.58e-018(4.11e-018)	1.37e-025(2.31e-025)	5.03e-034(1.05e-033)	6.39e-043(9.26e-043)	4.02e-046(1.14e-045)
F17	7.06e-011(2.26e-011)	7.22e-011(2.15e-010)	5.47e-013(6.25e-013)	1.49e-022(1.07e-022)	1.19e-018(8.03e-019)
F18	5.85e-014(1.44e-013)	1.97e-021(2.57e-021)	6.65e-030(1.62e-029)	4.40e-039(8.35e-039)	3.11e-042(7.85e-042)
F19	1.01e-014(6.87e-015)	4.09e-015(1.84e-015)	1.86e-015(4.93e-016)	8.84e-016(2.85e-016)	4.29e-016(1.99e-016)
F20	5.14e-002(1.27e-001)	3.09e-014(2.61e-014)	1.17e-014(3.69e-015)	5.03e-015(2.27e-015)	2.16e-015(2.21e-015)

4.5. Statistical results of tests

For statistical analysis, a pairwise comparison of a problem-based or multi-problem-based statistical comparison method (Derrac et al., 2011) has been applied to justify the performance of the proposed method. In multi-problem-based pairwise comparison, the mean of the optimum results calculated in different independent runs are used to obtained the which method are more appropriate than the other compared algorithm. In this study, the average of global minimum values obtained as the result of 30 runs for its Wilcoxon signed rank test multi problem based comparison of the algorithms.

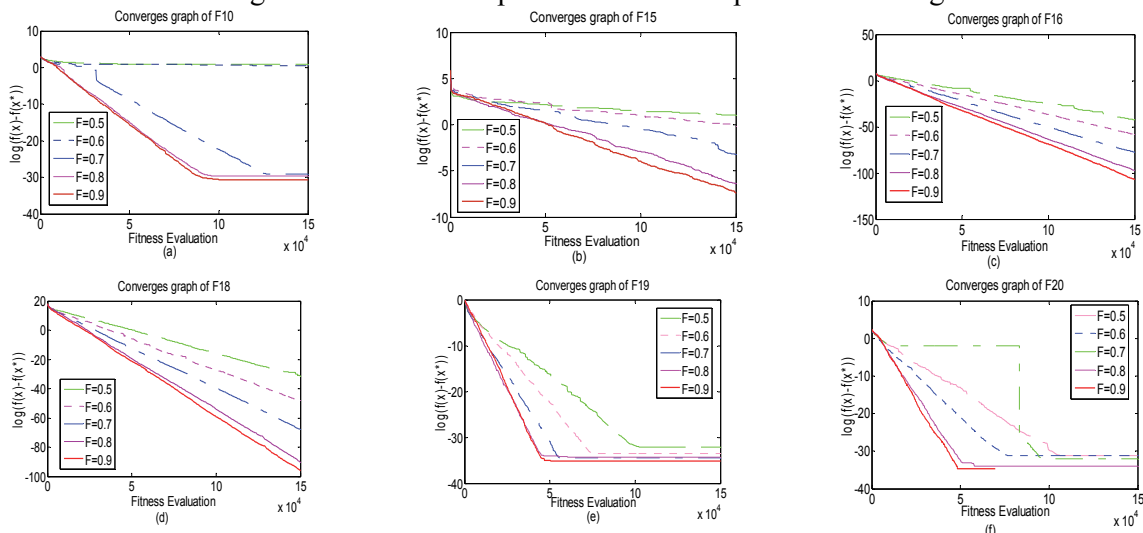


Fig. 3. Convergence graphs of six different benchmark function: (a) F10, (b) F15, (c) F16, (d) F18, (e) F19, (f) F20

Table 5 presents the multi-problem-based pairwise statistical comparison results using the averages of the global minimum values obtained through 30 runs of HBSA and the comparison algorithms to solve the benchmark problems. These results show that HBSA was statistically more successful than all of the comparison algorithms, with a statistical significance value $\alpha = 0.05$.

Table 4

Comparison of statistical result of error function value with BSA (For different value of control parameter)

F	BSA(F=3* π)	BSA(F=0.9)	HBSA(F=3* π)	HBSA(F=0.9)
F1	6.58e-009(3.71e-009)	7.75e+002(3.80e+002)	5.91e-027(6.95e-027)	1.78e-045(2.77e-045)
F2	1.19e-005(5.85e-006)	6.98e+000(1.72e+000)	1.91e-015(1.56e-015)	7.39e-026(6.64e-026)
F3	1.95e-007(2.21e-007)	1.85e+004(1.06e+004)	1.91e-025(1.81e-025)	2.52e-043(5.96e-043)
F4	5.85e+000(1.14e+000)	2.65e+001(4.44e+000)	5.73e-004(2.79e-004)	1.31e-005(7.69e-006)
F5	1.08e+002(4.27e+001)	1.58e+005(2.24e+005)	7.41e+001(3.99e+001)	6.25e+001(2.90e+001)
F6	0.00e+000(0.00e+000)	7.03e+002(2.63e+002)	2.30e+000(1.60e+000)	4.27e+000(1.80e+000)
F7	2.73e-002(9.28e-003)	1.59e-001(9.90e-002)	1.50e-002(4.56e-003)	8.09e-003(2.48e-003)
F8	6.93e+002(2.50e+002)	3.12e+003(4.48e+002)	1.48e+000(1.53e+000)	1.03e+000(8.42e-001)
F9	1.95e+001(4.56e+000)	2.06e+001(5.39e+000)	4.81e+001(5.85e+000)	9.92e+000(3.55e+000)
F10	3.55e-005(3.17e-005)	5.37e+000(1.02e+000)	7.90e-014(1.09e-014)	5.67e-014(1.43e-014)
F11	5.77e-004(2.21e-003)	7.16e+000(6.95e+000)	2.14e-003(4.65e-003)	4.02e-003(6.45e-003)
F12	3.13e-009(3.42e-009)	6.88e+000(2.07e+001)	1.02e-002(2.66e-002)	1.09e-002(2.68e-002)
F13	5.53e-010(6.39e-010)	1.00e+004(5.48e+004)	1.59e-008(8.16e-008)	7.10e-018(1.96e-017)
F14	1.17e+000(1.62e-001)	3.84e+000(6.32e-001)	6.77e-001(1.30e-001)	4.50e-001(7.31e-002)
F15	1.20e+001(2.79e+000)	3.60e+001(6.95e+000)	8.83e-002(5.36e-002)	9.02e-004(1.87e-003)
F16	5.72e-010(4.76e-010)	3.84e+001(1.52e+001)	3.39e-028(3.16e-028)	4.02e-046(1.14e-045)
F17	2.97e-008(2.70e-008)	1.51e+003(9.86e+002)	4.58e-015(2.90e-015)	1.19e-018(8.03e-019)
F18	6.82e-006(6.29e-006)	5.06e+005(2.17e+005)	6.26e-024(1.23e-023)	3.11e-042(7.85e-042)
F19	4.10e-013(3.35e-013)	3.12e-002(1.48e-002)	5.88e-016(1.40e-016)	4.29e-016(1.99e-016)
F20	2.56e-011(3.38e-011)	4.62e-001(2.29e-001)	2.63e-015(1.66e-015)	2.16e-015(2.21e-015)

Table 5

Multi-problem based statistical pairwise comparison of comparison algorithms and HBSA ($\alpha = 0.05$)

Algorithm vs. HBSA	p-Value	R+	R-	Winner
FDR vs. HBSA	0.014	39	171	HBSA
FIPS vs. HBSA	0.01	13	197	HBSA
UPSO vs. HBSA	0.247	74	136	HBSA
CLPSO vs. HBSA	0.073	75	153	HBSA
CPSO-H vs. HBSA	0.079	78	152	HBSA
BSA vs. HBSA	0.012	38	172	HBSA

An examination of the results obtained from the tests which is given in Table 5 reveals that the success of HBSA in solving the unconstrained optimization problems is better than the other comparison algorithms.

5. Conclusions

A new hybrid global search algorithm has been proposed by combining BSA with QA. For the balance of the exploration and the exploitation of BSA, in this paper, we have proposed a hybrid BSA approach, called HBSA, which combines the exploration of QA with the exploitation of BSA. The simplified quadratic approximation (QA) can improve the accuracy of the solution and also enhances the local search ability of the algorithm. Since the hybrid global search algorithms have a good trade-off between the exploration and the exploitation, it makes our proposed HBSA approach very effective and efficient. To verify the performance of HBSA, 20 benchmark functions have been chosen from literature. Experimental results were compared with five PSO variant's given in Table 2. For different values of control parameter, the result was also executed which were given in Table 3. For the multi-problem-based pairwise statistical comparison of HBSA and the other algorithms, the average values of the solutions obtained in Table 2 were used. Table 5 shows the p value and R+ and R- values obtained in this comparison. Analysis of these values when $\alpha = 0.05$ shows that HBSA was statistically more successful than all of the comparison algorithms. Finally, the overall experimental results demonstrate the good performance of HBSA. For the future work, this method can also apply to solve the

constrained optimization problem, multiobjective optimization problem and to solve the various complex problem in the field of engineering.

Acknowledgement

The authors would like to thank Dr. P. N. Suganthan for providing the source codes of UPSO, Fully informed particle swarm (FIPS), FDR-PSO, CPSO-H, and CLPSO.

References

- Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.
- Beyer, H. G., & Arnold, D. V. (2001). Theory of evolution strategies—A tutorial. In *Theoretical aspects of evolutionary computing* (pp. 109-133). Springer, Berlin, Heidelberg.
- Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, 219(15), 8121-8144.
- Deep, K., & Das, K. N. (2008). Quadratic approximation based hybrid genetic algorithm for function optimization. *Applied Mathematics and Computation*, 203(1), 86-98.
- Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3-18.
- Erol, O. K., & Eksin, I. (2006). A new optimization method: big bang–big crunch. *Advances in Engineering Software*, 37(2), 106-111.
- 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.
- Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers & Structures*, 89(23-24), 2325-2336.
- Gandomi, A. H., Talatahari, S., Yang, X. S., & Deb, S. (2013). Design optimization of truss structures using cuckoo search algorithm. *The Structural Design of Tall and Special Buildings*, 22(17), 1330-1349.
- 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. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mechanica*, 213(3-4), 267-289.
- Kao, Y. T., & Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing*, 8(2), 849-857.
- Kennedy, J., & Eberhart, R. C. 1995. Particle Swarm Optimization. In *IEEE International Conference on Neural Networks (Perth, Australia)*, IEEE Service Center, Piscataway, NJ (pp. 1942-1948).
- Kundra, H., & Sood, M. (2010). Cross-country path finding using hybrid approach of PSO and BBO. *International Journal of Computer Applications*, 7(6), 15-19.
- Li, X., Zhang, J., & Yin, M. (2014). Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural Computing and Applications*, 24(7-8), 1867-1877.
- 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 on evolutionary computation*, 10(3), 281-295.
- Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, 188(2), 1567-1579.
- Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: simpler, maybe better. *IEEE transactions on evolutionary computation*, 8(3), 204-210.

- Mohan, C., & Shanker, K. (1994). A controlled random search technique for global optimization using quadratic approximation. *Asia-Pacific Journal of Operational Research*, 11(1), 93-101.
- Nama, S., Saha, A., & Ghosh, S. (2016a). 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. (2016b). Improved symbiotic organisms search algorithm for solving unconstrained function optimization. *Decision Science Letters*, 5(3), 361-380.
- Nama, S., Saha, A. K., & Ghosh, S. (2017c). A hybrid symbiosis organisms search algorithm and its application to real world problems. *Memetic Computing*, 9(3), 261-280.
- Nama, S., Saha, A. K., & Ghosh, S. (2017d). Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c- Φ backfill. *Applied Soft Computing*, 52, 885-897.
- Nama, S., & Saha, A. (2018a). An ensemble symbiosis organisms search algorithm and its application to real world problems. *Decision Science Letters*, 7(2), 103-118.
- Nama, S., Saha, A.K., (2018b). A new hybrid differential evolution algorithm with self-adaptation for function optimization. *Applied Intelligence*, 48, 1657-1671.
- Parsopoulos, K. E. (2004). UPSO: A unified particle swarm optimization scheme. *Lecture Series on Computer and Computational Science*, 1, 868-873.
- Peram, T., Veeramachaneni, K., & Mohan, C. K. (2003, April). Fitness-distance-ratio based particle swarm optimization. In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE* (pp. 174-181). IEEE.
- Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, 13(5), 2592-2612.
- Nemati, S., Basiri, M. E., Ghasem-Aghae, N., & Aghdam, M. H. (2009). A novel ACO-GA hybrid algorithm for feature selection in protein function prediction. *Expert systems with applications*, 36(10), 12086-12094.
- Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), 702-713.
- Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.
- Van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE transactions on evolutionary computation*, 8(3), 225-239.
- 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., & Press, L. (2010). *Nature-Inspired Metaheuristic Algorithms Second Edition*.
- Yildiz, A. R. (2013). Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations. *Applied Soft Computing*, 13(3), 1433-1439.
- Zhang, X., & Tang, L. (2009). A new hybrid ant colony optimization algorithm for the vehicle routing problem. *Pattern Recognition Letters*, 30(9), 848-855.
- Zhang, G., Shao, X., Li, P., & Gao, L. (2009). An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 56(4), 1309-1318.

