Expert Systems with Applications 42 (2015) 6686–6698

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

TSA: Tree-seed algorithm for continuous optimization

Mustafa Servet Kiran *

Department of Computer Engineering, Faculty of Engineering, University of Selcuk, 42075 Konya, Turkey

ARTICLE INFO

Article history:
Available online 4 May 2015

Keywords:
Heuristic search
Tree and seed
Numeric optimization
Multilevel thresholding

ABSTRACT

This paper presents a new intelligent optimizer based on the relation between trees and their seeds for continuous optimization. The new method is in the field of heuristic and population-based search. The location of trees and seeds on *n*-dimensional search space corresponds with the possible solution of an optimization problem. One or more seeds are produced from the trees and the better seed locations are replaced with the locations of trees. While the new locations for seeds are produced, either the best solution or another tree location is considered with the tree location. This consideration is performed by using a control parameter named as search tendency (ST), and this process is executed for a pre-defined number of iterations. These mechanisms provide to balance exploitation and exploration capabilities of the proposed approach. In the experimental studies, the effects of control parameters on the performance of the method are firstly examined on 5 well-known basic numeric functions. The performance of the proposed method is also investigated on the 24 benchmark functions with 2, 3, 4, 5 dimensions and multilevel thresholding problems. The obtained results are also compared with the results of state-of-art methods such as artificial bee colony (ABC) algorithm, particle swarm optimization (PSO), harmony search (HS) algorithm, firefly algorithm (FA) and the bat algorithm (BA). Experimental results show that the proposed method named as TSA is better than the state-of-art methods in most cases on numeric function optimization and is an alternative optimization method for solving multilevel thresholding problem.

1. Introduction

Optimization is a task performed to find the best solution amongst the possible solutions for an optimization problem with regard to some criteria. Optimization deals with optimization problems, and can contain maximization or minimization processes. If S is a search space, F, $F \subseteq S$, is a set of acceptable solutions of S and f is an objective function, minimization or maximization is to find $\vec{x} \in F$ in Eqs. (1) and (2), those define minimization and maximization processes, respectively.

$$f(\vec{x}) \leqslant f(\vec{y}) \quad \forall \vec{y} \in F \tag{1}$$

$$f(\vec{x}) \geqslant f(\vec{y}) \quad \forall \vec{y} \in F \tag{2}$$

If the function to be optimized is non-continuous or non-differentiable, or optimizing the function is computationally expensive for high dimensional search space, heuristic search methods can be applied to optimize it.

The expert or intelligent systems aim to present a solution methodology to solve this type of problems (nonlinear or non-differentiable optimization problems). Over the last decades, a growing popularity of population-based intelligent search methods has been occurred due to their successes in solving the optimization problems, which are easy adaptation and implem entation for the optimization problem and low computation costs. These techniques use biological or physical phenomena in local interactions amongst the agents while solving the optimization problem. Genetic algorithm (GA) (Goldberg, 1989;Holland, 1992) has been developed by considering Darwinian evolutionary theory. In GA, crossover and mutation mechanisms are used to produce a new generation. Ant colony optimization (ACO) (Dorigo, Maniezzo, & Colorni, 1996) mimics the behavior of real ants between the nest and food source. Basically, ACO is proposed to solve discrete optimization problems and the main concept is based on pheromone mechanism to construct the solutions. Particle swarm optimization (PSO) (Kennedy & Eberhart, 1995) has been inspired by cognitive and social behaviors of fish or birds. The solution update rule in PSO uses subtraction operation between the current solution and the best (the best of population and the best solution obtained so far by the particle) solutions to produce a new possible solution. Harmony search (HS) (Geem, Kim, & Loganathan, 2001) algorithm simulates the improvisation process of musicians. Bat algorithm (BA) (Yang, 2010b) imitates the echolocation behavior of bats.

* Tel.: +90 332 223 1992; fax: +90 332 223 2106.
E-mail addresses: mskiran@selcuk.edu.tr, mustafaservetkiran@gmail.com

http://dx.doi.org/10.1016/j.eswa.2015.04.055
0957-4174/

Firefly algorithm (FA) (Yang, 2010a) has been inspired by the flashing behavior of fireflies. Artificial bee colony algorithm (ABC) (Karaboga & Basturk, 2007) simulates foraging and information sharing behaviors of bees in a hive. In ABC, the same subtraction-based update rule is used in both employed bee phase and onlooker bee phase of the algorithm, and the best solution obtained so far is not used in this algorithm. As well as organisms' behaviors, some methods modeled physical laws for solving optimization problems. Simulated Annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983) has been inspired by thermodynamic effects, and gravitational search algorithm (GSA) (Rashedi, Nezamabadi-Pour, & Saryazdi, 2009) is based on the Newtonian laws of gravity. In GSA, all the solutions are used to produce a new solution by using the Newtonian laws of gravity.

In the present study, it aims to establish a new optimizer named as tree–seed algorithm (TSA) to solve continuous optimization problems. In the new proposed method, two aspects, exploration and exploitation, are considered to overcome the characteristics of the optimization problems. For exploration, the method is designed as population-based, the search starts from multiple-points on the search space of the problem, and randomness is included in local interactions amongst the agents. These agents in the proposed method are called as *trees*. In order to compensate exploitation, *seeds* are used. The number of seeds produced from a tree is randomly obtained. The locations of trees and seeds on n-dimensional search space correspond to the possible solutions of the optimization problem, and while a new seed is produced from a tree, either another tree location or the best tree (the best solution obtained so far) location is considered with its own location. The search tendency to the best or randomly selected tree is handled with a control parameter named as *search tendency* (*ST*) and this control in the TSA provides a local intensification and convergence to the optimum or near optimum for the problem.

The paper is organized as follows: the study is introduced in Section 1, and the main contribution of the study is presented in this section. TSA method is detailed in Section 2, experimental studies and comparisons are conducted in Section 3, the obtained results are reported and discussed in Section 4 and finally, the study is concluded and future directions are given in Section 5.

1.1. Main contribution

In this study, a novel population-based iterative search algorithm is developed for solving continuous optimization problems. In the present study, two mechanisms are merged to balance exploration and exploitation capabilities of the method. First mechanism aims to improve exploration capability of the method by using search tendency parameter. The new solutions are produced by considering the current solution and the best or randomly selected solution with this parameter. The second mechanism focuses on improving the exploitation of the method. In this mechanism, more than one solution is created around a solution. Therefore, the search around the found solutions is improved by using the second search mechanism. Based on the literature review of the study, these mechanisms show the novelty and differences from the approaches exist in the literature.

2. TSA: Tree–seed algorithm

The natural phenomena in TSA is the relationship between trees and their seeds. In nature, trees spread to the surface through their seeds. These seeds grow over time and new trees become from these seeds. If we assume that the surface of these trees as a search space for the optimization problem, the location of trees and seeds can be considered as possible solutions for the optimization problem. To obtain a location of a seed that will be produced from a tree is important for the optimization problem because this process constitutes the core of search. We propose two search equations for this process. The first equation (Eq. (1)) considers the tree location that the seed will be produced for this tree and the best location of the tree population. This search equation also improves the local search or intensification capability of the proposed algorithm. The second update rule (Eq. (2)) uses two different tree locations for producing a new seed for the tree.

$$S_{i,j} = T_{i,j} + \alpha_{i,j} \times (B_j - T_{r,j}) \quad (3)$$

$$S_{i,j} = T_{i,j} + \alpha_{i,j} \times (T_{i,j} - T_{r,j}) \quad (4)$$

where, $S_{i,j}$ is jth dimension of ith seed that will be produced ith tree, $T_{i,j}$ is the jth dimension of ith tree, B_j is the jth dimension of best tree location obtained so far, $T_{r,j}$ is the jth dimension of rth tree randomly selected from the population, α is the scaling factor randomly produced in range of $[-1, 1]$ and i and r are different indices.

The most important point is which equation will be selected to produce a new seed location. This selection is controlled by a control parameter of the method named as search tendency (ST) in range of [0, 1]. The higher value of ST provides a powerful local search and speed convergence, the lower value of ST causes slow convergence but powerful global search. In other words, the exploration and exploitation capabilities of the TSA are controlled by ST parameter.

In the beginning of search with TSA, the initial tree locations which are possible solutions for the optimization problem are produced by using Eq. (5).

$$T_{i,j} = L_{j,\min} + r_{i,j}(H_{j,\max} - L_{j,\min}) \quad (5)$$

where, $L_{j,\min}$ is the lower bound of the search space, $H_{j,\max}$ is the higher bound of the search space and $r_{i,j}$ is a random number produced for each dimension and location, in range of [0, 1].

For minimization, the best solution is selected from the population using Eq. (6).

$$B = \min\{f(\vec{T}_i)\} i = 1, 2, \ldots, N \quad (6)$$

where, N is the number of trees in the population.

While the new seed locations are generated for a tree, the number of seeds can be more than one and this number depends on the population size. In the analysis of effects of control parameters to the performance of TSA, 10% of population size is the minimum number of seeds produced for a tree and 25% of the population size is the maximum number of seeds produced from a tree. The number of seed production is completely random in TSA.

The detailed algorithmic framework of the TSA is given in Fig. 1. In the algorithm, it is seen how to use the control parameter ST in TSA. If randomly produced number (rand in Fig. 1) in range of [0, 1] is less than ST, Eq. (3) is used for updating the dimension, otherwise Eq. (4) is used.

3. Experimental studies

3.1. Performance analysis and comparisons on the benchmark functions

In the experiments, the effects of control parameters to the performance of TSA are analyzed and then the performance of TSA with the selected control parameters is compared with the results of PSO, ABC, HS, FA, BA methods. In the analyses, TSA is applied to solve five basic benchmark functions (F1, F10, F11, F13, F15 in the Table 1) to obtain the best control parameters. The methods are run on the 2,3,4,5 dimensional 24 numeric functions given in Table 1 to compare with the other methods. For all experiments, the termination condition is selected as the maximum number of

```
Step 1. The initialization of the algorithm.
    Set the number of population size (N).
    Set the ST parameter for the method.
    Set the dimensionality of the problem (D).
    Decide the termination condition
    Generate N random tree location on the D-dimensional search space using Eq. 5 (T).
    Evaluate the tree location using objective function specified for the problem.
    Select the best solution (B) using Eq.6.
Step 2. Searching with Seeds
    FOR all trees
        Decide the number of seeds produced for this tree.
        FOR all seeds
            FOR all dimensions
                IF(rand<ST)
                    Update this dimension using Eq. 3 (S)
                ELSE
                    Update this dimension using Eq. 4 (S)
                END IF
            END FOR
        END FOR
        Select the best seed and compare it with the tree.
        If the seed location is better than tree location, the seed substitutes for this tree.
    END FOR
Step 3. Selection of Best Solution
    Select the best solution of population using Eq. 7.
    If new best solution is better than the previous best solution, new best solution
    substitutes for the previous best solution.
Step 4. Testing Termination condition
    If termination condition is not met, go to Step 2.
Step 5. Reporting
    Report the best solution.
```

Fig. 1. The algorithmic framework of the proposed algorithm.

function evaluations (Max_FEs) and it is set by using the dimensionality of the function given in Eq. (7) as given in (Suganthan et al., 2005)

$$Max\_FEs = D \times 10,000 \tag{7}$$

The numeric benchmark functions used in the experiments have some characteristics. If a function has more than one local optimum, this function is called as multimodal (M) and these types of functions test the global search ability of the algorithm. Unimodal functions (U) have only one local optimum, and it is global optimum. The exploitation ability of the algorithm is examined on these types of problems. If a function with *n*-variable can be written as the sum of the *n* functions of one variable, then this function is called as a separable (S) function. Non-separable functions cannot be written in this form because there is interrelation amongst variables of these functions. Some functions have flatness on the surface and solving such functions is difficult for algorithms because these functions do not give any direction information to the methods. If the global optimum of the function is in the narrow curving valley Rosenbrock's Banana function, the methods should keep up with the direction changes in the functions(Kiran & Gunduz, 2013). The dimensionality of the function is another issue for the method because the search space of the problem increases exponentially. Therefore, these benchmark functions are selected to investigate the performance of the proposed method.

3.1.1. The analysis of control parameters of TSA

The control parameters of TSA are accepted as population size and ST. The population size is selected as 10, 20, 30, 40 and 50, and ST parameter is selected as 0.1, 0.2, 0.3, 0.4 and 0.5. In these premises, TSA is run 30 times for solving 2, 3, 4 and 5 dimensional functions, mean and standard deviation of the results are reported in Tables 2–21.

The analysis results show that a small size of the population is enough to obtain well-quality and robust results using TSA because the search in TSA is performed with the seeds. Higher values of ST parameter have caused the stagnation behavior for TSA because more dimensions of seed location are affected by the best solution in population obtained so far. Therefore, the small values for population size and ST parameter are required for obtaining good quality results with TSA. In the analysis, it is also shown that when the dimension of the optimization problem is increased, the performance of the method is decreased. This is because the dimensionality has caused the growth of search space exponentially. TSA is more successful in solving unimodal numeric functions than the multimodal numeric functions because the intensification capability of TSA (producing more than one seeds for a tree and effecting the best solution) is more effective than the global search capability. But results obtained with TSA for all numeric functions are in an acceptable and comparable level.

3.1.2. Comparison of TSA with the state-of-art methods on benchmark functions

According to parameter analysis of TSA, ST is set to 0.1 and the population size is taken as 10 in the comparative studies. In these premises, TSA is applied to solve the rest of the functions and obtained results are compared with the results of BA, FA, HS, ABC and PSO algorithms. The specific control parameters of BA are

Table 1
Benchmark Functions used in Experiments.

No of Func.	Name	Search Range	C	Function
F1	Sphere	$[-100,100]^D$	US	$f_1(\vec{X}) = \sum_{i=1}^{n} x_i^2$
F2	Elliptic	$[-100,100]^D$	UN	$f_2(\vec{X}) = \sum_{i=1}^{n} (10^6)^{(i-1)/(n-1)} x_i^2$
F3	SumSquares	$[-10,10]^D$	US	$f_3(\vec{X}) = \sum_{i=1}^{n} i x_i^2$
F4	SumPower	$[-10,10]^D$	MS	$f_4(\vec{X}) = \sum_{i=1}^{n} \lvert x_i \rvert^{(i+1)}$
F5	Schwefel2.22	$[-10,10]^D$	UN	$f_5(\vec{X}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$
F6	Schwefel2.21	$[-100,100]^D$	UN	$f_6(\vec{X}) = \max_i \{\lvert x_i \rvert,\ 1 \leqslant i \leqslant n\}$
F7	Step	$[-100,100]^D$	US	$f_7(\vec{X}) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$
F8	Quartic	$[-1.28,1.28]^D$	US	$f_8(\vec{X}) = \sum_{i=1}^{n} i x_i^4$
F9	QuarticWN	$[-1.28,1.28]^D$	US	$f_9(\vec{X}) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$
F10	Rosenbrock	$[-10,10]^D$	UN	$f_{10}(\vec{X}) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$
F11	Rastrigin	$[-5.12,5.12]^D$	MS	$f_{11}(\vec{X}) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$
F12	Non-Continuous Rastrigin	$[-5.12,5.12]^D$	MS	$f_{12}(\vec{X}) = \sum_{i=1}^{n} [y_i^2 - 10\cos(2\pi y_i) + 10]$ $y_i = \begin{Bmatrix} x_i & \lvert x_i \rvert < \frac{1}{2} \\ \frac{round(2x_i)}{2} & \lvert x_i \rvert \geqslant \frac{1}{2} \end{Bmatrix}$
F13	Griewank	$[-600,600]^D$	MN	$f_{13}(\vec{X}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F14	Schwefel2.26	$[-500,500]^D$	UN	$f_{14}(\vec{X}) = 418.98 * n - \sum_{i=1}^{n} x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$
F15	Ackley	$[-32,32]^D$	MN	$f_{15}(\vec{X}) = -20\exp\left\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right\} - \exp\left\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right\} + 20 + e$
F16	Penalized1	$[-50,50]^D$	MN	$f_{16}(\vec{X}) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1) \quad u_{x_i,a,k,m} = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leqslant x_i \leqslant a \\ k(x_i - a)^m & x_i < -a \end{cases}$
F17	Penalized2	$[-50,50]^D$	MN	$f_{17}(\vec{X}) = \frac{1}{10}\left\{\sin^2(\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_{i+1})\right]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$
F18	Alpine	$[-10,10]^D$	MS	$f_{18}(\vec{X}) = \sum_{i=1}^{n} \lvert x_i \cdot \sin(x_i) + 0.1 \cdot x_i \rvert$
F19	Levy	$[-10,10]^D$	MN	$f_{19}(\vec{X}) = \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + \lvert x_n - 1 \rvert[1 + \sin^2(3\pi x_n)]$
F20	Weierstrass	$[-0.5,0.5]^D$	MN	$f_{20}(\vec{X}) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k_{\max}}\left[a^k\cos(2\pi b^k(x_i + 0.5))\right]\right) - D\sum_{k=0}^{k_{\max}}\left[a^k\cos(2\pi b^k 0.5)\right] \quad a = 0.5,\ b = 3,\ k_{\max} = 20$
F21	Schaffer	$[-100,100]^D$	MN	$f_{21}(\vec{X}) = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^{n} x_i^2}\right) - 0.5}{\left(1 + 0.001 * \left[\sum_{i=1}^{n} x_i^2\right]\right)^2}$
F22	Himmelblau	$[-5,5]^D$	MS	$f_{22}(\vec{X}) = \frac{1}{n}\sum_{i=1}^{n}(x_i^4 - 16x_i^2 + 5x_i)$
F23	Michalewicz	$[0,\pi]^D$	MS	$f_{23}(\vec{X}) = -\sum_{i=1}^{n}\sin(x_i)\sin^{20}\left(\frac{i * x_i^2}{\pi}\right)$
F24	Dixon&Price	$[-10,10]^D$	UN	$f_{23}(\vec{X}) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$

Table 2
The analysis results of 2-dimensional functions for ST = 0.1.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	3.99E−232	0.00E+00	2.87E−102	1.57E−101	1.56E−65	4.19E−65	1.70E−43	4.50E−43	2.42E−32	4.14E−32
F10	3.67E−16	1.42E−15	2.12E−09	5.36E−09	2.42E−06	9.37E−06	9.63E−06	1.84E−05	1.10E−04	4.61E−04
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F13	1.23E−04	2.80E−03	3.28E−04	1.39E−03	1.93E−04	7.51E−04	3.22E−04	1.08E−03	5.98E−05	1.61E−04
F15	0.00E+00	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00

Table 3
The analysis results of 3-dimensional functions for ST = 0.1.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.01E−238	0.00E+00	8.01E−110	3.62E−109	1.17E−66	5.11E−66	1.96E−44	2.72E−44	9.65E−33	1.34E−32
F10	3.30E−03	1.05E−02	9.74E−03	1.37E−02	2.03E−02	4.92E−02	2.25E−02	3.58E−02	1.75E−02	2.62E−02
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.55E−16	1.95E−15
F13	3.06E−03	3.88E−03	2.96E−03	3.99E−03	3.97E−03	4.07E−03	6.11E−03	4.32E−03	7.88E−03	4.54E−03
F15	0.0E+00	0.0E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00

loudness (A) and rate of pulse emission (r). They are set to 1.5 and 0.5, respectively. In addition, α and β constants for BA are set to 0.9 according to (Yang & Gandomi, 2012) and the population size for BA is taken as 20 for the experiments based on (Yang, 2012). The minimum value for attractiveness (β), absorption coefficient (γ), randomization (α) and population size are set to 0.2, 0.5, 1, 20 as FA control parameters, respectively (Yang, 2011). There are five control parameters named as harmony memory size (HMS), harmony memory consideration rate (HSMCR), pitch adjusting rate (PAR) and arbitrary distance bandwidth (BW) in HS algorithm. These control parameters are set to 5, 0.9, 0.3, 0.01 according to (Wang & Yan, 2013), respectively. ABC algorithm has only two

Table 4
The analysis results of 4-dimensional functions for ST = 0.1.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	4.64E−240	0.00E+00	2.08E−111	9.27E−111	2.32E−68	5.78E−68	1.76E−44	7.62E−44	2.19E−33	2.68E−33
F10	1.70E−01	3.67E−01	2.20E−01	4.14E−01	1.30E−01	2.07E−01	1.10E−01	1.30E−01	1.14E−01	1.13E−01
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.29E−15	4.41E−14	2.30E−09	1.05E−08
F13	7.95E−03	6.69E−03	1.21E−02	8.15E−03	2.19E−02	1.15E−02	2.22E−02	1.06E−02	2.80E−02	1.29E−02
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	1.01E−15	6.49E−16

Table 5
The analysis results of 5-dimensional functions for ST = 0.1.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	7.64E−243	0.00E+00	1.17E−113	4.16E−113	2.41E−69	5.94E−69	9.01E−46	1.35E−45	7.35E−34	7.79E−34
F10	3.15E−01	1.05E+00	4.50E−01	6.62E−01	3.35E−01	2.99E−01	4.04E−01	3.00E−01	4.49E−01	3.09E−01
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.05E−15	4.41E−14	2.85E−07	1.52E−06	1.29E−03	4.81E−03
F13	2.02E−02	1.57E−02	2.53E−02	1.53E−02	4.02E−02	2.23E−02	4.66E−02	1.74E−02	5.88E−02	2.20E−02
F15	7.11E−16	1.45E−15	1.01E−15	6.49E−16	8.88E−16	0.00E+00	1.01E−15	6.49E−16	1.24E−15	1.08E−15

Table 6
The analysis results of 2-dimensional functions for ST = 0.2.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	7.12E−233	0.00E+00	1.18E−100	6.35E−100	6.22E−62	1.76E−61	3.98E−41	6.79E−41	8.48E−30	4.25E−29
F10	1.81E−12	5.25E−12	6.82E−07	2.49E−06	6.81E−06	2.96E−05	3.54E−05	4.30E−05	6.53E−05	1.09E−04
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F13	5.08E−04	1.80E−03	7.39E−05	4.05E−04	1.54E−05	7.49E−05	2.50E−04	1.35E−03	9.95E−05	3.22E−04
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	1.60E−15	1.45E−15

Table 7
The analysis results of 3-dimensional functions for ST = 0.2.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	3.40E−242	0.00E+00	1.83E−105	8.30E−105	6.05E−64	2.08E−63	1.67E−42	2.39E−42	2.47E−31	3.46E−31
F10	4.13E−02	1.62E−01	1.52E−02	3.48E−02	1.47E−02	1.78E−02	2.06E−02	2.78E−02	2.35E−02	3.29E−02
F11	6.63E−02	2.52E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.67E−14	8.81E−14
F13	2.60E−03	3.99E−03	2.33E−03	3.40E−03	4.12E−03	4.18E−03	5.44E−03	4.51E−03	9.68E−03	4.72E−03
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	1.48E−15	1.35E−15

Table 8
The analysis results of 4-dimensional functions for ST = 0.2.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.27E−247	0.00E+00	1.05E−108	3.48E−108	8.65E−66	2.44E−65	1.71E−43	3.25E−43	8.10E−32	1.05E−31
F10	3.54E−01	1.05E+00	7.86E−02	1.03E−01	1.08E−01	1.10E−01	1.09E−01	1.15E−01	1.46E−01	1.30E−01
F11	6.63E−02	2.52E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.09E−14	5.58E−14	1.37E−09	6.82E−09
F13	7.49E−03	7.13E−03	1.05E−02	8.51E−03	2.05E−02	1.09E−02	2.65E−02	1.18E−02	3.09E−02	1.20E−02
F15	1.13E−15	9.01E−16	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	1.01E−15	6.49E−16

Table 9
The analysis results of 5-dimensional functions for ST = 0.2.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.79E−248	0.00E+00	1.33E−110	2.44E−110	4.71E−67	9.43E−67	2.50E−44	5.87E−44	1.53E−32	1.59E−32
F10	4.45E−01	9.48E−01	3.63E−01	4.24E−01	2.51E−01	2.19E−01	2.72E−01	2.72E−01	4.75E−01	3.05E−01
F11	1.33E−01	3.44E−01	0.00E+00	0.00E+00	7.11E−16	3.89E−15	2.01E−03	9.71E−03	5.11E−02	1.91E−01
F13	1.90E−02	1.25E−02	2.54E−02	1.57E−02	3.93E−02	2.04E−02	6.08E−02	2.72E−02	5.81E−02	1.87E−02
F15	1.24E−15	1.08E−15	8.88E−16	0.00E+00	1.01E−15	6.49E−16	1.01E−15	6.49E−16	2.43E−15	1.79E−15

Table 10
The analysis results of 2-dimensional functions for ST = 0.3.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.68E−215	0.00E+00	4.59E−95	1.49E−94	2.72E−58	9.01E−58	3.06E−39	6.96E−39	3.38E−29	8.00E−29
F10	8.19E−11	4.41E−10	1.27E−06	3.85E−06	1.61E−05	2.70E−05	1.25E−04	3.54E−04	2.79E−04	9.83E−04
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F13	9.87E−04	2.56E−03	1.07E−04	5.55E−04	1.54E−05	6.13E−05	2.61E−04	1.22E−03	1.31E−04	4.59E−04
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	6.34E−15	3.82E−15

Table 11
The analysis results of 3-dimensional functions for ST = 0.3.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.28E−234	0.00E+00	1.04E−100	3.56E−100	2.40E−61	3.65E−61	4.29E−40	1.01E−39	9.70E−30	1.95E−29
F10	4.52E−01	1.08E+00	2.11E−02	4.73E−02	2.69E−02	5.03E−02	2.78E−02	3.66E−02	3.90E−02	7.96E−02
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.19E−13	1.68E−12
F13	4.14E−03	4.09E−03	2.66E−03	3.33E−03	5.77E−03	3.89E−03	6.23E−03	5.48E−03	8.47E−03	5.03E−03
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	4.20E−15	1.85E−15

Table 12
The analysis results of 4-dimensional functions for ST = 0.3.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.17E−239	0.00E+00	8.28E−105	3.73E−104	3.19E−63	4.41E−63	1.65E−41	2.45E−41	2.92E−30	4.46E−30
F10	2.90E−01	3.95E−01	1.36E−01	1.66E−01	1.02E−01	1.14E−01	1.29E−01	1.14E−01	1.33E−01	1.36E−01
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.76E−13	3.15E−12	3.35E−06	1.82E−05
F13	7.33E−03	5.93E−03	1.13E−02	7.72E−03	2.23E−02	9.21E−03	2.51E−02	1.36E−02	3.26E−02	1.30E−02
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	3.61E−15	1.53E−15

Table 13
The analysis results of 5-dimensional functions for ST = 0.3.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	8.10E−243	0.00E+00	2.14E−106	7.42E−106	2.27E−64	4.57E−64	3.72E−42	6.06E−42	5.54E−31	5.81E−31
F10	3.63E−01	7.51E−01	2.66E−01	2.05E−01	2.95E−01	2.31E−01	3.66E−01	2.72E−01	3.75E−01	3.33E−01
F11	3.65E−01	8.05E−01	0.00E+00	0.00E+00	1.76E−12	7.78E−12	1.16E−03	6.33E−03	2.06E−02	1.01E−01
F13	1.74E−02	9.87E−03	2.88E−02	1.98E−02	3.84E−02	1.83E−02	6.25E−02	2.45E−02	6.43E−02	2.14E−02
F15	1.36E−15	1.23E−15	1.01E−15	6.49E−16	1.01E−15	6.49E−16	1.13E−15	9.01E−16	3.02E−15	1.77E−15

Table 14
The analysis results of 2-dimensional functions for ST = 0.4.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	4.15E−208	0.00E+00	1.42E−89	5.03E−89	6.61E−55	1.54E−54	1.33E−36	2.81E−36	2.65E−27	5.67E−27
F10	1.77E−09	8.54E−09	1.30E−06	3.08E−06	1.79E−05	2.42E−05	4.03E−05	3.96E−05	1.25E−04	2.58E−04
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F13	9.94E−06	5.10E−05	9.38E−05	3.94E−04	2.54E−05	7.31E−05	1.81E−04	4.41E−04	3.82E−04	6.21E−04
F15	1.01E−15	6.49E−16	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	3.24E−14	2.48E−14

Table 15
The analysis results of 3-dimensional functions for ST = 0.4.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	4.71E−220	0.00E+00	2.24E−95	1.01E−94	1.36E−57	3.04E−57	2.76E−38	3.15E−38	6.54E−28	7.03E−28
F10	2.41E−02	5.81E−02	1.62E−02	2.28E−02	2.51E−02	3.06E−02	2.31E−02	1.89E−02	2.12E−02	3.68E−02
F11	1.33E−01	7.27E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.39E−11	3.67E−10
F13	2.84E−03	4.08E−03	2.54E−03	3.22E−03	5.57E−03	4.28E−03	7.71E−03	5.92E−03	9.36E−03	5.51E−03
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	1.75E−14	1.10E−14

Table 16
The analysis results of 4-dimensional functions for ST = 0.4.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.50E−230	0.00E+00	5.58E−99	1.80E−98	1.06E−59	1.54E−59	3.12E−39	6.41E−39	1.69E−28	3.49E−28
F10	5.75E−01	1.11E+00	1.67E−01	1.95E−01	1.30E−01	1.11E−01	1.24E−01	9.27E−02	1.75E−01	1.48E−01
F11	3.32E−02	1.82E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.47E−11	3.72E−10	2.49E−04	1.18E−03
F13	6.43E−03	5.90E−03	1.09E−02	6.17E−03	2.03E−02	7.64E−03	2.66E−02	1.43E−02	3.11E−02	1.17E−02
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	9.30E−15	3.90E−15

Table 17
The analysis results of 5-dimensional functions for ST = 0.4.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	5.52E−235	0.00E+00	2.86E−101	1.34E−100	3.34E−61	6.24E−61	3.45E−40	6.04E−40	3.27E−29	4.25E−29
F10	5.52E−01	1.09E+00	2.30E−01	1.94E−01	3.47E−01	2.77E−01	3.40E−01	2.53E−01	3.69E−01	3.21E−01
F11	9.95E−02	3.04E−01	0.00E+00	0.00E+00	8.72E−13	4.77E−12	4.59E−04	2.46E−03	2.67E−02	7.49E−02
F13	1.61E−02	9.46E−03	3.00E−02	1.50E−02	4.57E−02	1.82E−02	5.48E−02	2.14E−02	8.08E−02	2.27E−02
F15	1.60E−15	1.45E−15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	1.13E−15	9.01E−16	5.15E−15	1.96E−15

Table 18
The analysis results of 2-dimensional functions for ST = 0.5.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	4.76E−187	0.00E+00	6.65E−83	1.81E−82	1.15E−50	3.36E−50	2.01E−34	4.30E−34	1.58E−25	2.39E−25
F10	7.53E−10	1.82E−09	2.03E−06	3.73E−06	7.84E−05	1.77E−04	1.83E−04	5.00E−04	3.54E−04	5.11E−04
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.78E−15	6.90E−15
F13	5.15E−04	1.87E−03	1.04E−04	3.90E−04	2.11E−04	9.06E−04	2.88E−04	1.35E−03	3.45E−04	7.45E−04
F15	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	2.47E−13	2.29E−13

Table 19
The analysis results of 3-dimensional functions for ST = 0.5.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.25E−208	0.00E+00	9.66E−90	3.12E−89	1.16E−53	2.79E−53	1.28E−35	1.34E−35	2.28E−26	2.13E−26
F10	1.68E−02	4.20E−02	3.22E−02	5.77E−02	4.25E−02	7.89E−02	3.99E−02	4.69E−02	3.86E−02	4.98E−02
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.78E−15	7.15E−15	1.24E−08	6.03E−08
F13	2.20E−03	3.63E−03	3.83E−03	3.66E−03	6.04E−03	4.30E−03	7.08E−03	4.81E−03	1.12E−02	6.35E−03
F15	1.01E−15	6.49E−16	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	1.86E−13	1.51E−13

Table 20
The analysis results of 4-dimensional functions for ST = 0.5.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	7.85E−218	0.00E+00	1.48E−93	3.57E−93	9.36E−56	1.95E−55	5.95E−37	6.24E−37	1.02E−26	9.94E−27
F10	3.33E−01	7.14E−01	1.19E−01	1.69E−01	1.37E−01	1.57E−01	1.60E−01	1.33E−01	1.14E−01	1.04E−01
F11	6.63E−02	2.52E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.16E−06	2.28E−05	1.27E−04	5.06E−04
F13	6.18E−03	5.74E−03	1.31E−02	8.99E−03	2.16E−02	1.16E−02	3.07E−02	1.50E−02	3.52E−02	1.39E−02
F15	1.13E−15	9.01E−16	8.88E−16	0.00E+00	8.88E−16	0.00E+00	8.88E−16	0.00E+00	6.38E−14	3.50E−14

Table 21
The analysis results of 5-dimensional functions for ST = 0.5.

Func.	Pop = 10		Pop = 20		Pop = 30		Pop = 40		Pop = 50	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	8.96E−226	0.00E+00	1.31E−96	2.61E−96	7.26E−58	1.81E−57	1.22E−37	2.31E−37	1.76E−27	1.47E−27
F10	3.54E−01	7.77E−01	2.66E−01	2.61E−01	3.39E−01	2.97E−01	3.11E−01	2.05E−01	4.22E−01	3.12E−01
F11	9.95E−02	3.04E−01	0.00E+00	0.00E+00	1.10E−11	4.17E−11	1.79E−02	6.57E−02	1.10E−01	1.71E−01
F13	1.23E−02	8.09E−03	3.23E−02	1.35E−02	4.91E−02	1.95E−02	6.07E−02	2.55E−02	7.97E−02	2.70E−02
F15	5.49E−02	3.01E−01	1.13E−15	9.01E−16	8.88E−16	0.00E+00	1.84E−15	1.60E−15	2.35E−14	1.00E−14

Table 22
The comparison results of BA, FA, HS, ABC, PSO and TSA on the 2-dimensional benchmark functions.

FNO	BA		FA		HS		ABC		PSO		TSA	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	4.63E+01	1.32E+02	9.42E−08	7.30E−08	3.85E−03	1.13E−02	1.64E−18	1.64E−18	1.79E−05	6.86E−05	**3.99E−232**	**0.00E+00**
F2	6.47E+04	1.98E+05	1.68E+02	4.82E+02	3.76E+03	1.99E+04	1.76E−18	2.03E−18	5.21E+03	4.12E+03	**2.08E−228**	**0.00E+00**
F3	4.29E−10	4.55E−10	1.43E−09	1.62E−09	1.53E−09	1.87E−09	1.27E−18	1.23E−18	4.06E−07	1.50E−06	**1.67E−230**	**0.00E+00**
F4	3.43E−02	1.88E−01	1.97E−11	2.31E−11	7.82E−10	1.26E−09	6.33E−19	6.53E−19	1.41E−07	7.72E−07	**5.45E−270**	**0.00E+00**
F5	1.75E−01	5.18E−01	3.47E−05	2.04E−05	4.35E−05	2.68E−05	1.78E−17	8.74E−18	4.55E−03	1.85E−02	**4.30E−125**	**2.35E−124**
F6	7.17E+00	6.80E+00	2.80E−04	1.19E−04	2.07E−01	2.73E−01	3.64E−17	2.04E−17	2.54E−02	6.88E−02	**3.46E−99**	**1.47E−98**
F7	1.76E+02	2.05E+02	6.67E−02	2.54E−01	4.33E−01	6.26E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	1.37E−19	2.26E−19	8.99E−22	1.69E−21	4.28E−18	1.41E−17	3.06E−20	7.32E−20	1.50E−17	7.66E−17	**0.00E+00**	**0.00E+00**
F9	4.38E−02	6.80E−02	5.29E−02	4.46E−02	4.10E−03	5.35E−03	4.30E−04	2.95E−04	2.10E−03	3.45E−03	**2.81E−04**	**2.14E−04**
F10	2.07E+00	5.93E+00	1.24E−08	1.42E−08	1.92E+00	2.89E+00	8.62E−03	1.19E−02	1.91E+00	2.26E+00	**3.67E−16**	**1.42E−15**
F11	3.32E+00	2.97E+00	5.77E−08	5.48E−08	3.32E−02	1.82E−01	**0.00E+00**	**0.00E+00**	6.19E−02	2.12E−01	**0.00E+00**	**0.00E+00**
F12	3.83E+00	3.47E+00	6.89E−08	6.12E−08	6.18E−07	1.44E−06	**0.00E+00**	**0.00E+00**	6.47E−02	2.09E−01	**0.00E+00**	**0.00E+00**
F13	1.85E+00	2.11E+00	2.67E−03	3.95E−03	4.41E−02	3.23E−02	**0.00E+00**	**0.00E+00**	1.41E−02	8.50E−03	1.23E−03	2.80E−03
F14	6.55E+04	0.00E+00	5.53E+01	6.77E+01	6.61E−01	1.49E+00	**0.00E+00**	**0.00E+00**	1.38E+02	1.10E+02	3.95E+00	2.16E+01
F15	6.98E+00	3.66E+00	2.61E−04	1.18E−04	1.30E−04	9.29E−05	**0.00E+00**	**0.00E+00**	4.32E−02	1.59E−01	**0.00E+00**	**0.00E+00**
F16	7.88E+00	2.02E+01	3.40E−08	3.42E−08	5.18E−02	2.84E−01	1.20E−18	9.84E−19	1.23E−03	6.61E−03	**2.36E−31**	**4.45E−47**
F17	2.51E+02	9.44E+02	3.05E−08	2.88E−08	1.34E−02	1.79E−02	1.72E−18	1.73E−18	7.03E−06	2.71E−05	**1.35E−32**	**5.57E−48**
F18	4.04E−03	1.49E−02	4.00E−06	2.02E−06	5.71E−05	1.23E−04	2.13E−17	9.57E−18	8.63E−03	1.90E−02	**2.04E−17**	**1.11E−16**
F19	2.54E+00	2.73E+00	5.11E−06	3.09E−06	2.20E−02	4.47E−02	1.03E−19	1.48E−19	7.91E−04	1.75E−03	**1.35E−31**	**6.68E−47**
F20	4.77E−02	1.10E−01	1.75E−03	5.82E−04	1.02E−02	4.14E−03	**0.00E+00**	**0.00E+00**	5.17E−03	1.03E−02	**0.00E+00**	**0.00E+00**
F21	2.55E−01	1.35E−01	7.14E−03	3.08E−03	3.03E−02	2.09E−02	**1.63E−05**	**4.34E−05**	6.71E−03	3.81E−03	1.15E−03	2.93E−03
F22	−7.41E+01	6.59E+00	**−7.83E+01**	**7.76E−09**	**−7.83E+01**	**2.34E−08**	**−7.83E+01**	**1.45E−14**	**−7.83E+01**	**2.33E−01**	**−7.83E+01**	**1.45E−14**
F23	**−1.92E+00**	**6.06E−02**	−1.80E+00	6.14E−10	−1.80E+00	2.29E−08	−1.80E+00	9.03E−16	−1.16E+00	2.64E−01	−1.80E+00	9.03E−16
F24	1.28E−3	6.84E−2	4.41E−09	5.24E−09	4.66E−07	6.61E−07	8.13E−16	1.83E−15	6.11E−3	3.06E−2	**3.74E−32**	**2.25E−33**

control parameters named as population size and limit specific to the method. The number of bees in ABC hive is set to 40 and limit is set to 100 according to Matlab Code of ABC available in (Karaboga, 2008). The PSO control parameters are set according to PSO tutorials available in (Hu, 2006). The number of particles in PSO is set to 30, c1 (coefficient of cognitive component) and c2 (coefficient of social component) control parameters are set to 2.

Using these control parameters, each method is run 30 times with random initialization to solve for each 2, 3, 4, and 5 dimensional functions. The obtained results are reported in Table 22 for 2-dimensional functions, Table 23 for 3-dimensional functions, Table 24 for 4-dimensional functions, and Table 25 for 5-dimensional functions. The results of the method that produce the best result are marked with bold font type.

3.2. Performance assessment of TSA on multilevel thresholding

Image thresholding is one of the popular and simple methods used for image segmentation (Pal & Pal, 1993). Because image thresholding uses only the brightness of the image for image segmentation, this thresholding process is fast and it is preferred for real-world applications. The performance of image thresholding directly affects the success of the other image processing stages. Therefore, it is important that image thresholding should be

Table 23
The comparison results of BA, FA, HS, ABC, PSO and TSA on the 3-dimensional benchmark functions.

FNO	BA		FA		HS		ABC		PSO		TSA	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	3.68E+02	4.90E+02	1.43E−06	8.26E−07	1.05E−03	5.73E−03	6.27E−18	4.59E−18	6.36E−04	1.69E−03	**1.01E−238**	**0.00E+00**
F2	5.23E+06	8.52E+06	1.23E+03	1.46E+03	7.46E−01	3.07E+00	5.21E−18	3.37E−18	2.03E+06	3.56E+06	**2.96E−233**	**0.00E+00**
F3	3.46E−03	1.90E−02	2.12E−08	1.31E−08	3.33E−09	3.58E−09	5.57E−18	3.09E−18	4.52E−02	2.12E−01	**4.93E−239**	**0.00E+00**
F4	1.39E−02	7.59E−02	6.58E−11	9.98E−11	1.57E−10	1.97E−10	1.17E−18	1.23E−18	1.07E−01	2.67E−01	**0.00E+00**	**0.00E+00**
F5	1.87E+00	3.13E+00	1.42E−04	5.07E−05	5.44E−05	3.41E−05	3.48E−17	1.22E−17	9.87E−02	1.91E−01	**3.57E−135**	**1.90E−134**
F6	1.46E+01	1.17E+01	8.50E−04	2.81E−04	1.66E−01	2.30E−01	1.63E−16	1.02E−16	3.52E−02	6.64E−02	**1.48E−86**	**7.96E−86**
F7	6.18E+02	5.76E+02	6.67E−02	2.54E−01	1.67E−01	4.61E−01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F8	2.42E−17	3.04E−17	5.90E−20	7.00E−20	3.21E−17	1.13E−16	1.92E−19	1.84E−19	5.16E−05	2.80E−04	**0.00E+00**	**0.00E+00**
F9	7.55E−02	7.56E−02	4.60E−02	4.95E−02	3.03E−03	3.49E−03	9.74E−04	6.49E−04	5.87E−03	5.04E−03	**2.70E−04**	**1.81E−04**
F10	1.35E+01	4.09E+01	**4.45E−06**	**7.69E−06**	1.38E+00	1.73E+00	2.35E−02	2.57E−02	6.21E+00	4.65E+00	3.30E−03	1.05E−02
F11	8.52E+00	7.14E+00	3.32E−01	4.77E−01	5.02E−07	7.40E−07	**0.00E+00**	**0.00E+00**	2.03E+00	1.31E+00	**0.00E+00**	**0.00E+00**
F12	9.38E+00	7.08E+00	4.00E−01	4.98E−01	2.14E−06	6.26E−06	**0.00E+00**	**0.00E+00**	2.33E+00	4.32E+00	6.67E−02	2.54E−01
F13	5.75E+00	5.99E+00	1.29E−02	8.00E−03	6.80E−02	6.09E−02	**1.19E−12**	**6.49E−12**	1.07E−01	5.32E−02	3.06E−03	3.88E−03
F14	6.55E+04	0.00E+00	8.29E+01	7.06E+01	3.03E−01	6.49E−01	**7.58E−15**	**4.15E−14**	3.39E+02	1.16E+02	3.95E+00	2.16E+01
F15	1.25E+01	3.50E+00	8.05E−04	2.40E−04	1.22E−04	6.60E−05	3.55E−16	1.08E−15	1.78E+00	2.59E+00	**0.00E+00**	**0.00E+00**
F16	5.09E+04	2.79E+05	1.04E−07	8.66E−08	3.46E−02	1.89E−01	4.89E−18	3.64E−18	2.79E−02	1.10E−01	**1.57E−31**	**6.68E−47**
F17	2.33E+05	1.25E+06	1.88E−07	1.14E−07	6.16E−03	9.76E−03	5.08E−18	3.84E−18	1.63E−02	4.77E−02	**1.35E−32**	**5.57E−48**
F18	1.89E−01	3.82E−01	1.42E−05	6.07E−06	6.16E−05	6.56E−05	**4.81E−17**	**1.79E−17**	3.40E−01	4.65E−01	6.74E−12	3.69E−11
F19	5.03E+00	3.28E+00	8.23E−06	5.07E−06	1.10E−02	3.35E−02	1.94E−18	1.73E−18	3.23E−01	1.64E+00	**1.35E−31**	**6.68E−47**
F20	5.57E−01	3.76E−01	4.53E−03	1.02E−03	1.67E−02	6.33E−03	**0.00E+00**	**0.00E+00**	9.41E−02	1.08E−01	**0.00E+00**	**0.00E+00**
F21	3.53E−01	1.27E−01	9.68E−03	2.03E−04	2.99E−02	2.32E−02	**4.20E−03**	**4.63E−03**	1.34E−02	1.33E−02	9.50E−03	5.79E−03
F22	−6.99E+01	6.24E+00	**−7.83E+01**	**2.74E−08**	**−7.83E+01**	**1.91E−08**	**−7.83E+01**	**1.45E−14**	−7.65E+01	3.42E+00	**−7.83E+01**	**1.45E−14**
F23	−2.58E+00	2.32E−01	−2.74E+00	4.97E−02	**−2.76E+00**	**8.74E−08**	**−2.76E+00**	**6.85E−16**	−1.48E+00	3.23E−01	**−2.76E+00**	**5.09E−16**
F24	4.32E+00	1.22E+2	7.89E−08	4.65E−08	2.09E−06	6.33E−06	3.1E−14	5.69E−14	5.1E+00	1.44E+2	**3.21E−26**	**1.76E−25**

Table 24
The comparison results of BA, FA, HS, ABC, PSO and TSA on the 4-dimensional benchmark functions.

FNO	BA		FA		HS		ABC		PSO		TSA	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	8.95E+02	9.55E+02	5.03E−06	3.17E−06	1.19E−08	2.03E−08	1.32E−17	7.80E−18	2.22E+00	5.94E+00	**4.64E−240**	**0.00E+00**
F2	1.83E+07	3.42E+07	2.08E+03	2.20E+03	9.21E−02	2.07E−01	1.36E−17	8.12E−18	4.78E+06	1.68E+07	**4.49E−236**	**0.00E+00**
F3	5.91E−08	4.90E−08	8.18E−08	3.34E−08	1.04E−08	1.15E−08	1.42E−17	8.04E−18	1.52E+00	2.24E+00	**3.78E−241**	**0.00E+00**
F4	1.18E+00	3.43E+00	1.16E−10	2.35E−10	3.03E−10	8.17E−10	2.31E−18	1.81E−18	1.58E+01	2.69E+01	**0.00E+00**	**0.00E+00**
F5	2.36E+00	3.27E+00	3.18E−04	9.52E−05	8.35E−05	3.85E−05	5.17E−17	1.85E−17	2.36E+00	3.29E+00	**1.93E−142**	**1.02E−141**
F6	2.40E+01	9.69E+00	1.44E−03	3.73E−04	2.40E−01	2.70E−01	4.84E−16	3.44E−16	3.87E+00	6.69E+00	**8.29E−75**	**3.09E−74**
F7	2.00E+03	1.56E+03	6.67E−02	2.54E−01	3.00E−01	5.96E−01	**0.00E+00**	**0.00E+00**	1.07E+00	2.35E+00	**0.00E+00**	**0.00E+00**
F8	4.08E−16	4.86E−16	6.30E−19	5.42E−19	1.13E−16	3.01E−16	8.51E−19	8.21E−19	4.98E−03	1.53E−02	**0.00E+00**	**0.00E+00**
F9	1.93E−01	1.62E−01	4.16E−02	4.21E−02	3.88E−03	4.27E−03	1.64E−03	1.04E−03	2.64E−02	3.25E−02	**3.85E−04**	**2.25E−04**
F10	3.82E+01	8.15E+01	**9.03E−03**	**7.05E−03**	1.19E+00	1.48E+00	2.61E−02	2.81E−02	7.07E+02	2.53E+03	1.70E−01	3.67E−01
F11	1.59E+01	8.14E+00	1.13E+00	1.10E+00	8.40E−07	6.98E−07	**0.00E+00**	**0.00E+00**	1.01E+01	6.78E+00	**0.00E+00**	**0.00E+00**
F12	1.70E+01	8.71E+00	1.03E+00	6.69E−01	6.22E−07	5.83E−07	**0.00E+00**	**0.00E+00**	9.42E+00	9.03E+00	3.67E−01	4.90E−01
F13	1.52E+01	1.13E+01	2.79E−02	2.09E−02	7.99E−02	5.07E−02	**2.79E−10**	**1.39E−09**	3.74E−01	2.24E−01	7.95E−03	6.79E−03
F14	6.55E+04	0.00E+00	1.84E+02	1.15E+02	3.84E−01	6.88E−01	**7.58E−15**	**4.15E−14**	5.85E+02	1.79E+02	2.37E+01	4.82E+01
F15	1.42E+01	2.58E+00	1.25E−03	2.85E−04	2.11E−04	2.06E−04	**2.13E−15**	**1.77E−15**	1.05E+01	5.40E+00	8.88E−16	0.00E+00
F16	7.09E+04	2.98E+05	1.68E−07	1.03E−07	7.25E−09	1.13E−08	1.25E−17	6.52E−18	7.58E−01	1.67E+00	**1.18E−31**	**2.23E−47**
F17	2.12E+05	4.06E+05	3.44E−07	1.74E−07	6.85E−03	1.04E−02	1.19E−17	6.79E−18	4.32E+00	2.28E+01	**1.35E−32**	**5.57E−48**
F18	3.14E−01	6.00E−01	3.51E−05	7.42E−06	1.32E−04	1.22E−04	8.16E−17	4.32E−17	1.46E+00	7.95E−01	**4.08E−17**	**1.55E−16**
F19	9.14E+00	7.04E+00	1.03E−05	6.26E−06	1.47E−02	3.80E−02	7.01E−18	4.94E−18	3.51E+00	4.90E+00	**1.35E−31**	**6.68E−47**
F20	1.36E+00	5.44E−01	8.97E−03	1.71E−03	2.15E−02	7.33E−03	**0.00E+00**	**0.00E+00**	8.25E−01	7.14E−01	**0.00E+00**	**0.00E+00**
F21	4.17E−01	1.07E−01	9.72E−03	2.77E−07	4.79E−02	2.85E−02	9.09E−03	2.38E−03	1.19E−01	1.35E−01	**5.91E−03**	**5.65E−10**
F22	−7.03E+01	6.36E+00	**−7.83E+01**	**4.29E−08**	**−7.83E+01**	**2.37E−08**	**−7.83E+01**	**1.71E−14**	−7.00E+01	6.78E+00	**−7.83E+01**	**2.19E−14**
F23	−3.14E+00	4.01E−01	−3.65E+00	5.11E−02	**−3.70E+00**	**1.93E−06**	**−3.70E+00**	**1.49E−15**	−1.72E+00	3.36E−01	**−3.70E+00**	**1.70E−02**
F24	5.04E+00	1.58E+01	4.1E−07	2.41E−07	1.56E−1	2.85E−1	2.3E−13	5.86E−13	8.14E+1	1.07E+2	**1.59E−26**	**8.7E−26**

Table 25
The comparison results of BA, FA, HS, ABC, PSO and TSA on the 5-dimensional benchmark functions.

FNO	BA		FA		HS		ABC		PSO		TSA	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
F1	1.41E+03	1.11E+03	1.07E−05	4.91E−06	2.39E−08	2.81E−08	2.25E−17	8.00E−18	2.98E+01	5.14E+01	**7.64E−243**	**0.00E+00**
F2	3.10E+07	4.40E+07	6.32E+03	8.83E+03	2.13E−01	8.02E−01	1.82E−17	9.31E−18	1.17E+07	1.92E+07	**3.85E−240**	**0.00E+00**
F3	8.81E−03	3.34E−02	2.24E−07	1.10E−07	3.74E−08	5.12E−08	2.19E−17	8.59E−18	3.39E+01	3.94E+01	**5.44E−239**	**0.00E+00**
F4	1.29E+00	4.15E+00	2.54E−10	2.38E−10	1.99E−10	3.23E−10	3.20E−18	2.15E−18	2.45E+02	2.35E+02	**0.00E+00**	**0.00E+00**
F5	7.03E+00	7.38E+00	5.29E−04	1.13E−04	1.56E−04	8.49E−05	7.76E−17	2.06E−17	8.65E+00	6.67E+00	**3.49E−147**	**1.75E−146**
F6	2.80E+01	9.88E+00	2.12E−03	4.44E−04	3.79E−01	3.35E−01	2.08E−14	4.56E−14	1.69E+01	1.37E+01	**9.47E−63**	**5.15E−62**
F7	2.49E+03	1.23E+03	**0.00E+00**	**0.00E+00**	2.00E−01	4.84E−01	**0.00E+00**	**0.00E+00**	2.60E+02	1.07E+03	3.33E−02	1.83E−01
F8	3.48E−15	2.43E−15	1.91E−18	1.68E−18	7.49E−17	9.64E−17	2.29E−18	1.90E−18	8.29E−02	1.91E−01	**0.00E+00**	**0.00E+00**
F9	3.45E−01	2.82E−01	4.80E−02	3.64E−02	5.77E−03	5.29E−03	2.39E−03	1.22E−03	2.94E−01	2.64E−01	**5.16E−04**	**3.05E−04**
F10	7.95E+00	2.42E+01	2.15E−01	8.80E−01	1.52E+00	1.68E+00	**3.42E−02**	**3.57E−02**	1.84E+03	3.81E+03	3.15E−01	1.05E+00
F11	2.07E+01	9.29E+00	2.35E+00	1.39E+00	1.69E−06	1.36E−06	**0.00E+00**	**0.00E+00**	2.47E+01	9.34E+00	**0.00E+00**	**0.00E+00**
F12	2.65E+01	1.17E+01	2.40E+00	7.24E−01	3.33E−02	1.83E−01	**0.00E+00**	**0.00E+00**	2.30E+01	1.09E+01	4.67E−01	5.71E−01
F13	2.64E+01	1.57E+01	3.64E−02	2.62E−02	7.62E−02	4.84E−02	**1.23E−03**	**2.80E−03**	3.22E+00	1.06E+01	2.02E−02	1.57E−02
F14	6.55E+04	0.00E+00	2.44E+02	1.38E+02	1.74E−01	3.62E−01	**0.00E+00**	**0.00E+00**	9.02E+02	1.96E+02	1.18E+01	3.61E+01
F15	1.53E+01	2.78E+00	1.60E−03	4.02E−04	8.83E−03	4.72E−02	3.20E−15	1.08E−15	1.58E+01	3.30E+00	**7.11E−16**	**1.45E−15**
F16	1.12E+05	2.82E+05	2.57E−07	1.12E−07	1.72E−08	2.25E−08	2.07E−17	8.82E−18	1.48E+03	7.90E+03	**9.42E−32**	**3.34E−47**
F17	1.92E+06	4.35E+06	5.83E−07	2.76E−07	6.03E−03	1.04E−02	**1.94E−17**	**7.85E−18**	1.59E+05	5.83E+05	3.66E−04	2.01E−03
F18	5.57E−01	7.05E−01	5.46E−05	2.63E−05	1.48E−04	1.20E−04	**1.27E−16**	**7.56E−17**	3.42E+00	1.52E+00	1.30E−09	7.14E−09
F19	1.58E+01	1.22E+01	1.49E−05	7.31E−06	1.47E−02	3.80E−02	**1.14E−17**	**4.66E−18**	7.68E+00	4.57E+00	7.32E−03	2.79E−02
F20	2.26E+00	5.13E−01	1.29E−02	1.81E−03	3.29E−02	8.45E−03	**0.00E+00**	**0.00E+00**	2.68E+00	1.38E+00	**0.00E+00**	**0.00E+00**
F21	4.60E−01	4.59E−02	9.72E−03	4.45E−11	6.97E−02	3.63E−02	9.44E−03	1.07E−03	2.99E−01	1.57E−01	**7.16E−03**	**9.38E−10**
F22	−6.95E+01	5.08E+00	−7.80E+01	1.43E+00	**−7.83E+01**	**4.71E−08**	**−7.83E+01**	**1.58E−14**	−6.24E+01	7.77E+00	**−7.83E+01**	**1.77E−14**
F23	−3.39E+00	4.98E−01	−4.57E+00	7.56E−02	**−4.69E+00**	**1.11E−06**	**−4.69E+00**	**1.54E−15**	−1.76E+00	2.71E−01	**−4.69E+00**	**6.70E−02**
F24	2.07E+01	8.89E+01	8.66E−07	3.66E−07	1.56E−01	2.87E−01	3.31E−13	6.46E−13	1.17E+03	1.57E+03	**1.82E−26**	**8.99E−26**

performed to reveal as much as possible information in the images (Pekdemir, 2012).

There are many non-parametric methods and their applications for image thresholding are presented in the literature (Akay, 2013; Bhandari, Kumar, & Singh, 2015; Bhandari, Singh, Kumar, & Singh, 2014; Horng, 2011; Kapur, Sahoo, & Wong, 1985; Otsu, 1979; Pal & Pal, 1993; Pun, 1980; Sezgin & Tasaltin, 2000; Yin & Chen, 1997). A well-known method is Otsu's method based on histogram data of image. Therefore, we used maximization of between class variance proposed by Otsu (1979) for obtaining thresholds given an image.

3.2.1. Problem Formulation

Bi-level image thresholding focuses on dividing the image into two parts as background and foreground and can be used for retrieval of information and image segmentation. Let L be gray levels in a given image, t threshold value between 0 and $L-1$, I a given image, bi-level thresholding can be defined as:

$$\begin{aligned} P_F &= \{M(x,y) \in I \mid 0 \leqslant M(x,y) \leqslant t-1\} \\ P_B &= \{M(x,y) \in I \mid t \leqslant M(x,y) \leqslant L-1\} \end{aligned} \tag{8}$$

Bi-level thresholding can be extended to multilevel thresholding by increasing number of parts for the image and threshold values.

$$\begin{aligned} P_0 &= \{M(x,y) \in I | 0 \leqslant M(x,y) \leqslant t_0 - 1\} \\ P_1 &= \{M(x,y) \in I | t_0 \leqslant M(x,y) \leqslant t_1 - 1\} \\ &\cdots \\ P_n &= \{M(x,y) \in I | t_{n-1} \leqslant M(x,y) \leqslant L - 1\} \end{aligned} \tag{9}$$

For bi-level thresholding, t value can be easily obtained but multilevel thresholding can require more computational efforts. Therefore the heuristic search algorithm is more appropriate for finding the optimal t values in multilevel thresholding of an image.

For bilevel thresholding, Otsu (1979) defined and verified between-class variance for obtaining t threshold value. When we maximize the sum of the sigma functions calculated for each class, we can obtain optimal t value for bi-level thresholding. The mathematical definition of objective function is given as follows:

$$t^* = arg\max[f_b(t)] \tag{10}$$

$$f_b(t) = \sigma_0 + \sigma_1 \tag{11}$$

$$\sigma_0 = \omega_0(\mu_0 - \mu_T)^2 \quad \text{and} \quad \sigma_1 = \omega_1(\mu_1 - \mu_T)^2 \tag{12}$$

$$\mu_0 = \frac{1}{\omega_0}\sum_{i=0}^{t-1} i \times p_i \quad \text{and} \quad \mu_1 = \frac{1}{\omega_1}\sum_{i=t}^{L-1} i \times p_i \tag{13}$$

$$\omega_0 = \sum_{i=0}^{t-1} p_i \quad \text{and} \quad \omega_1 = \sum_{i=t}^{L-1} p_i \tag{14}$$

$$p_i = x_i / X \tag{15}$$

where, x_i is the number of pixels at level i, X is the total number of pixels at each level and p_i is the normal value of ith gray levels in Eq. (15). ω_0 and ω_1 are the estimated probability of 0th and 1st classes occurrence in Eq. (14), μ_0 is the mean intensity of 0th class, μ_1 is the mean intensity of 1st class Eq. (13) and μ_T is the mean intensity of the original image in Eq. (12). σ_0 is the variance of 0th class and σ_1 is the variance of 1st class in Eq. (11).

According to basic formulation of between-class variance, we extend between-class variance-based image thresholding to multilevel thresholding given as follows:

$$t^* = arg\max[f_m(t)] \tag{16}$$

$$f_b(t) = \sum_{i=1}^{n} \sigma_i \tag{17}$$

$$\sigma_0 = \omega_0(\mu_0 - \mu_T)^2, \quad \sigma_1 = \omega_1(\mu_1 - \mu_T)^2, \ldots, \quad \sigma_n = \omega_n(\mu_n - \mu_T)^2 \tag{18}$$

$$\mu_0 = \frac{1}{\omega_0}\sum_{i=0}^{t_0-1} i \times p_i, \quad \mu_1 = \frac{1}{\omega_1}\sum_{i=t_0}^{t_1-1} i \times p_i, \ldots, \quad \mu_n = \frac{1}{\omega_n}\sum_{i=t_n}^{L-1} i \times p_i \tag{19}$$

$$\omega_0 = \sum_{i=0}^{t_0-1} p_i, \quad \omega_1 = \sum_{i=t_0}^{t_1-1} p_i, \ldots, \quad \omega_n = \sum_{i=t_n}^{L-1} p_i \tag{20}$$

3.2.2. Application of TSA and ABC to the problem

Finding optimal threshold values in multilevel thresholding is a crucial task for image segmentation applications. If we divide an image into 5 parts with different gray level, we need 4 threshold values. When an image has 256 gray levels, Eq. (17) should be calculated 172061505 times for obtaining the optimal t values (Pekdemir, 2012). Therefore, this process can require more computational efforts and this exhaustive search cannot be suitable for real-time applications.

To overcome the high computation cost, heuristic search methods can be applied to solve the multilevel thresholding problem. Due to the fact that we obtain more quality results by using TSA and ABC on the benchmark functions, TSA and ABC methods are applied to maximize between-class variance (Eq. (17)) in multilevel thresholding for images given in Fig. 2. Lena and Cameraman images are frequently used in image processing applications as test images, and other pictures (Horses, Ostrich, Soldier, Starfish, Wherry, Zebra) are taken from (Martin, Fowlkes, Tal, & Malik, 2001). For TSA, the location of seeds or tree represents threshold value in the range of [0, 256] on 2, 3, 4 or 5 dimensional search space. The range is between 0 and 255 because images with 8 bit gray level are used in the experiments.

Fig. 2. The images used for testing the performance of ABC and TSA on multilevel thresholding.

For all images, we can try to obtain optimal 2, 3, 4 and 5 threshold values by using ABC and TSA. For the adaptation of the methods to the problem, before the objective function is evaluated by a seed of TSA or an agent of ABC, the possible solution (threshold values) is sorted and rounded. The TSA and ABC methods are run 10 times for each image and for each number of threshold value, and obtained the best results given in Tables 26–29 are compared with each other. Control parameters for the methods are set according to experiment 1 (solving the benchmark functions).

According to comparison tables, TSA and ABC show similar performance in solving the multilevel threshold problem but the running time of TSA is slightly better than the ABC algorithm as given in Fig. 3.

Table 26
The results of maximizing between-class variance for 2 threshold values by ABC and TSA.

Image	ABC			TSA		
	BCV	Elapsed time	Thresholds	BCV	Elapsed time	Thresholds
Cameraman	3929.462	1.76	65,142	3929.462	1.53	65,142
Horses	2458.408	2.03	91,169	2458.408	1.78	91,168
Lena	2634.77	1.78	77,145	2634.77	1.57	77,145
Ostrich	1086.159	2.17	75,135	1086.159	1.79	75,135
Soldier	1515.494	2.13	105,167	1515.494	2.03	105,167
Starfish	2590.543	2.47	83,156	2590.543	1.81	83,156
Wherry	3482.043	2.28	105,188	3482.043	1.97	105,188

Table 27
The results of maximizing between-class variance for 3 threshold values by ABC and TSA.

Image	ABC			TSA		
	BCV	Elapsed time	Thresholds	BCV	Elapsed time	Thresholds
Cameraman	4012.297	3.02	55,119,156	4012.297	2.72	55,119,156
Horses	2624.742	3.66	83,140,187	2624.742	3.27	83,140,187
Lena	2859.198	2.96	65,119,170	2859.198	2.65	65,119,170
Ostrich	1151.347	3.79	69,101,149	1151.347	3.39	69,101,149
Soldier	1667.793	3.88	92,133,184	1667.793	3.30	92,133,184
Starfish	2831.242	3.91	67,118,177	2831.242	3.84	67,118,177
Wherry	3721.888	3.96	98,158,217	3721.888	3.66	98,158,217

Table 28
The results of maximizing between-class variance for 4 threshold values by ABC and TSA.

Image	ABC			TSA		
	BCV	Elapsed time	Thresholds	BCV	Elapsed time	Thresholds
Cameraman	4069.157	4.78	37,92,138,169	4069.157	4.39	37,92,138,169
Horses	2711.834	5.37	68,114,150,194	2711.834	4.83	68,114,150,194
Lena	2943.803	4.38	57,101,138,179	2943.803	4.00	57,101,138,179
Ostrich	1192.025	5.60	65,93,128,178	1192.025	5.14	65,93,128,178
Soldier	1741.774	5.46	82,117,150,197	1741.774	5.26	82,117,150,197
Starfish	2917.5	5.78	59,100,137,187	2917.5	5.30	59,100,137,187
Wherry	3780.383	5.82	78,117,161,217	3780.383	5.45	78,117,161,217

Table 29
The results of maximizing between-class variance for 5 threshold values by ABC and TSA.

Image	ABC			TSA		
	BCV	Elapsed time	Thresholds	BCV	Elapsed time	Thresholds
Cameraman	4103.198	7.33	30,80,122,150,174	4103.198	6.49	30,80,122,150,174
Horses	2756.931	7.65	59,102,132,162,201	2756.931	6.99	59,102,132,162,201
Lena	2979.169	6.06	56,97,129,157,189	2979.169	5.60	56,97,129,157,189
Ostrich	1217.057	7.75	54,76,99,130,179	1217.057	7.05	54,76,99,130,179
Soldier	1784.428	7.96	74,106,132,163,207	1784.428	7.09	74,106,132,163,207
Starfish	2964.623	8.15	50,84,115,148,192	2964.623	7.31	50,84,115,148,192
Wherry	3818.731	8.09	75,113,150,183,224	3818.731	7.11	75,113,150,183,224

4. Results and discussion

Based on the comparison tables, it is shown that the proposed method is better than the other methods in most cases for solving the numeric benchmark functions.

In solving 2-dimensional functions in Table 22, ABC algorithm is better than the other methods on F13, F14 and F21 functions, BA is better than the other methods on F23 function, and all the algorithms have equal performance on F22 function, except BA. For F7 functions ABC, PSO and TSA are better than the other methods but they have equal performance. TSA and ABC show equal performance in solving 2-dimensional F11, F12, F15 and F20 functions. For the rest of 2-dimensional functions, TSA is better than the other methods.

For the results of 3-dimensional functions in Table 23, ABC algorithm is better than the other methods on F12, F13, F14 and F21 functions, and FA is better than the other methods in solving F10 function. FA, HS, ABC and TSA are better than the other methods on F22 function and HS, ABC and TSA are better than the other methods on F23 function. ABC, PSO and TSA have equal performance in solving F7 function, and ABC and TSA show equal performance on solving F11 and F20 functions. In solving the rest of 3-dimensional functions, TSA shows better performance than the other methods. Based on Table 24, FA is better than the other

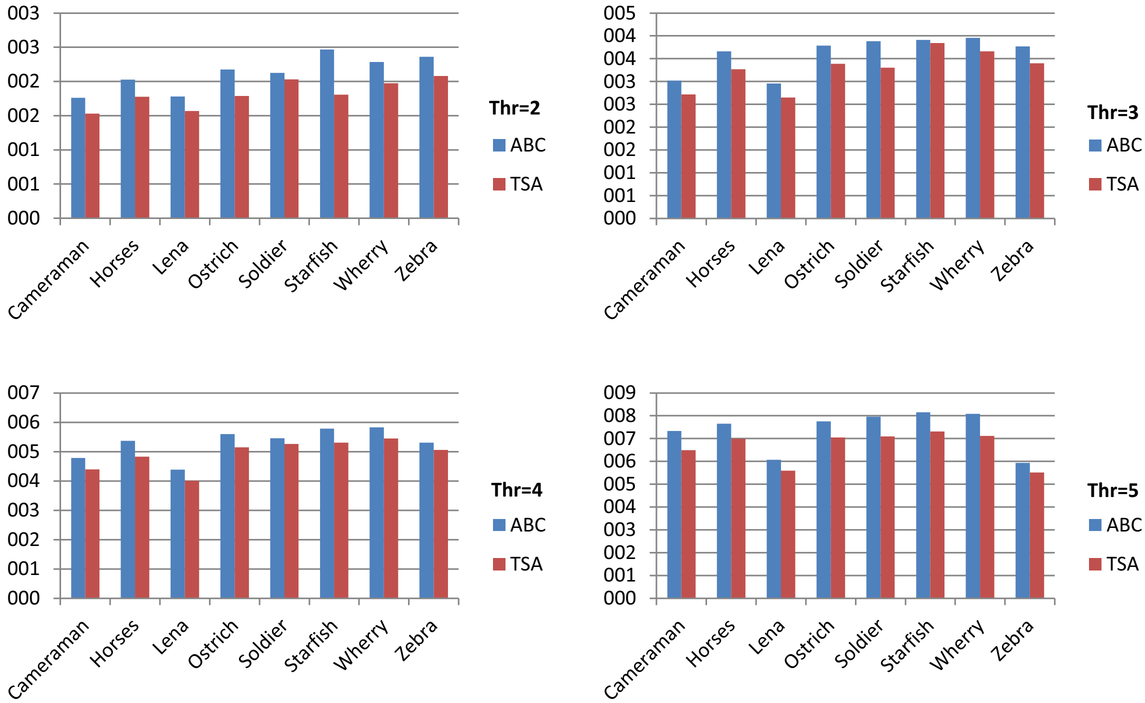

Fig. 3. The time comparison of ABC and TSA on the multilevel thresholding problem for different number of thresholding.

methods in solving F10 function, and ABC is better than the other methods on F12, F13, F14 and F15 functions. ABC and TSA are better than the other methods on solving F7, F11 and F20 functions but have equal performance. For F22 function, FA, HS, ABC and TSA have equal performance, and HS, ABC and TSA have equal performance on F23 function. TSA is better than the other methods in solving F1, F2, F3, F4, F5, F6, F8, F9, F16, F17, F18, F19, F21 and F23. Generally, TSA has higher or equal performance on 19 4-dimensional benchmark functions, but lower performance on 5 4-dimensional functions. At the same time, it is shown that TSA has higher or equal performance on 16 5-dimensional functions but slightly lower performance on 8 5-dimensional functions from Table 25.

Based on the image thresholding experiment, TSA method is successful in solving the multilevel thresholding problem as well as ABC algorithm, and has lower computation time than the ABC according to Fig. 3. In addition, this application shows that the TSA can be used in real-time image segmentation in terms of solution quality and computation time.

According to function optimization and image thresholding application, it is shown that TSA is an alternative and competitive solver for optimizing continuous optimization problems. The main reason is that TSA uses both random solution strategy (Eq. (4)) and the solution strategy with best tendency (Eq. (3)), and one or more seed locations are produced for each tree location at the each cycle of TSA. Based on standard deviations in the results tables, the TSA is a robust algorithm against unimodality, multimodality and initialization conditions. It is mentioned that when the dimension of the problem is increased, the performance of the methods decreases because the characteristics of the problem and search space range, which exponentially increases by depending on dimensionality of the problem, have caused an issue in solving the problem for the methods.

5. Conclusion and future works

The present study introduces a new population-based heuristic search algorithm. The algorithm simulates reproduction behavior of trees with seeds. In the algorithm, the location of trees and seeds n-dimensional space represents the possible solution for the optimization problem. TSA is an iterative algorithm, one or more seeds are produced for each tree, the best seed location is selected, and this seed substitutes for the parent tree. This process is carried out until a termination condition is met. The maximum number of function evaluations is used for terminating TSA method in this study. Based on the results tables, the performance of the proposed method is better than the other intelligent optimization methods in terms of solution quality and robustness in most cases. In solving multilevel thresholding, TSA has lower computation time than ABC algorithm according to comparison table.

As seen from the results, the proposed approach is better than the other methods in most cases because the search tendency of TSA is controlled and one or more solutions can be obtained around a solution to improve intensification of the population. The decision or design parameters of the problem are updated by using the solution (current tree location) with the best solution (best tree location) or randomly selected solution (neighbor tree location). The seeds (candidate solutions) obtained by this operation are compared with the current tree location and the best candidate solution is replaced with the current tree. More seeds provide to improve the exploitation ability of the method and affecting randomly selected tree provide to improve the exploration ability of the method. This search mechanism is different from the search mechanisms of the other population-based search algorithms.

The proposed method in this paper cannot be applied to solve discrete or binary optimization problem as is. Moreover, the performance of the proposed method should be compared with the other methods on solving huge dimensional optimization problems.

Future works include performance analysis of TSA on constrained optimization in short-term and we still pursue to develop a discrete version of TSA for solving discrete optimization problems. Two major topics for TSA in the future are discretization and adaptation to solve constrained optimization problems. In addition, the proposed method can be applied to neural network training, data clustering and optimization of high dimensional problems.

Acknowledgments

The author thanks Professor Binshan Lin and specialist referees for the constructive comments which are presented to improve both the content and the presentation of the study. The author also would like to thank *"The Scientific Projects Coordinatorship of Selcuk University"* and *"The Scientific and Technological Research Council of Turkey"* for their institutional supports.

References

Akay, B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing, 13*(6), 3066–3091. http://dx.doi.org/10.1016/j.asoc.2012.03.072.

Bhandari, A. K., Kumar, A., & Singh, G. K. (2015). Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. *Expert Systems with Applications, 42*(3), 1573–1601. http://dx.doi.org/10.1016/j.eswa.2014.09.049.

Bhandari, A. K., Singh, V. K., Kumar, A., & Singh, G. K. (2014). Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. *Expert Systems with Applications, 41*(7), 3538–3560. http://dx.doi.org/10.1016/j.eswa.2013.10.059.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics, 26*(1), 29–41. http://dx.doi.org/10.1109/3477.484436.

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

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading: Addison Wesley.

Holland, J. H. (1992). Adaptation in natural and artificial systems. *Second Edition*. Cambridge: MIT Press.

Horng, M. H. (2011). Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. *Expert Systems with Applications, 38*(11), 13785–13791. http://dx.doi.org/10.1016/j.eswa.2011.04.180.

Hu, X., (2006). Particle swarm optimization tutorial. Retrieved 29.11.2014.

Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision Graphics and Image Processing, 29*(3), 273–285. http://dx.doi.org/10.1016/0734-189x(85)90125-2.

Karaboga, D., (2008). Artificial bee colony algorithm matlab code. Retrieved 06.12.2014.

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. http://dx.doi.org/10.1007/s10898-007-9149-x.

Kennedy, J., & Eberhart, R., (1995). Particle swarm optimization. In *1995 IEEE international conference on neural networks proceedings, Vols. 1–6* (pp. 1942–1948).

Kiran, M. S., & Gunduz, M. (2013). A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems. *Applied Soft Computing, 13*(4), 2188–2203. http://dx.doi.org/10.1016/j.asoc.2012.12.007.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. http://dx.doi.org/10.1126/science.220.4598.671.

Martin, D., Fowlkes, C., Tal, D., & Malik, J., (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Eighth IEEE international conference on computer vision, Vol II, Proceedings* (pp. 416–423).

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems Man and Cybernetics, 9*(1), 62–66.

Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. *Pattern Recognition, 26*(9), 1277–1294. http://dx.doi.org/10.1016/0031-3203(93)90135-J.

Pekdemir, G., (2012). *A performance analysis of metaheuristic algorithms on multilevel thresholding problems* (Master). Konya, Turkey: Selçuk University.

Pun, T. (1980). A new method for grey-level picture thresholding using the entropy of the histogram. *Signal Processing, 2*(3), 223–237. http://dx.doi.org/10.1016/0165-1684(80)90020-1.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*(13), 2232–2248. http://dx.doi.org/10.1016/j.ins.2009.03.004.

Sezgin, M., & Tasaltin, R. (2000). A new dichotomization technique to multilevel thresholding devoted to inspection applications. *Pattern Recognition Letters, 21*(2), 151–161. http://dx.doi.org/10.1016/S0167-8655(99)00142-7.

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*. Singapore: Nanyang Technological University, IIT Kanpur: Kanpur Genetic Algorithms Laboratory.

Wang, X. H., & Yan, X. F. (2013). Global best harmony search algorithm with control parameters co-evolution based on PSO and its application to constrained optimal problems. *Applied Mathematics and Computation, 219*(19), 10059–10072. http://dx.doi.org/10.1016/j.amc.2013.03.111.

Yang, X. S. (2010a). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation, 2*(2), 78–84.

Yang, X. S. (2010b). A new metaheuristic bat-inspired algorithm. *Nicso 2010: Nature Inspired Cooperative Strategies for Optimization, 284*, 65–74.

Yang, X.S., (2011). Firefly algorithm matlab code. Retrieved 10.12.2014.

Yang, X.S., (2012). Bat algorithm matlab code. Retrieved 18.12.2014.

Yang, X. S., & Gandomi, A. H. (2012). Bat algorithm: A novel approach for global engineering optimization. *Engineering Computations, 29*(5-6), 464–483. http://dx.doi.org/10.1106/02644401211235834.

Yin, P. Y., & Chen, L. H. (1997). A fast iterative scheme for multilevel thresholding methods. *Signal Processing, 60*(3), 305–313. http://dx.doi.org/10.1016/S0165-1684(97)00080-7.