**A Study on Swarm Intelligent Algorithms for Multilevel Image Thresholding**

**Abstract**

Multilevel thresholding is essentially a combinatorial optimization problem. Traditional thresholding methods can no longer meet the requirements of real-time applications with the increasing of thresholding numbers. Therefore, eleven swarm intelligent algorithms are employed to solve the multilevel thresholding based on between class variance method and Kapur’s entropy method. Our comparative study indicates that the Cuckoo Search has a superior performance on multilevel thresholding based on both methods when compared to other algorithms. After analysing the mechanism and defects of the eleven swarm intelligent algorithms and the characteristics of pattern search algorithm, a strategy which related to stepsize-fixed pattern search algorithm is proposed to improve the swarm intelligent algorithms’ performance on multilevel thresholding. Our results indicate that swarm intelligent algorithms whose algorithm mechanism has a better balance of global exploration and local exploitation (such as Cat Swarm Optimization, Cuckoo Search, etc.) will lead to sound improvements after the proposed strategy applied.

**Keywords:** Swarm Intelligent Algorithms, Multilevel Thresholding, Pattern Search

**1 Introduction**

As an elementary step of image processing, visual analysis and pattern recognition, image segmentation is widely applied to character recognition, industry automation, machine visual, medical image analysis, on-line product monitor, optical stripe interpretation, military object detection, and so on[1]. It involves segmenting an image into non-overlapping, homogenous regions[2].

Image segmentation is a traditional problem, and a classical one as well. But no general segmentation method or general judge method of whether it’s properly segmented is found[3]. Large quantities of segmentation methods can mainly be categorized into three types[4]: edge detection or connection based methods, region-dependent segmentation, and thresholding segmentation. Additional, categorization can be done based on the certain theories they used[1], for instance, mathematical morphology based segmentation, information theory based segmentation, fuzzy theory based segmentation, neural networks based segmentation, meta-heuristic algorithms based segmentation, and so forth.

Thresholding technique is the most popular technique out of all the existing approaches used for segmentation of various types of images[5], because it’s easy to implement and understand. It’s widely used in applications such as infrared thermal nondestructive testing, objects extraction from SAR images, MRI segmentation, agricultural fruits quality monitor, etc[6].

The aim of thresholding is to find out the optimal threshold set to separate pixels into two or several classes. A great quantity of thresholding methods are proposed, and have different taxonomies. For example, according to the action scope of the selected thresholds, it can be partitioned to local thresholding method and global thresholding method[2]; according to the thresholding numbers, it can be divided into bi-level thresholding (also named binarization) and multilevel thresholding; Segmentation based on histogram can be categorized into maximum entropy thresholding (MET) method[7], maximum between class variance (BCV) method[8], and so on. According to the number of dimensions used in the histogram-based methods, they can further be divided to one-dimensional histogram methods, two-dimensional histogram methods[9], and three-dimensional histogram methods[10]; The entropy-based methods can further be divided to Shannon entropy method, Kapur’s entropy (KE) method, maximum Tsallis entropy thresholding[11] (MTET), exponential entropy method[12], reciprocal entropy method[13], Arimoto entropy method[14], etc. Entropy-based approaches have drawn attentions of many researchers. Superior performance of KE is recognized by the researchers[15] as compared to other thresholding techniques. The researcher has reported that the results using KE give improved performance and produce better average scores than the other entropy-based calculations on nondestructive sample images[1].

One more general classification is to divide numerous thresholding methods into parametric methods and non-parametric approaches. In the parametric approaches, the statistical parameters of the segmented classes in the image are estimated. They are computationally expensive, and their performance may vary depending on the initial conditions. Some parametric approaches are proposed, such as the P-tile method[16], the histogram concavity analysis method[17], etc. In the non-parametric approaches, the thresholds are determined by some criteria which are sophisticated designed, such as BCV, KE, etc. No matter in the parametric or non-parametric methods, they can be considered as combinatorial optimization problems[18] (that is, different approaches correspond to different specific optimization problems), and the computation complexity nonlinearly increases when bi-level thresholding extends to multilevel thresholding. The real-time performance is decisive to practical applications when the threshold number increases.

Lots of work have been tried to improve the real-time performance, and mainly can be grouped into two categories. The first one is to simplify the calculations the related segment algorithms needed, generally, decreasing re-calculating of some statistical variables[19] or use a divide-to-conquer strategy to lower the dimensions of the segment algorithms[20-23].The second one is to use the meta-heuristic algorithms which proved to have rapidity and stability for optimization problems to search the optimal parameters of the segment methods[18].

Meta-heuristic optimization algorithms are becoming more and more popular[24] because they are: ➀ simple: rely on rather simple concepts and are easy to implement; ➁ fast: faster than traditional techniques; ➂ black-box: to optimize problems covering different disciplines with a general framework. Nature-inspired meta-heuristic algorithms solve optimization problems by mimicking biological or physical phenomena. They mainly can be grouped in three categories (see Fig.1, all figures are listed at section “Figures”): evolution-based, physics-based, and swarm-based methods[25].

As a sub-branch of meta-heuristic algorithms, Swarm Intelligent (SI) algorithms are widely concerned and investigated. Just as the “No Free Lunch” (NFL) theorem[26] proposed by Wolpert and Macready indicates, no meta-heuristic is best suitable for solving all optimization problems, so new SI algorithms are proposed every year.

Some early algorithms have proved to be so useful in different application domains. Particle Swarm Optimization (PSO)[27], which is based on the social behavior of bird flocking. For its simple iterative mechanism, lots of research work based on PSO are gained popularity since its inception, and there are still scholars try to do researches based on it, and some of those researches can be tracked in website [28]. Some variants of PSO are listed in reference [29]. Shuffled Frog Leaping Algorithm (SFLA)[30], which based on the evolution of memes carried by individuals and a global exchange of information among the population[31], aims to model and mimic the behavior of frogs searching for food laid on stones randomly located in a pond. It takes the advantages of genetic-based memetic algorithm and the social behavior-based particle swarm optimization algorithm, and it has found applications in areas such as multivariable PID controllers[32], web document classification[33], and job shop scheduling[34]. Artificial Bee Colony Algorithm (ABCA)[35], inspired by the foraging behavior of honey bee swarm, and three kind of bees are designed in the algorithm, employed bees associated with specific food sources, onlooker bees observing the dance of employed bees within the hive to choose a food source, and scout bees searching for food sources randomly[36]. Numerical analysis reveals that the performance of ABCA is competitive to other population-based metaheuristic techniques[37]. The greatest advantage of ABCA is it employs very few control parameters. Owing to its accuracy and ease of implementation, it has attained lots of interest and has been exploited to solve various practical optimization problems, some of the improved variants are listed in reference [38], and some of the dynamics can be tracked in references[36, 38]. Cat Swarm Optimization (CSO)[39], inspired by the behaviors of cats, consists two sub-models, i.e., tracing mode and seeking mode, and has proved efficient in applications in areas such as filter design[40], flow-shop scheduling[41]. Firefly Algorithm (FA)[42], based on the flashing patterns and behavior of fireflies, can efficiently solve highly nonlinear, multimodal design problems in engineering fields[43]. Cuckoo Search (CS)[44], based on the obligate brood parasitic behavior of some cuckoo species in combination with the levy flight behavior of some birds and fruit flies, has attracted great attention due to its promising efficiency in solving many optimization problems and real-world applications[45]. Similar to the ABCA, it also deals with very few control parameters. Bat Algorithm (BA)[46], inspired by the micro-bats that use an echolocation for orientation and prey seeking. BA employs two parameters: the pulse rate and the loudness. The former regulates an improvement of the best solution, while the latter influences an acceptance of the best solution, some of the improved ones are chaotic bat algorithm[47], improved bat algorithm[48], quantum-behaved bat algorithm[49], just name a few. Other early proposed algorithms are Bacterial Foraging (BF) algorithm[50], Artificial Fish Swarm Algorithm (AFSA)[51], Ant Colony Optimization (ACO)[52], Monkey Search (MS)[53], Krill Herd (KH) algorithm[54], Dolphin Echolocation Optimization (DEO)[55], just name a few.

Recently, some new SI algorithms are proposed based on the NFL theorem to solve optimization problems. Grey Wolf Optimizer (GWO), mimics the leadership hierarchy and hunting mechanism of grey wolves in nature[56]. Four groups of grey wolves named alpha, beta, delta, and omega are employed to simulate the leadership hierarchy. In addition, three main steps of hunting, searching for prey, encircling prey, and attacking prey, are designed to perform optimization. It has proved to be feasible in application areas, such as training multi-layer perceptrons[57], solving optimal reactive power dispatch[58], etc. Ant Lion Optimization (ALO)[59], mimics the hunting mechanism of ant lions in nature. Five main steps of hunting prey are implemented, in detail, the random walk of ants, building traps, entrapment of ants in traps, catching preys, and re-building traps. It has found applications in areas, such as, reactive power dispatch[60], voltage stability enhancement[61], and so on. Moth Flame Optimization (MFO)[62], simulating the death behavior of moths, designs algorithms based on a different way of thinking. It has found applications in terrorism prediction[63], feature selection[64], etc. The Whale Optimization Algorithm (WOA)[65], is a new optimization technique for solving optimization problems. This algorithm includes three operators to simulate the search for prey, encircling prey, and bubble-net foraging behavior of humpback whales, some of the improved ones are enhanced whale optimization algorithm[66], adaptive whale optimization algorithm[67].

The swarm-based computing approaches stand out for their ability to search best solutions from any objective functions[68]. The use of these algorithms has become wide spread as they can produce high-quality solutions for difficult problems. Thresholding is essentially a constrained combinatorial optimization problem. According to statistical analysis of the objective values, swarm based algorithms are more accurate for multilevel thresholding problems[69]. Therefore, combining SI algorithms with multilevel thresholding to decrease the time-consumed to find out the optimal thresholds to meet the real-time application requirements, especially when the threshold number is high in which situation the computation complexity is highly enough to modern compute processors, is widely concerned.

Fruitful achievements have been reached, numerous thresholding works based on originally swarm algorithms or improved ones[70] are available in papers. Reference [71] adopts the PSO to search the near-optimal minimum cross entropy thresholding (MCET) thresholds based on a recursive programming technique. Reference [72] employs both cooperative learning and comprehensive learning along with cloning of fitter particles to improve the performance of PSO, the proposed algorithm (named HCOCLPSO) is used to multilevel thresholding, and shows good results. Reference [73] uses the quantum-behaved PSO employing the cooperative method to save computation time and to conquer the curse of dimensionality in multilevel thresholding based on BCV, and gets effective and efficient results. Reference [74] introduces an aggressive behavior to the movement rule in PSO, and then examines the proposed algorithm (named AgPSO) in thresholding based on BCV, and results are more efficient compared to other PSO variants considered in the paper. Reference [75] proposes a thresholding method named exponential cross entropy at first, and then adopts the niche chaotic mutation PSO algorithm to find the optimal multi-thresholds, experimental results show that the segmented images of proposed method are more accurate and their visual effect is improved significantly when compared with Shannon entropy thresholding. Reference [76] proposes a simpler and efficient PSO algorithm based on Bayesian theorem and the characters of intensity images, and the authors also adopt a new population initialization strategy to solve multilevel thresholding, results indicate the proposed algorithm(named BPSO) can produce better results in comparison with the other three existing methods on Berkeley datasets. Reference [77] proposes a novel optimization technique called Fibonacci PSO (FPSO) to decide the optimal thresholds by maximizing the objective function of Tsallis entropy, and the results obtained by the proposed method are significantly better than genetic algorithm(GA), BF, PSO and the Golden Ratio PSO (GRPSO, as proposed in reference [78]). Reference [29] firstly combines intermediate disturbance searching strategy with PSO (IDPSO), which enhances the global search ability of particles and increases their convergence rates, then applies the IDPSO to optimize 16 benchmark problems compared to ten known PSO variants, finally applies the IDPSO to multilevel thresholding, and experimental results show that it can effectively segment an image faster. Reference [79] combines the intrinsic principles of quantum mechanics with PSO and differential evolution (DE) algorithms, proposing the QPSO and QDE algorithms, and then, the QPSO and QDE are designed to find optimal thresholds of color images at different levels based on KE. Finally, a t-test statistical measurement is performed to ascertain the supremacy of the proposed algorithms. Reference [80] employs adaptive inertia and adaptive population strategies to improve the performance of PSO (and proposed new algorithm named Modified PSO, MPSO for short). The exploration or exploitation states are estimated by whether the gBest has been updated in *k* consecutive generations or not. The MPSO is used to find the optimal thresholds by maximizing BCV-based multilevel thresholding. Experimental results illustrate the solution quality of MPSO is better when compared to PSO and GA. Reference [81] applies the proposed dynamic-context cooperative quantum-behaved PSO to optimize the parameters for Otsu image segmentation for processing medical images. Experimental results show that the proposed method outperforms other state-of-the-art methods as listed in the paper. Reference [82] uses the modified PSO to optimize the proposed segmentation method based on two-dimensional Kullback-Leibler divergence, and results of extensively conducted experiment on the Berkeley Segmentation Dataset and Benchmark (BSD300) show the robustness and effectiveness.

Reference uses ACO to obtain the optimal parameters in segmentation based on fuzzy entropy approach. Reference [83] adopts the honey bee mating optimization (HBMO) to search for multilevel thresholding based on MCET method. Experimental results validate that the proposed HBMO-based MCET algorithm is efficient. Reference [84] adopts the honey bee mating optimization (HBMO) to search for multilevel thresholding based on MET. Experimental results validate that the proposed HBMO-based MET algorithm (MEHBMOT for short) is efficient. Reference [85] integrates ABCA with the Powell’s conjugate gradient method to compute optimum thresholds for MTET. Experimental results demonstrate that the integrated algorithm (called Refined ABCA, RABC for short) is efficient. Reference [86] analyses and discusses applying a family of artificial bee colony algorithms, namely, the standard ABCA, ABCA/best/1, ABCA/best/2, IABCA/best/1, IABCA/rand/1, CABCA, and some PSO-based algorithms to multilevel thresholding based on BCV. Experimental results show that the simple, general IABCA/best/1 is the best one among those compared algorithms. Reference [87] uses ABCA to calculate the parameters in Gaussian mixture model which is used to approximate the 1 D histogram of an image. Experimental results demonstrate the ABCA shows advantages as compared to other well-known algorithms. Reference [15] uses an improved solution search equation around the best solution of previous iteration to improve exploitation of ABCA, and also employs chaotic system and opposition-based learning method to improve global convergence when generating population. Experimental results demonstrate that the proposed algorithm (called MABC) is most promising in satellite image thresholding based on KE, BCV, and MTET as compared to PSO, ABC and GA. Reference [88] combines bee-to-bee communication pattern and multi-population cooperative mechanism to propose a modified ABCA (MABCA). The proposed new algorithm is employed to resolve the multilevel image segmentation problem, and experimental results demonstrate its superiority. Reference [89] adopts the Patch-Levy-based Bees Algorithm (PLBA) to solve multilevel thresholding based on KE and BCV. Experimental results demonstrate that the PLBA converges faster than basic bee algorithms, and also indicate that the bee algorithms-based thresholding outperform BF-based, quantum mechanism based thresholding, and also better than the non-metaheuristic-based Two-Stage Multi-threshold Otsu method (TSMO). In addition, the proposed PLBA-based algorithm is of high degree of stability. Reference [90] combines HBMA and cooperative learning to propose a cooperative honey bee mating-based algorithm (CHBMA) for natural scenery image segmentation using multilevel thresholding. Experiments demonstrate that the CHBMA can deliver more effective and efficient results than state-of-the-art population-based thresholding methods.

Reference [91] applies BF for multilevel thresholding based on KE and BCV. Experimental results on test images demonstrate the proposed BF technique is more robust and effective than PSO-based and GA-based thresholding. Reference [92] improves BF by adaptively selecting the exploitation and exploration state in chemotaxis of E.coli. bacterial to propose an adaptive BF for segmentation based on fuzzy entropy. Testing on benchmark gray images proves the proposed algorithm is suitable for thresholding. Reference [93] applies BF to recognize and segment the area of lips based on thresholding technique. Experimental results show that the proposed algorithm has less computational complexity and error and it is also efficient. Reference [94] passes the best bacteria among all the chemotactic steps to the subsequent generations to improve the global searching ability and convergence of BF. Experimental results on fourteen benchmark images affirm that the proposed algorithm (named Modified Bacterial Foraging algorithm, MBF for short) is more robust, proficiency and faster than BF-based, PSO-based, GA-based thresholding based on KE and BCV. Reference [95] proposes a new heuristic optimization algorithm, named amended bacterial foraging (ABF) algorithm for multilevel thresholding of MR brain images. Experimental results on 10 axial, T2 weighted Magnetic Resonance brain image slices indicate that the ABF is computationally more efficient, prediction wise more accurate and converges faster as compared to BF, PSO and GA. Reference[51] applies AFSA to thresholding based on two-dimensional Fisher function criterion. Experimental results demonstrate that the AFSA is a good method to select optimum 2D thresholds. Reference [96] designs an updating rule to extend the length of each frog jump by emulating frog perception and action uncertainties. The modified SFLA is applied to three-dimensional (3-D) Otsu thresholding. Experimental results show that the proposed algorithm is better as compared to the original 3-D Otsu and the fast recursive 3-D Otsu methods in terms of computation time and number of fitness function evaluation. Reference [97] applies the SFLA to multilevel thresholding based on MCET. Experimental results show that the SFLA is computationally effective as compared to HBMO, FA, PSO and ABCA. Reference [98] proposes a modified SFLA (MSFLA) for MR brain image segmentation. Compared to 3D-Otsu thresholding with SFLA and GA and also with the algorithm of segmentation using the Rician Classifier (RiCE), the proposed MSFLA is able to achieve better segmentation quality and execution time. Reference [99] applies FA to multilevel thresholding based on MCET. Experimental results show that the FA-based MCET is superior to the ones of PSO-based and QPSO-based MCET, but is similar to the HBMO-based MCET. Reference [100] proposes a Brownian distribution (BD) guided FA and presented with bounded search technique to improve the optimization accuracy with lesser search iterations. The proposed algorithm is applied to thresholding based on BCV. Compared to Levy flight (LF) guided FA and random operator guided FA, the BD guided FA provides better objective function value, PSNR, and SSIM, whereas LF guided FA faster in converge with lower CPU time. Reference [101] proposes two strategies (named Cauchy mutation and neighborhood strategy) to help fireflies escape from local optimal and accelerate the convergence. Experimental results show that the proposed algorithm (called Improved Firefly Algorithm, IFA for short) is more suitable for thresholding as compared to Darwinian PSO, hybrid DE, and FA. Reference [102] takes the advantages of DE and ABCA, and combines with BA to propose an improved bat algorithm (IBA). The IBA is then applied to thresholding based on KE and BCV, results indicate that IBA is comparable with PSO-based, FA-based, DE-based, BA-based, CS-based thresholding. Reference [103] adopts a newly proposed BA to thresholding based on maximizing the fuzzy entropy. Testing results on some natural and infrared images indicate that the proposed algorithm is better than ABCA, GA, PSO and ACO based thresholding in terms of objective function values, and also indicate the proposed algorithm is robust, adaptive, encouraging on the score of CPU time. Reference [104] applies CS to multilevel thresholding based on edge magnitude of an image, the second order statics (i.e. the gray level co-occurrence matrix) is used for obtaining thresholds by optimizing the edge magnitude. Results of the new proposed method are more encouraging both qualitatively and quantitatively than BCV. Reference [105] applies CS to multilevel thresholding based on MTET. Results indicate the proposed algorithm are encouraging in terms of CPU time and objective values as compared to BF, ABCA, PSO and GA. Reference [106] investigates the performance of CS and FA in multilevel thresholding based on KE and BCV. As compared to PSO, DE, both FA and CS exhibit superior performance and robustness. Reference [5] employs the CS, egg-laying radius based cuckoo search (ELR-CS) and wind driven optimization (WDO) to multilevel thresholding based on Kapur’s entropy. Results on color satellite image suggest that the CS performs well over all the testing images as compared to ELR-CS and WDO in terms of accuracy and keeping features of the image while WDO is more computationally efficient, and ELR-CS also show competitive results in some of the testing cases. Reference [107] investigates the flower pollination (FP) and the social spider optimization (SSO) algorithms to multilevel thresholding based on KE and BCV. Compare to PSO and BA, experimental results suggest that FP and SSO are more superior in terms of fitness values, PSNR and SSIM indices, and SSO can provide promising results though the number of thresholds increase. Reference [108] applies SSO to multilevel thresholding based on KE and BCV. Results on pre-tested images suggest that the SSO is superior to PSO in terms of optimal threshold values and computational time. Reference [109] combines the fundamental features of quantum systems with ACO, DE and PSO to propose the Quantum-based meta-heuristics, those Quantum-based algorithms are applied to multilevel thresholding based on MCET. Experimental results suggest that the proposed Quantum ACO is the best as compared to composite DE (CoDE), backtracking search optimization algorithm (BSA), ACO, PSO, and the other two quantum-based algorithms (i.e. QDE and QPSO).

Not all the SI algorithms have same optimization performance to specific optimization problems, so finding the “Free Lunch” is of great importance. So new meta-heuristics based thresholding methods are proposed every year, and they are compared to algorithms in literature. Here we will firstly try to give a comparative study of eleven SI algorithms applying to multilevel thresholding based on KE and BCV, as far as we know, the newly proposed swarm based algorithms (i.e. GWO, ALO, MFO and WOA) are firstly investigated to thresholding in this paper. What’s more important, after analyzing the characters of the experimental results, one improvement strategy kind of a plugin aims to enhance the exploitation of the swarm algorithms is also proposed.

The remainder of this paper is organized as follows. In the second section, the multilevel thresholding problem is formulated; in section three, the general flow of applying SI algorithms to multilevel thresholding is introduced; in section four one strategy related with pattern search is proposed to improve the performance of the eleven compared algorithms; and then in section five, the experimental study based on section three and section four are performed; and finally, section six is dedicated to conclusion.

**2 Multilevel thresholding problem formulation**

The aim of multilevel thresholding is to find out one optimal threshold set ***T*** to separate pixels into several classes. Multilevel thresholding is essentially a constrained combinatorial optimization problem.

Suppose there are *L* gray levels in a given image ***I***, *d* thresholds divide it into *d*+1 classes, which can be mathematical described by[110]:

 (1)

where ***I***(*i*,*j*) is the gray level of point (*i*,*j*), *tk* (*k*=0,1,2,…,*d*) is the *k*th threshold value, and *t0*=0, *td+1*=*L*, . It’s called bi-level thresholding (also named as binarization) when  whereas multilevel thresholding when, and , obviously, .

Selecting optimal thresholds for bi-level thresholding is not computationally expensive while selecting optimal thresholds for multilevel thresholding is exponentially computationally expensive with the increasing number of thresholds[19]. Generally, thresholding methods search for the optimal thresholds by optimizing some criterion functions (also known as objective functions). In this study, two widely investigated thresholding methods namely between class variance method and Kapur’s entropy method are employed.

**2.1 Between class variance method**

Between class variance method (BCV)[8], also referred as Otsu method or Maximum Inter-class Variance method or Minimum Intra-class Variance method, is proposed by Otsu in 1979. The basic idea of BCV is that the optimal thresholds segment the image into segments with maximum BCV.

Considering an image with *Ng* pixels, the number of the *i*th gray level is *Ni*, the probability of the *i*th gray level is *Pi*, then .  divides image into *d*+1 parts. If the probability of the *i*th part is  and the mean gray level of the *i*th part is, then the BCV of the image can be formulated as follows.

 (2)

where, , and  is the mean gray level of the whole image.

The BCV method to search the optimal thresholds  is to optimize the combinatorial optimal function, that is:

 (3)

where means arguments,  means finding maximum values. The meaning of the equation (3) is to find out the optimal argument ***T*** to maximize the function.

**2.2 Kapur’s entropy method**

Entropy is a metric of uncertainty, the greater the entropy, the more the uncertainty. Kapur’s Entropy (KE) method[7], also named as maximum entropy method, is proposed by Kapur et.al scholars in 1980. The basic idea of KE is that distributions of higher entropy will have higher multiplicities and are thus more likely to be observed[111].

The KE of the image can be formulated as follows.

 (4)

The KE method to search the optimal thresholds  is to optimize the combinatorial optimal function, that is:

 (5)

The meaning of other symbols in equation (4) and equation(5) is the same as in BCV method.

**3 Swarm intelligent algorithms applied to multilevel thresholding**

With the increasing of threshold number, no matter in the BCV based thresholding method or in the KE based thresholding method, the computation efficiency is not satisfactory with deterministic algorithms, so here we employ eleven swarm intelligent algorithms to improve the computation efficiency of solving multilevel thresholding based on BCV method and based on KE method.

The NFL theory says[26] that “any elevated performance over one class of problems is offset by performance over another class”. That means different SI algorithms have different optimization performance when applied to solve different specific problems. Therefore, comparative study of several SI algorithms applied to multilevel thresholding are enumerate investigated. The SI algorithms applied to solve multilevel thresholding in our study are PSO, SFLA, ABCA, CSO, FA, CS, BA, GWO, ALO, WOA and MFO.

The framework of using SI algorithms to solve multilevel thresholding problem based on BCV and KE methods is given in Fig.2.

The difference of SI algorithms is totally decided by their iterative mechanisms. Different SI algorithms have different iterative mechanisms, but all of them share one commonness that the iterative mechanism can mainly be divided into two search phases: exploration and exploitation. How and when the SI algorithms perform the two phases will lead to different optimization performance.

In order to have a better understanding of how to use the SI algorithms to solving BCV based and KE based multilevel thresholding problems. Here we take CS algorithm as an example.

Cuckoo search proposed by Yang and Deb in 2009 combines the breeding behavior of cuckoo birds with the Levy flight feature of animals and birds designing an efficient algorithm to solve optimization problems.

In the CS algorithm, one nest represents one feasible solution, to the optimization problem in the solution space. One egg of cuckoo bird represents one new feasible solution,, where  and *Dim* is the dimension of the problem. Cuckoo bird lays one egg each time , and choosing corresponding nest to lay the egg through Levy flight. The best nest,  is reserved to next generation. The number of nests is fixed to *N* in the CS algorithm formulation. Fraction  of worse nests are abandoned in each iteration. Goodness or badness of the nestand new eggis evaluated by the fitness value,, and the fitness function is generally directly related to the optimization problem.

One simple way to simulate Levy flight is proposed by Yang and Deb in a mathematical way formulated as follows:

 (6)

 (7)

where ,  are constant factors, represents gamma distribution, represents sine function, represents one vector whose *Dim* elements are random numbers derived from standard normal distribution.

Abandoning fraction  of worse nests is formulated in a mathematical way as follows:

 (8)

Where  when  otherwise .

 (9)

 (10)

Where 、 are random integers between 1 and *N* subject to uniform distribution and , rand is a random number between 0 and 1 subject to uniform distribution.

Corresponding to the multilevel thresholding situation, one nest or one host nest or one cuckoo bird egg is correspond to one solution to the BCV or the KE optimization problems, that is, the solution to equation(3) or equation(5), and the new egg is the CS iteratively calculated new solution to the BCV or the KE equations. The CS is terminated when the terminal condition meets (generally terminated when the maximum iteration *MaxT* is reached, just like the way our experiment designed, or terminated when the searching precision is meet, or other conditions), and the best nest,  the CS gets is corresponding to the optimal solution  or , for the reason that CS and other SI algorithms are randomized algorithms, the best nest the CS gets is generally the approximately solution to the optimal solution, which can be explained as sub optimal solution, and sometimes it’s equals to the optimal solution.

Detailed pseudo-code of the CS algorithm applied to BCV-based multilevel thresholding is described in Fig.3.

As to use the CS algorithm to solve KE-based multilevel thresholding, the only thing you have to do is to change the fitness function  to, and that is why the SI algorithms are called general computation schemes designed for optimization problems. It is easy and convenient to apply other SI algorithms to solve multilevel thresholding problems in a similar way.

Detailed introduction of other ten SI algorithms will not be given for the sake of concision, algorithms’ descriptions can be tracked by reading related reference as explained in section 5.1.

**4 Improved swarm intelligent algorithms applied to multilevel thresholding**

Though the SI algorithms show good performance in some optimization problems, but it’s a long way to go to meet the requirements of different specific situations. In order to improve the optimization performance of the SI algorithms, scholars have tried different ways, mainly can be grouped into three kinds: the first is to improve the local search ability, the second is to improve the global search ability, the third is to consider it in a comprehensively way, that is try to find a better balance of local search and global search. Here the local search is typically referred as exploitation and global search referred as the exploration.

Based on observing and analyzing previous scholars’ research and our experimental study in applying swarm intelligent algorithms to multilevel thresholding, we find that in most cases of failed finding the optimal thresholds, the thresholds the SI algorithms got is actually very close to the true optimal thresholds, only very small difference in some dimensions. For example, when applying CS algorithm to solve BCV based multilevel thresholding, supposing the threshold number is 5, and the optimal thresholds is, the CS generally gets trapped in  or similar results in most failed cases. The result is very closer to the optimal thresholds, so why not add some fine search strategies around the result  to improve its exploitation ability?

Based on the hypothesis that “the global optimal solution of the optimization function must exist in the numerous local-optimal solutions”, here we adopt pattern search algorithm to design such a strategy to improve the exploitation ability of the SI algorithms, and further to improve the optimization performance.

Pattern search (PS) algorithm[112], proposed by Hooke and Jeeves in 1961, is also named Hooke-Jeeves method. The basic idea of PS is to find out one “valley” that has smaller function value, and iteratively get closer to the minima along with the “valley”[113]. PS starts searching from the initial base point, and alternatively executing axis-based exploratory move and pattern move. Fig.4 is one demonstration of exploratory move and pattern move.

The detailed calculation steps of PS are sequentially executed as following:

➀ Set base point, *D* axis direction, initial stepsize, accelerate factor, reduction ratio, minimum stepsize, and set,,;

➁ If, then set, and go to step➃; else go to step➂;

➂ If  then set and go to step➃; else set and go to step➃;

➃ If then set and go to step➁; else go to step➄;

➄ If then go to step➅; else go to step➆;

➅ Set,, and set,, then go to step➁;

➆ If then stop iteration, and get point; else set,,,and set, and go to step➁.

where operator “” means to assign the value obtained by the right expression to the left variable.

The direction of pattern move is similar to the steepest descent direction[113]. It’s reliable and easy to implement when applied to optimization problems in which the optimization dimension is low.

Analyzing the mechanism of the PS algorithm, we can learn that the axis-based exploratory move can provide such a strategy to execute fine search around the base point, and pattern move can be thought of some kind of strategy getting out of local trick.

Denoting the PS algorithm as function  where  is a vector representing the searching initial base point, and  represents the control parameters of PS algorithm.

If we set, then the algorithm is stepsize-fixed pattern search algorithm. Fig.5 is one demonstration of the execution of PS when the dimension is 2 and the parameters aforementioned are used (the smaller number means the pattern search algorithm searches first)

From Fig.5 we know that, the PS starts from the base point  in position 1, then goes to position 2 by axis-based exploratory move, then goes to position 6 by axis-based exploratory move, then goes to position 10 by pattern move, and so forth, finally goes to position 19, where the function value of the can no longer be improved through axis-based exploratory move and pattern move. When we have performed pattern search algorithm from the base point, the pattern search result  can be promised to be the best solution in its four adjoining points, thus we can enhance the local convergence of the base point through pattern search algorithm.

Comparing the situation encountered in CS apply to BCV-based multilevel thresholding problem, we can consider using the stepsize-fixed pattern search algorithm to re-exploit the solution space around the swarm global history best solution, to improve the local performance of the SI algorithms when applied to multilevel thresholding. The reasons why we design the strategy in this way are: firstly, this can improve the local search precision, secondly, this can enhance its local convergence, thirdly, multilevel thresholding is essentially a constrained combinatory optimization problem, when applying swarm intelligent algorithms to multilevel thresholding, the minimum dimension distance of two feasible solutions is 1 (taking  and  as a case), so the stepsize-fixed pattern search can provide a step by step fine search of the feasible solution, for example, supposing the optimal thresholds is, the stepsize-fixed pattern search algorithm provide such a possibility to let  get to  in a fine search process, finally, we provided something kind of a plugin to swarm intelligent algorithms, which means it’s easy to apply this to other swarm intelligent algorithms in the same way, even apply this to other algorithms who have a such search mechanism.

Fig.6 is the flowchart of the improved SI algorithms (denoted as P-*Algorithms*, for instance, P-PSO, P-ABCA, etc.) applied to multilevel thresholding. As the flowchart in Fig.6 expressed, without doubt, the PS algorithm do increase the complexity of the SI algorithm in each iteration, but the PS have a guidance in searching direction, especially, have a fine searching around the swarm global history best solution, thus it may be expected to improve the optimization performance of the SI algorithms just like other fusion strategies did[114-116], and also it may be tricked by sub-optimal solutions.

In order to have a better understanding of how to implement the improved algorithms, here we take CS as an example, the detailed pseudo-code of P-CS algorithm is given in Fig.7.

From Fig.7, we can learn that it’s very convenient to apply the pattern search algorithm to the swarm intelligent algorithms in such a designed way.

We have reasons to believe that, if the original algorithm has a good balance of exploitation and exploratory, or the original algorithm has a good algorithm mechanism to escape from the sub-optimal solutions, then the P-Algorithms would have a better performance than the corresponding original algorithms. And we can also predict that, if the subsequent iterative mechanism of the original algorithm does not learn from the swarm global history best solution, then the P-Algorithms may not have a satisfactory improvement.

**5 Experimental study**

In this study, the PSO, SFLA, ABCA, CSO, FA, CS, BA, GWO, ALO, WOA and MFO totally eleven algorithms and corresponding improved algorithms were compared each other. Experiments have been done on seven images to select the optimal multilevel thresholds. KE and BCV are used as fitness objective functions. For both thresholding criteria, the exhaustive search method was conducted firstly to derive the optimal solutions for off-line performance analysis.

All the algorithms have been implemented in MATLAB 7.12 on a PC with Intel(R) Core(TM) i5 [M450 2.40GHz](mailto:M450@2.40GHz) processor with 2.99 GB of RAM and Windows 7 ×32 Pro operating system. Experiments were carried out on seven test images which have been widely employed in thresholding researches[18, 106, 110], namely, Zebra(481\*321), Cameraman(256\*256), Aerial(256\*256), Barbara(512\*512), Boat(512\*512), Goldhill(512\*512), Lake(512\*512). These testing images and their unique gray level histograms (normalized) are shown in Fig.8.

The number of thresholds *d* explored in our experiments is 2~5. Since SI algorithms are stochastic algorithms, each algorithm was repeated 50 times for each image and for each *d* value. The optimal thresholds, and its corresponding optimal objective function values provided by the exhaustive search for KE and between-class variance methods are presented in Table 1 (all the Tables are listed at section “Tables”).

**5.1 Parameter settings**

For fair comparison, same size of the swarm population of 40 (ABCA and CS are set to 20 for the reason that they evaluate objective function twice in per iteration) and maximum iteration number of 200 are set for all algorithms. Besides these common control parameters, other parameters do have a greatly effect on their performance. Testing by trial and error has been done to ensure each algorithm gets relatively good results.

Each individual and each updated individual of the SI algorithms will be converted to integers by the function  in MATLAB 7.12 and will also be boundary constrained. Modifications are needed for the reason that some parameters of the original proposed SI algorithms are no longer suitable for multilevel thresholding.

For the PSO, we adopted the implementation in reference[117], the inertia weight , the cognitive and social components , and we limited the maximum velocity to ;

For the SFLA, we took the implementation in reference[30], number of memeplex , number of evolutions in a memeplex between two successive shuffling , maximum step size , where  is the largest gray level value of the testing image;

For the ABCA, we implement it in the way as reference[38], the maximum number of trial for abandoning a source ;

For the CSO, the original proposed way in reference[39] is accepted, the seeking memory pool , seeking range of the selected dimension , counts of dimension to change , self-position considering , the constant , and the mixture ratio ;

For the FA and CS are accomplished as the way in reference[106], for the FA, , , , , where  and  are the largest and smallest non-zero gray level value of the testing image; for the CS, , ;

For the BA, the original proposed way in reference[118] is applied, the maximum and minimum frequency , , , where  is a standard normal distributed random number, and the pulse emission and loudness were fixed ;

For the GWO and ALO, the original parameters in the proposed references[56, 59] are used, as to the WOA and MFO, the parameter  is linearly decreasing from 2 to 1.9, and other parameters are the same as the original proposed[62, 65].

Finally, parameters of the pattern search algorithm are set to .

**5.2 Experiment results analysis**

In order to have an intuitive sense of the segment difference of the compared algorithms, segmented images and grey level histograms labelled with segment thresholds have been given in Fig.9 to Fig.10 for some images when the threshold number is 5. Because we have run 50 times to have a relatively fair comparison between the eleven SI algorithms and corresponding improved algorithms, and in most cases the best thresholds of the 50 runs of each SI algorithms are very close to each other, especially when the threshold number is small, so here we take the worst thresholds the SI algorithms get in the 50 runs to show visual differences. From Fig.9 to Fig.10, the visual difference is not so distinctive, so quantified comparative study is needed.

To have a quantified comparative study of the eleven algorithms (denoted as original algorithms) and the improved algorithms (denoted as P-Algorithms), the mean value (abbreviated as M.V.) and variance (abbreviated as Var.) for each algorithm based on BCV and KE for seven test images over 50 runs have been calculated. The mean convergence time (abbreviated as M.C.T.) for each algorithm based on between-class variance and KE for seven test images over 50 runs have been recorded. The success searching ratio (abbreviated as S.S.R.) for 50 runs have been calculated, and success searching means that the best solution the SI algorithms output exactly equals to the optimal thresholds provided by the exhaustive search.

M.V. can be used to evaluate the searching precision of the compared algorithms, Var. can be used to assessing the stability of the compared algorithms, M.C.T. can be used to evaluate the convergence speed of the compared algorithms, and S.S.R. (calculated off-line) can be used to predict the probability in what degree the algorithm can find the optimal thresholds. Therefore, the four assessment indexes are used to evaluate the comprehensive optimization performance of the compared algorithms.

5.2.1 BCV-based thresholding

At this section, the considered eleven algorithms and their corresponding P-Algorithms are employed to optimize BCV method, that is, to optimize the objective function equation (3) represents. Results are listed from Table 2 to Table 13.

➀ Comparison between the original algorithms:

From Table 2 to Table 3 and Table 5, we can conclude that: (i) when *d*=2, all the algorithms have same performance. Except for the ABCA, FA and GWO, all algorithms can find out the optimal results over 50 runs. Comparing the fitness mean value with the exhaustive results in Table 1, we can know the results obtained by ABCA, FA and GWO are very close to the optimal results; (ii) with the increasing of threshold numbers, the complexity of the optimization problem increases, so the problem is much more difficult to deal with. The optimization performance of each algorithm becomes worse, especially when the value of *d* is high, such as, when *d*=5, all the algorithms cannot find the optimal results in a probability of 100 percent over 50 repeating experiments. But, from Table 2 we can learn that the CS, ALO and WOA showed robustness to different threshold numbers; (iii) the difference in the histogram leads to the difference in optimization performance. For example, from the seven test images, observations from the results showed that all the algorithms in the test image “Aerial” having poor performance, and from Fig.5, comparing with other testing, the histogram of “Aerial” has much more peaks and does not have prominent main peak which leads to much more sub-optimal solutions in the optimization space (or the function space of BCV method).

From Table 4, we can know that the convergence speed of BA, PSO and WOA is faster than other compared algorithms in all cases. The simple iterative mechanisms and searching processing they adopted contribute to their faster convergence speed.

From Table 6, considering the searching precision, stability, convergence speed and the success searching ratio, the comprehensive performance ranking of the compared algorithms is:

CS>BA>WOA>ALO>PSO>FA>MFO>CSO>ABCA=SFLA>GWO.

where notation “>” means “superior to”, notation “=” means “equal to”.

Attention need to pay, the four assessment indexes have same weight here. If some of them are much more cared about, differentiated weights can be assigned to the four assessment indexes.

➁ Comparison between the improved algorithms:

From Table 7 and Table 9, we can conclude that: (i) when the threshold numbers is small (for example, *d*=2, 3 or 4), all the improved algorithms have good performance similarly. The P-CSO, P-CS, P-ALO, P-WOA can find out the optimal thresholds in a probability of 100 percent. All the P-algorithms can find out the optimal thresholds in a probability above 85 percent when the threshold numbers *d* is under 4; (ii) similar to the results of the original algorithms, most of the P-algorithms also seem to get trapped in the “Aerial” when *d*=4 or 5, and when *d* is large (for example *d*=5), the performance of all the P-algorithms except for P-ALO seems to get worse, which means the P-algorithms becomes more easier to be tricked when the threshold number getting larger.

From Table 8, we can get that the P-algorithms show no regular patterns which means the proposed strategy has different effectiveness to the improvement of the original algorithms’ convergence speed.

From Table 10, considering the searching precision, stability, mean convergence time and the success searching ratio, the performance ranking of the compared P-algorithms is:

P-CS>P-CSO>P-WOA>P-ALO=P-BA>P-PSO>P-FA>P-MFO=P-GWO>P-SFLA=P-ABCA.

➂ Comparison of the original algorithms with the improved algorithms:

Table 11 is the difference of the mean value form improved algorithms and the original algorithms. The results can represent the improvement degree of searching precision the proposed strategy bringing to. From Table 11, we can observe that only 5 cases out of 308 (=28\*11) showed decreasing in fitness mean value which strongly imply that the proposed strategy is of effectiveness to improve the searching precision of the original algorithms when applied to multilevel thresholding based on BCV method. Observations from Table 11 indicates that if the original algorithms whose searching mechanism containing “following (or flying to) the swarm global history best solution” (taking PSO as a case) or whose local searching operator containing “exploiting around the swarm global history best solution” (taking BA as a case), then the corresponding P-algorithms may be not expected to have improvements just as we have predicted in section 4.

Table 12 is the difference of the variance from the original algorithms and the improved algorithms. The results can represent the improvement degree of stability our proposed strategy bringing to. From the data in Table 12, we can derive that our proposed strategy is of effectiveness to improve the stability of the original algorithms when applied to multilevel thresholding based on BCV method. The stability of the P-algorithms will not be expected to improve where the optimization problem has plenty sub-optimal solutions (taking “Aerial” as a case).

Table 13 is the difference of the mean convergence time from the original algorithms and the improved algorithms. The results can represent the improvement degree of convergence speed the proposed strategy bringing to. From Table 13, we can know that the proposed strategy is of effectiveness to improve the convergence speed of the original algorithms (except for ABCA) in most cases because of the “searching direction” the pattern search algorithm bringing to. As to the ABCA, comparing the data in Table 11, Table 12 and in Table 13, we can derive that the proposed strategy let the ABCA have a great promotion in searching precision and stability at a cost of consuming more time.

5.2.2 KE-based thresholding

Set the KE method, that is, the equation (5) as the objective function, and repeat the same experiment processes in section 5.2.1. Results are listed from Table 14 to Table 24.

➀ Comparison between the original algorithms:

Observing the data from Table 14 to Table 16, and comparing the corresponding results in multilevel thresholding based on BCV method, conclusions can be concluded similarly as in BCV method. Firstly, when the threshold number is small all the original algorithms have similar optimization performance, and the performance getting poorer with the increasing in threshold number; Secondly, if the histogram is different, then the optimization performance is different as well; Finally, the convergence speed of the BA, PSO is faster than other compared algorithms.

Table 17 shows the performance ranking in the four considered assessment indexes. Comprehensively, the performance ranking (four assessment indexes considered) of the original algorithms when applied to multilevel thresholding based on KE method is:

CS>PSO>ALO>WOA>CSO>MFO>BA>FA>SFLA>GWO>ABCA.

➁ Comparison between the improved algorithms:

Observing data from Table 18 to Table 20, similar conclusions can be derived, that are, when the threshold number is small, all the P-algorithms showed similar optimization performance, and the increasing in threshold number do have a negative effect on the performance of the P-algorithms. And analyzing data in Table 26, we know that the P-PSO, P-FA, P-BA have faster convergence speed compared to other P-algorithms in most cases.

Table 21 shows the performance ranking in the four considered assessment indexes. Comprehensively, the performance ranking (four assessment indexes considered) of the P-algorithms when applied to multilevel thresholding based on KE method is:

P-CS>P-CSO>P-ALO>P-WOA>P-MFO=P-GWO>P-PSO>P-SFLA>P-FA>P-ABCA=P-BA

➂ Comparison of the original algorithms with the improved algorithms:

Analysing data in Table 22, we can conclude that the proposed strategy can improve the searching precision of all the algorithms except for PSO, BA, WOA and CS in the situation where the threshold number is large.

Observing data in Table 23, we can derive that the proposed strategy can make all the considered algorithms (except for PSO, FA, BA and WOA) more stable. The reason why the exceptions occur is that their searching mechanisms containing “only following (or flying to) the swarm global history best solution” or their local searching operator containing “exploiting around the swarm global history best solution”, in other words, these algorithms cannot have a good balance of global exploration and local exploitation in these specified cases.

The experimental results in Table 24 indicates that the “searching direction” brought by the pattern search can have a positive effect on accelerating the convergence speed of the original algorithms when applied to multilevel thresholding based on KE method.

Comparing data in Table 22, Table 23 and Table 24, we can conclude that the proposed strategy can improve the performance of the original algorithms when applied to multilevel thresholding based on KE method with respect to all or some aspects of searching precision, stability and convergence speed.

**6 Conclusion**

With the increasing of threshoding numbers, traditional thresholding methods can no longer meet the requirements of real-time applications. Currently, combining swarm intelligent algorithms which have strong optimization capability to find optimal thresholds based on certain criterion becomes a hot research spot. In this study eleven SI algorithms are involved, the purpose of this study is to try to find the “Free Lunch” in multilevel thresholding based on BCV method and KE method. As far as we know, the CSO, GWO, ALO, WOA, MFO are firstly applied to multilevel thresholding by our study, and some of them provide competitive results in some testing cases.

Comparison studies of the eleven algorithms with respect to the searching precision, stability, convergence speed and success searching ratio in the multilevel thresholding based on BCV method and KE method have been made. Results indicate that the performance of swarm intelligent algorithms is different when different thresholding criterions used. But considering the searching precision, stability, convergence speed and success searching ratio, the CS has a superior performance on multilevel thresholding based on both the BCV method and the KE method, while ABCA, SFLA and GWO have an inferior performance on both methods when compared to other algorithms.

After analyzing the mechanism of the eleven swarm intelligent algorithms and pattern search algorithm, a strategy which related to pattern search algorithm is proposed in our study to improve the swarm intelligent algorithms’ performance on multilevel thresholding. The strategy is that applying the stepsize-fixed pattern search algorithm to re-exploit the global history best solution of the swarm intelligent algorithms in per iteration. Results of our experiments indicate that the mechanisms of swarm intelligent algorithms which have good balance of global exploration and local exploitation will lead to better improvements after the proposed strategy applied. But the proposed strategy has differentiated improvements in the performance of swarm intelligent algorithms when different specific thresholding criterions used.

What’s more universally significant, as we know, the BCV based and the KE based thresholding are essentially low dimensional combinatorial constrained optimization problems, and our 616 testing cases (BCV based 308 cases plus KE based 308 cases) proved that the proposed strategy is feasible (in fact, another 616 cases are tested to further validate the feasibility of the proposed strategy) when the applied algorithm has a good balance of global exploration and local exploitation. Therefore, when these improved algorithms and the proposed strategy applied to similar situations, same sound effectiveness can also be expected to get.

**Tables**

**Table 1.** Optimal thresholds and fitness values provided by the exhaustive search for both between class variance method and KE method

| **Images** | ***d*** | **Between-class variance method** | | | | | | **KE method** | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **optimal thresholds** | | | | | **function value** | **optimal thresholds** | | | | | **function value** |
| **Zebra** | 2 | 100 | 173 |  |  |  | 1404.910975737 | 98 | 159 |  |  |  | 12.127936570 |
| 3 | 92 | 130 | 191 |  |  | 1535.677208161 | 93 | 136 | 177 |  |  | 15.080425383 |
| 4 | 84 | 112 | 142 | 199 |  | 1590.828937490 | 92 | 134 | 168 | 207 |  | 17.885720606 |
| 5 | 77 | 101 | 123 | 151 | 204 | 1618.295354656 | 90 | 130 | 161 | 191 | 222 | 20.451120166 |
| **Cameraman** | 2 | 70 | 144 |  |  |  | 3650.335068357 | 128 | 193 |  |  |  | 12.168753410 |
| 3 | 59 | 119 | 156 |  |  | 3725.715046630 | 44 | 104 | 193 |  |  | 15.227393861 |
| 4 | 42 | 95 | 140 | 170 |  | 3780.686670235 | 44 | 97 | 146 | 197 |  | 18.395542181 |
| 5 | 36 | 82 | 122 | 149 | 173 | 3812.009213462 | 25 | 62 | 100 | 146 | 197 | 21.144610141 |
| **Aerial** | 2 | 125 | 178 |  |  |  | 1808.171050536 | 68 | 159 |  |  |  | 12.538208248 |
| 3 | 109 | 147 | 190 |  |  | 1905.410606582 | 68 | 130 | 186 |  |  | 15.751881495 |
| 4 | 104 | 134 | 167 | 202 |  | 1957.017965982 | 68 | 117 | 159 | 200 |  | 18.615899102 |
| 5 | 99 | 123 | 148 | 175 | 205 | 1980.656737348 | 68 | 108 | 141 | 174 | 207 | 21.210455499 |
| **Barbara** | 2 | 82 | 147 |  |  |  | 2608.610778507 | 96 | 168 |  |  |  | 12.668336540 |
| 3 | 75 | 127 | 176 |  |  | 2785.163280467 | 76 | 127 | 178 |  |  | 15.747087798 |
| 4 | 66 | 106 | 142 | 182 |  | 2856.262131671 | 60 | 99 | 141 | 185 |  | 18.556786861 |
| 5 | 57 | 88 | 118 | 148 | 184 | 2890.976609405 | 58 | 95 | 133 | 172 | 210 | 21.245645311 |
| **Boat** | 2 | 93 | 155 |  |  |  | 1863.346730649 | 107 | 176 |  |  |  | 12.574798244 |
| 3 | 73 | 126 | 167 |  |  | 1994.536306242 | 64 | 119 | 176 |  |  | 15.820902860 |
| 4 | 65 | 114 | 147 | 179 |  | 2059.866280428 | 48 | 88 | 128 | 181 |  | 18.655733570 |
| 5 | 51 | 90 | 126 | 152 | 183 | 2092.775965336 | 48 | 88 | 128 | 174 | 202 | 21.401608305 |
| **Goldhill** | 2 | 94 | 161 |  |  |  | 2069.516570915 | 90 | 157 |  |  |  | 12.546472034 |
| 3 | 83 | 126 | 179 |  |  | 2220.379848566 | 78 | 131 | 177 |  |  | 15.607851663 |
| 4 | 69 | 102 | 138 | 186 |  | 2295.388796334 | 65 | 105 | 147 | 189 |  | 18.414375611 |
| 5 | 63 | 91 | 117 | 147 | 191 | 2331.165415242 | 59 | 95 | 131 | 165 | 199 | 21.099346930 |
| **Lake** | 2 | 85 | 154 |  |  |  | 3974.738214185 | 91 | 163 |  |  |  | 12.520359742 |
| 3 | 78 | 140 | 194 |  |  | 4112.631097687 | 72 | 119 | 169 |  |  | 15.566286745 |
| 4 | 67 | 110 | 158 | 198 |  | 4180.886161109 | 70 | 111 | 155 | 194 |  | 18.365636309 |
| 5 | 57 | 88 | 127 | 166 | 200 | 4216.943583790 | 64 | 99 | 133 | 167 | 199 | 21.024982760 |

**Table 2.** Mean value and variance obtained from eleven original algorithms based on BCV method for seven images over 50 runs

| **Algorithms** | ***d*** | **Zebra** | | **Cameraman** | | **Aerial** | | **Barbara** | | **Boat** | | **Goldhill** | | **Lake** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** |
| **PSO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | 1841.951250256 | 6.69e+002 | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | 1582.004660797 | 4.17e+002 | **3780.686670235** | **1.90e-024** | 1924.812190648 | 4.75e-025 | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1600.162197651 | 1.73e+002 | 3803.979793654 | **2.11e-023** | 1974.941706959 | **1.90e-024** | 2888.199451186 | 9.05e+001 | 2092.771372509 | 1.43e-005 | **2331.165415242** | **2.11e-025** | **4216.943583790** | **8.44e-025** |
| **SFLA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | 1535.626369104 | 5.22e-003 | 3725.625990393 | 4.66e-003 | 1905.267553199 | 2.14e-002 | 2785.086036226 | 1.41e-002 | 1994.427500845 | 6.42e-003 | 2220.307780032 | 6.07e-003 | 4112.541013193 | 8.28e-003 |
| 4 | 1590.192559201 | 1.06e-001 | 3779.841881065 | 2.31e-001 | 1955.784328298 | 4.53e-001 | 2855.248629732 | 3.83e-001 | 2058.902376360 | 4.70e-001 | 2294.652892779 | 1.99e-001 | 4180.139124793 | 1.65e-001 |
| 5 | 1616.879307661 | 5.11e-001 | 3809.670844590 | 9.08e-001 | 1978.045934436 | 1.51e+000 | 2888.401665828 | 1.47e+000 | 2090.757774993 | 9.67e-001 | 2329.157474569 | 4.92e-001 | 4215.004119338 | 9.09e-001 |
| **ABCA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | 1808.163928208 | 3.44e-004 | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | 1508.288244520 | 4.55e+002 | 3718.910069094 | 2.79e+001 | 1847.661277167 | 6.41e+002 | 2774.029322398 | 1.06e+002 | 1984.753346080 | 7.82e+001 | 2206.618034627 | 1.16e+002 | 4103.709524787 | 5.73e+001 |
| 4 | 1541.731285093 | 7.56e+001 | 3752.819581369 | 4.01e+001 | 1915.227567491 | 2.23e+001 | 2824.084310494 | 1.93e+002 | 2034.559185841 | 1.05e+002 | 2266.054535075 | 1.10e+002 | 4163.538580102 | 7.44e+001 |
| 5 | 1577.215555974 | 1.67e+001 | 3796.925268718 | 1.04e+001 | 1948.980850757 | 7.41e+001 | 2851.490095272 | 8.67e+001 | 2060.600981348 | 4.99e+001 | 2299.600412666 | 5.16e+001 | 4197.426299808 | 3.74e+001 |
| **CSO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | 3725.714622769 | 8.98e-006 | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | 1590.820964733 | 2.39e-004 | **3780.686670235** | **1.90e-024** | 1956.964156525 | 3.43e-003 | 2856.259885609 | 2.21e-005 | 2059.866160917 | 2.60e-007 | 2295.388279895 | 8.31e-006 | 4180.883790364 | 4.19e-005 |
| 5 | 1618.236873487 | 1.32e-003 | 3812.007511169 | 2.26e-005 | **1980.374726761** | 1.08e-001 | 2890.964700303 | 2.63e-004 | 2092.760107764 | 3.57e-004 | 2331.132543844 | 8.42e-004 | 4216.928234814 | 4.00e-004 |
| **FA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | 3974.735792612 | 2.93e-004 |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | 1837.511015582 | 4.09e+002 | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | 1924.812190648 | 4.75e-025 | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | 4180.883976390 | 5.60e-005 |
| 5 | **1618.295176818** | **1.58e-006** | 3803.979793654 | **2.11e-023** | 1974.941706959 | **1.90e-024** | **2890.976609405** | **1.35e-023** | 2092.772291075 | 1.49e-005 | 2330.450014686 | 2.56e+001 | **4216.943583790** | **8.44e-025** |
| **CS** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | 1957.016932157 | 4.95e-005 | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.287249606 | 5.91e-004 | 3812.008047353 | 2.24e-005 | 1979.376468845 | 4.80e+000 | 2890.975797693 | 2.91e-005 | **2092.775363595** | **4.25e-006** | 2331.164867954 | 1.37e-005 | 4216.943264382 | 3.90e-006 |

Note: the values in bold face represent the best results.

**Table 2 (Continued).** Mean value and variance obtained from eleven original algorithms based on BCV method for seven images over 50 runs

| **Algorithms** | ***d*** | **Zebra** | | **Cameraman** | | **Aerial** | | **Barbara** | | **Boat** | | **Goldhill** | | **Lake** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** |
| **BA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | 1903.934812405 | 1.09e+002 | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | 1950.576810915 | 1.69e+002 | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.266166182 | 3.08e-003 | 3806.870344563 | 1.52e+001 | 1976.312806878 | 6.08e+000 | **2890.976609405** | **1.35e-023** | 2092.771831792 | 1.49e-005 | **2331.165415242** | **2.11e-025** | **4216.943583790** | **8.44e-025** |
| **GWO** | 2 | 1404.910275676 | 2.45e-005 | 3650.334403758 | 2.21e-005 | 1808.165199740 | 6.94e-004 | 2608.609369542 | 4.86e-005 | 1863.339762181 | 9.11e-004 | 2069.514557169 | 2.03e-004 | 3974.736422078 | 1.61e-004 |
| 3 | 1535.632915229 | 1.87e-002 | 3725.711922862 | 3.47e-004 | 1905.328338899 | 3.34e-002 | 2785.076222404 | 6.21e-002 | 1994.522378685 | 2.80e-003 | 2220.265682793 | 6.84e-002 | 4112.569425375 | 3.27e-002 |
| 4 | 1590.607903667 | 4.04e-001 | 3780.584582060 | 1.11e-001 | 1952.878189886 | 1.10e+002 | 2856.105367606 | 3.01e-001 | 2059.568110085 | 6.33e-001 | 2295.208208981 | 3.08e-001 | 4180.640671422 | 6.21e-001 |
| 5 | 1618.040437321 | 3.38e-001 | 3811.817058211 | 3.61e-001 | 1974.892656778 | 4.30e+001 | 2890.700817521 | 6.91e-001 | 2092.337487805 | 1.09e+000 | 2329.315950553 | 4.98e+001 | 4215.278554761 | 5.35e+001 |
| **ALO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | **1957.017965982** | **0.00e+000** | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.286939665 | 1.01e-003 | 3811.848504400 | 1.29e+000 | 1979.511465545 | 5.34e+000 | 2890.976442173 | 4.61e-007 | 2092.771137384 | 5.64e-005 | 2331.165201556 | 2.12e-007 | 4216.942109238 | 3.16e-005 |
| **WOA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | 3780.686186188 | 1.17e-005 | **1957.017965982** | **0.00e+000** | 2856.261925020 | 2.14e-006 | 2059.866250799 | 4.39e-008 | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.290032981 | 6.94e-004 | **3812.009133018** | 1.58e-007 | 1979.040185154 | 6.86e+000 | 2890.976511451 | 2.35e-007 | 2092.770108129 | 9.37e-005 | 2331.165391499 | 2.82e-008 | 4216.227205655 | 2.57e+001 |
| **MFO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | 1535.674841356 | 1.37e-004 | 3725.710297408 | 1.97e-004 | 1905.409082730 | 4.72e-005 | 2785.162635000 | 1.02e-005 | 1994.531121500 | 2.47e-004 | 2220.377194822 | 6.24e-005 | **4112.631097687** | **2.11e-023** |
| 4 | 1590.813080385 | 7.64e-004 | 3780.672284240 | 1.06e-003 | 1957.006440722 | 5.84e-004 | 2856.252750481 | 2.73e-004 | 2059.866159922 | 3.56e-007 | 2295.383861025 | 2.94e-004 | 4180.865459879 | 1.87e-003 |
| 5 | 1618.192878263 | 8.97e-003 | 3811.690397131 | 2.33e+000 | 1980.247388838 | 1.90e+000 | 2890.941526444 | 8.13e-003 | 2092.747138252 | 1.38e-003 | 2331.019144955 | 1.04e-001 | 4216.877998692 | 1.23e-001 |

**Table 3.** Ranking of mean value from eleven original algorithms based on BCV method

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Statistical results** | **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| \ | **Zebra** | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 11 | 1 | 1 | 1 |
| 3 | 1 | 10 | 11 | 1 | 1 | 1 | 1 | 9 | 1 | 1 | 8 |
| 4 | 10 | 9 | 11 | 6 | 1 | 1 | 1 | 8 | 1 | 1 | 7 |
| 5 | 10 | 9 | 11 | 6 | 1 | 3 | 5 | 8 | 4 | 2 | 7 |
| **Cameraman** | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 11 | 1 | 1 | 1 |
| 3 | 1 | 10 | 11 | 7 | 1 | 1 | 1 | 8 | 1 | 1 | 9 |
| 4 | 1 | 10 | 11 | 1 | 1 | 1 | 1 | 9 | 1 | 7 | 8 |
| 5 | 9 | 7 | 11 | 3 | 9 | 2 | 8 | 5 | 4 | 1 | 6 |
| **Aerial** | 2 | 1 | 1 | 11 | 1 | 1 | 1 | 1 | 10 | 1 | 1 | 1 |
| 3 | 10 | 7 | 9 | 1 | 11 | 1 | 8 | 6 | 1 | 1 | 5 |
| 4 | 9 | 6 | 11 | 5 | 9 | 3 | 8 | 7 | 1 | 1 | 4 |
| 5 | 8 | 6 | 11 | 1 | 8 | 4 | 7 | 10 | 3 | 5 | 2 |
| **Barbara** | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 11 | 1 | 1 | 1 |
| 3 | 1 | 9 | 11 | 1 | 1 | 1 | 1 | 10 | 1 | 1 | 8 |
| 4 | 1 | 10 | 11 | 7 | 1 | 1 | 1 | 9 | 1 | 6 | 8 |
| 5 | 10 | 9 | 11 | 6 | 1 | 5 | 1 | 8 | 4 | 3 | 7 |
| **Boat** | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 11 | 1 | 1 | 1 |
| 3 | 1 | 10 | 11 | 1 | 1 | 1 | 1 | 9 | 1 | 1 | 8 |
| 4 | 1 | 10 | 11 | 7 | 1 | 1 | 1 | 9 | 1 | 6 | 8 |
| 5 | 4 | 10 | 11 | 7 | 2 | 1 | 3 | 9 | 5 | 6 | 8 |
| **Goldhill** | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 11 | 1 | 1 | 1 |
| 3 | 1 | 9 | 11 | 1 | 1 | 1 | 1 | 10 | 1 | 1 | 8 |
| 4 | 1 | 10 | 11 | 7 | 1 | 1 | 1 | 9 | 1 | 1 | 8 |
| 5 | 1 | 10 | 11 | 6 | 8 | 5 | 1 | 9 | 4 | 3 | 7 |
| **Lake** | 2 | 1 | 1 | 1 | 1 | 11 | 1 | 1 | 10 | 1 | 1 | 1 |
| 3 | 1 | 10 | 11 | 1 | 1 | 1 | 1 | 9 | 1 | 1 | 1 |
| 4 | 1 | 10 | 11 | 7 | 6 | 1 | 1 | 9 | 1 | 1 | 8 |
| 5 | 1 | 10 | 11 | 6 | 1 | 4 | 1 | 9 | 5 | 8 | 7 |
| **Total score** | \ | \ | 90 | 198 | 246 | 95 | 84 | 47 | 61 | 254 | 50 | 65 | 149 |
| **Total ranking** | \ | \ | 6 | 9 | 10 | 7 | 5 | 1 | 3 | 11 | 2 | 4 | 8 |

Note: In order to quantify the optimization performance of the compared algorithms, one scoring mechanism is adopted to perform statistical comparison, that is, the best result rank first and thus get score 1, the second best result rank second and thus get score 2, and deduce the rest from this. If two results are the same top 1 best then they both get score 1, and the result next to it get score 3 (=1+2). For example, if a minimum is a better result, than the ranking score of sequence [16 3 34 5 16 6] corresponds to [4 1 6 2 4 3].

**Table 4.** Mean convergence time (in seconds) eleven original algorithms consumed for 50 runs over seven images based on BCV method

| **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.0318 | 0.4722 | 0.3261 | 0.1599 | 0.0681 | 0.1000 | **0.0133** | 0.4753 | 0.3725 | 0.0334 | 0.0779 |
| 3 | 0.0680 | 3.3839 | 0.6148 | 0.5244 | 0.3106 | 0.2421 | **0.0298** | 0.8592 | 0.8249 | 0.0789 | 0.1910 |
| 4 | 0.1213 | 3.2370 | 0.6142 | 2.0089 | 0.7168 | 0.4472 | **0.0761** | 1.0138 | 1.3385 | 0.1255 | 0.2785 |
| 5 | **0.1515** | 2.7900 | 0.6197 | 2.5282 | 1.4438 | 0.7644 | 0.2264 | 1.1129 | 2.0834 | 0.2542 | 0.3915 |
| **Cameraman** | 2 | 0.0342 | 0.5619 | 0.2935 | 0.1649 | 0.0672 | 0.1098 | **0.0122** | 0.3844 | 0.3845 | 0.0507 | 0.0855 |
| 3 | 0.0774 | 3.4547 | 0.6488 | 0.8826 | 0.5262 | 0.2810 | **0.0428** | 0.8854 | 0.8119 | 0.1297 | 0.2107 |
| 4 | 0.1387 | 3.1826 | 0.5037 | 1.4207 | 0.8193 | 0.4103 | **0.0827** | 1.0134 | 1.4319 | 0.3315 | 0.3032 |
| 5 | **0.2453** | 3.1462 | 0.6067 | 2.3579 | 0.6213 | 0.6687 | 0.2659 | 1.1560 | 1.9636 | 0.3939 | 0.4670 |
| **Aerial** | 2 | 0.0412 | 0.7539 | 0.5311 | 0.2376 | 0.1943 | 0.1231 | **0.0161** | 0.5307 | 0.3778 | 0.0363 | 0.0909 |
| 3 | 0.1186 | 3.0573 | 0.4802 | 0.7819 | 0.2348 | 0.2746 | **0.0376** | 0.8118 | 0.8170 | 0.0966 | 0.1865 |
| 4 | **0.0900** | 2.9930 | 0.6056 | 2.1166 | 0.5731 | 0.5809 | 0.1435 | 0.9955 | 1.2814 | 0.1975 | 0.2966 |
| 5 | **0.1957** | 3.2477 | 0.4506 | 2.4597 | 1.0112 | 0.8084 | 0.4112 | 1.0490 | 2.0299 | 0.4199 | 0.4651 |
| **Barbara** | 2 | 0.0332 | 0.6074 | 0.3054 | 0.1588 | 0.0607 | 0.1004 | **0.0128** | 0.3998 | 0.3788 | 0.0324 | 0.0902 |
| 3 | 0.0611 | 3.3726 | 0.4327 | 0.4890 | 0.1617 | 0.2106 | **0.0241** | 0.8000 | 0.8043 | 0.1084 | 0.1841 |
| 4 | 0.1051 | 3.2183 | 0.4660 | 1.8575 | 0.4344 | 0.4167 | **0.0631** | 1.0344 | 1.3731 | 0.2133 | 0.3040 |
| 5 | 0.2359 | 3.4530 | 0.6360 | 2.3799 | 0.7781 | 0.6413 | **0.1909** | 1.1137 | 1.9388 | 0.3369 | 0.4781 |
| **Boat** | 2 | 0.0313 | 0.5600 | 0.3422 | 0.1701 | 0.0601 | 0.1031 | **0.0123** | 0.4263 | 0.3950 | 0.0366 | 0.0893 |
| 3 | 0.0677 | 3.0953 | 0.5332 | 0.5569 | 0.2662 | 0.2463 | **0.0277** | 0.8749 | 0.8114 | 0.0953 | 0.2018 |
| 4 | 0.1139 | 3.2490 | 0.4191 | 1.7956 | 0.4143 | 0.4163 | **0.0646** | 0.9654 | 1.4712 | 0.2281 | 0.2930 |
| 5 | 0.2155 | 2.9166 | 0.6338 | 2.5832 | 0.6971 | 0.7119 | **0.1767** | 1.1248 | 1.8890 | 0.3268 | 0.4561 |
| **Goldhill** | 2 | 0.0354 | 0.4750 | 0.3460 | 0.1475 | 0.0569 | 0.1000 | **0.0127** | 0.4716 | 0.3859 | 0.0354 | 0.0772 |
| 3 | 0.0623 | 3.1460 | 0.5753 | 0.5692 | 0.2953 | 0.2199 | **0.0295** | 0.7780 | 0.7115 | 0.0662 | 0.1826 |
| 4 | 0.1106 | 3.4191 | 0.5605 | 1.7219 | 0.4015 | 0.3798 | **0.0704** | 1.0152 | 1.3823 | 0.2081 | 0.2858 |
| 5 | 0.2428 | 3.5518 | 0.4746 | 2.5539 | 0.6956 | 0.6722 | **0.1819** | 1.0839 | 1.8901 | 0.2666 | 0.4693 |
| **Lake** | 2 | 0.0346 | 0.5969 | 0.3385 | 0.1895 | 0.0529 | 0.1086 | **0.0117** | 0.3967 | 0.3814 | 0.0275 | 0.0896 |
| 3 | 0.0614 | 3.5456 | 0.5579 | 0.4910 | 0.2190 | 0.2351 | **0.0267** | 0.8455 | 0.7773 | 0.0848 | 0.1816 |
| 4 | 0.1039 | 3.0961 | 0.5769 | 1.8834 | 0.4963 | 0.3831 | **0.0559** | 0.9976 | 1.3188 | 0.1989 | 0.3104 |
| 5 | **0.1820** | 2.8788 | 0.5022 | 2.3733 | 0.6287 | 0.5910 | 0.1886 | 1.1199 | 1.9882 | 0.3783 | 0.4303 |

Note: the values in bold face represent the best results.

**Table 5.** Success searching ratio of the eleven original algorithms based on BCV method (%)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| 2 | **100.00** | **100.00** | 98.00 | **100.00** | 99.71 | **100.00** | **100.00** | 96.29 | **100.00** | **100.00** | **100.00** |
| 3 | 87.71 | 22.57 | 0.00 | 99.71 | 86.86 | **100.00** | 99.71 | 84.57 | **100.00** | **100.00** | 92.00 |
| 4 | 83.43 | 0.00 | 0.00 | 78.57 | 84.57 | 99.43 | 97.14 | 82.57 | **100.00** | 99.14 | 76.86 |
| 5 | 51.71 | 0.00 | 0.00 | 28.29 | 64.00 | **80.29** | 66.86 | 42.00 | 78.29 | 87.14 | 35.14 |

Note: Success searching ratio equals to 50 runs divided by the success searching times in 50 runs; values in bold face represent the best results.

**Table 6.** Ranking of each assessment index of the eleven original algorithms based on BCV method

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Statistical results** | **Index** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| \ | **M.V.** | 6 | 9 | 10 | 7 | 5 | 1 | 3 | 11 | 2 | 4 | 8 |
| **Var.** | 5 | 9 | 10 | 7 | 3 | 1 | 4 | 11 | 2 | 6 | 8 |
| **M.C.T.** | 2 | 11 | 7 | 8 | 5 | 6 | 1 | 9 | 10 | 3 | 4 |
| **S.S.R.** | 5 | 9 | 11 | 6 | 8 | 1 | 4 | 10 | 1 | 1 | 7 |
| **Total score** | \ | 18 | 38 | 38 | 28 | 21 | 9 | 12 | 41 | 15 | 14 | 27 |
| **Total ranking** | \ | 5 | 9 | 9 | 8 | 6 | 1 | 2 | 11 | 4 | 3 | 7 |

Note: Same scoring strategy as used in Table 3 is used in Var., M.C.T. and S.S.R. to analysis the statistical performance of the original eleven algorithms.

**Table 7.** Mean value and variance obtained from eleven improved algorithms based on BCV method for seven images over 50 runs

| **Algorithms** | ***d*** | **Zebra** | | **Cameraman** | | **Aerial** | | **Barbara** | | **Boat** | | **Goldhill** | | **Lake** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** |
| **P-PSO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | 1844.902838610 | 8.20e+002 | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | 1924.812190648 | 4.75e-025 | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.293531193 | 1.36e-004 | 3804.140382050 | 1.29e+000 | 1973.949608224 | 4.92e+001 | **2890.976609405** | **1.35e-023** | 2092.771372509 | 1.43e-005 | **2331.165415242** | **2.11e-025** | **4216.943583790** | **8.44e-025** |
| **P-SFLA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | 3725.714511595 | 1.43e-005 | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | 1590.826106292 | 1.28e-004 | **3780.686670235** | **1.90e-024** | 1957.015972581 | 9.73e-005 | 2856.257290962 | 6.01e-005 | **2059.866280428** | **8.44e-025** | 2295.388543879 | 1.56e-006 | **4180.886161109** | **0.00e+000** |
| 5 | 1618.260161054 | 1.76e-003 | **3812.009213462** | **5.28e-024** | 1980.613555260 | 4.12e-003 | 2890.976332294 | 9.01e-007 | 2092.774281300 | 1.03e-005 | 2331.160712833 | 9.62e-005 | 4216.942845763 | 1.63e-005 |
| **P-ABCA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | 3725.708105269 | 3.14e-004 | 1887.691698495 | 1.01e+003 | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | 1925.455049208 | 2.07e+001 | 2856.258205297 | 2.57e-005 | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | 4180.877968413 | 1.60e-004 |
| 5 | 1618.281851725 | 3.89e-004 | 3805.418698106 | 9.70e+000 | 1974.933917469 | 6.93e-004 | **2890.976609405** | **1.35e-023** | 2092.773362734 | 1.34e-005 | 2331.160389303 | 8.17e-005 | 4216.943541255 | 9.05e-008 |
| **P-CSO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | **1957.017965982** | **0.00e+000** | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.290731531 | 1.41e-004 | **3812.009213462** | **5.28e-024** | 1980.629486886 | 1.44e-003 | **2890.976609405** | **1.35e-023** | 2092.775812242 | 1.17e-006 | 2331.164987870 | 3.31e-007 | **4216.943583790** | **8.44e-025** |
| **P-FA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | 1844.899250888 | 8.20e+002 | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | 1924.812190648 | 4.75e-025 | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | 4180.885614929 | 1.49e-005 |
| 5 | **1618.295176818** | **1.58e-006** | 3804.139982844 | 1.28e+000 | 1974.941706959 | **1.90e-024** | **2890.976609405** | **1.35e-023** | 2092.772444169 | 1.49e-005 | 2331.164933309 | 1.16e-005 | **4216.943583790** | **8.44e-025** |
| **P-CS** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | **1957.017965982** | **0.00e+000** | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.294821143 | 4.55e-006 | **3812.009213462** | **5.28e-024** | 1979.623950660 | 4.91e+000 | **2890.976609405** | **1.35e-023** | **2092.775965336** | **8.44e-025** | 2331.165367756 | 5.52e-008 | **4216.943583790** | **8.44e-025** |

Note: the values in bold face represent the best results.

**Table 7 (Continued).** Mean value and variance obtained from eleven improved algorithms based on BCV method for seven images over 50 runs

| **Algorithms** | ***d*** | **Zebra** | | **Cameraman** | | **Aerial** | | **Barbara** | | **Boat** | | **Goldhill** | | **Lake** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** |
| **P-BA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | 1953.797388449 | 9.53e+001 | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.271848861 | 2.57e-003 | 3806.557694474 | 1.43e+001 | 1975.398909390 | 2.45e+000 | **2890.976609405** | **1.35e-023** | 2092.771372509 | 1.43e-005 | **2331.165415242** | **2.11e-025** | **4216.943583790** | **8.44e-025** |
| **P-GWO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | 3725.714511595 | 1.43e-005 | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | 1590.827993757 | 4.45e-005 | **3780.686670235** | **1.90e-024** | 1953.151885907 | 1.12e+002 | 2856.261718369 | 4.18e-006 | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.277354575 | 1.05e-003 | **3812.009213462** | **5.28e-024** | 1977.855977407 | 8.56e+000 | 2890.974855829 | 1.54e-004 | 2092.774587488 | 8.83e-006 | 2330.447572409 | 2.56e+001 | **4216.943583790** | **8.44e-025** |
| **P-ALO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | **1957.017965982** | **0.00e+000** | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.291331891 | 4.36e-004 | 3811.848625066 | 1.29e+000 | 1980.296656057 | 1.87e+000 | **2890.976609405** | **1.35e-023** | 2092.775812242 | 1.17e-006 | 2331.165367756 | 5.52e-008 | **4216.943583790** | **8.44e-025** |
| **P-WOA** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | **1994.536306242** | **2.58e-024** | **2220.379848566** | **1.90e-024** | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | **1957.017965982** | **0.00e+000** | **2856.262131671** | **2.11e-025** | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | **4180.886161109** | **0.00e+000** |
| 5 | 1618.290032981 | 6.94e-004 | **3812.009213462** | **5.28e-024** | 1978.823443630 | 7.29e+000 | **2890.976609405** | **1.35e-023** | 2092.771984886 | 1.49e-005 | 2331.165391499 | 2.82e-008 | **4216.943583790** | **8.44e-025** |
| **P-MFO** | 2 | **1404.910975737** | **3.38e-024** | **3650.335068357** | **1.35e-023** | **1808.171050536** | **5.28e-026** | **2608.610778507** | **3.38e-024** | **1863.346730649** | **0.00e+000** | **2069.516570915** | **2.11e-025** | **3974.738214185** | **1.35e-023** |
| 3 | **1535.677208161** | **5.28e-026** | **3725.715046630** | **1.03e-023** | **1905.410606582** | **1.32e-024** | **2785.163280467** | **3.38e-024** | 1994.535269293 | 5.38e-005 | 2220.379485922 | 6.58e-006 | **4112.631097687** | **2.11e-023** |
| 4 | **1590.828937490** | **1.90e-024** | **3780.686670235** | **1.90e-024** | **1957.017965982** | **0.00e+000** | 2856.260184080 | 4.97e-005 | **2059.866280428** | **8.44e-025** | **2295.388796334** | **5.28e-024** | 4180.885614929 | 1.49e-005 |
| 5 | 1618.253880798 | 2.89e-003 | **3812.009213462** | **5.28e-024** | **1980.641195377** | 2.46e-003 | 2890.976332294 | 9.01e-007 | 2092.772268855 | 4.04e-005 | 2331.162063660 | 7.18e-005 | 4216.942930832 | 1.62e-005 |

**Table 8.** Mean convergence time (in seconds) eleven improved algorithms consumed for 50 runs over seven images based on BCV method

| **Images** | ***d*** | **P-PSO** | **P-SFLA** | **P-ABCA** | **P-CSO** | **P-FA** | **P-CS** | **P-BA** | **P-GWO** | **P-ALO** | **P-WOA** | **P-MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.0058 | 0.0060 | 0.0050 | 0.0058 | 0.0057 | **0.0050** | 0.0058 | 0.0066 | 0.0059 | 0.0061 | 0.0067 |
| 3 | 0.0093 | 0.0097 | 0.0085 | 0.0093 | 0.0093 | **0.0085** | 0.0093 | 0.0102 | 0.0092 | 0.0095 | 0.0101 |
| 4 | 0.0252 | 0.1761 | 0.3115 | 0.1090 | 0.0548 | 0.0519 | **0.0227** | 0.2097 | 0.1250 | 0.0309 | 0.1291 |
| 5 | 0.1819 | 0.9871 | 1.7209 | 1.1427 | 1.6907 | 0.8582 | **0.1518** | 1.1521 | 1.3160 | 0.1831 | 0.2982 |
| **Cameraman** | 2 | 0.0057 | 0.0056 | **0.0046** | 0.0056 | 0.0056 | 0.0046 | 0.0056 | 0.0064 | 0.0056 | 0.0063 | 0.0068 |
| 3 | 0.0419 | 0.5608 | 0.6517 | 0.1482 | 0.1534 | 0.1195 | **0.0182** | 0.2188 | 0.3669 | 0.0209 | 0.1214 |
| 4 | 0.0245 | 0.0221 | 0.0199 | 0.0217 | 0.0216 | 0.0203 | 0.0216 | 0.0227 | 0.0205 | **0.0159** | 0.0163 |
| 5 | 0.1060 | 0.5279 | 1.1543 | 0.2975 | 0.5059 | 0.3628 | 0.2245 | 0.4504 | 0.6377 | **0.0300** | 0.1340 |
| **Aerial** | 2 | 0.0059 | 0.0062 | 0.0058 | 0.0058 | 0.0058 | **0.0049** | 0.0059 | 0.0067 | 0.0062 | 0.0061 | 0.0066 |
| 3 | 0.0881 | 0.0631 | 0.5507 | 0.0538 | 0.0886 | 0.0430 | 0.0409 | 0.0542 | 0.0700 | **0.0180** | 0.0320 |
| 4 | 0.0420 | 0.2109 | 0.4328 | 0.4254 | 0.3079 | 0.2842 | 0.0967 | 0.3641 | 0.2736 | **0.0313** | 0.0931 |
| 5 | **0.1139** | 1.5726 | 1.3622 | 1.5256 | 0.9865 | 0.8633 | 0.1531 | 1.1738 | 1.2433 | 0.2824 | 0.4857 |
| **Barbara** | 2 | 0.0054 | 0.0052 | **0.0041** | 0.0051 | 0.0051 | 0.0041 | 0.0052 | 0.0060 | 0.0055 | 0.0059 | 0.0065 |
| 3 | 0.0077 | 0.0078 | **0.0065** | 0.0076 | 0.0076 | 0.0066 | 0.0076 | 0.0088 | 0.0081 | 0.0087 | 0.0093 |
| 4 | 0.0772 | 0.5267 | 1.0310 | 0.8069 | 0.3665 | 0.3574 | **0.0415** | 0.9781 | 0.8274 | 0.0534 | 0.2282 |
| 5 | 0.0464 | 0.4298 | 1.0481 | 0.1232 | 0.1468 | 0.1503 | **0.0399** | 0.3508 | 0.3880 | 0.0793 | 0.1999 |
| **Boat** | 2 | 0.0054 | 0.0054 | 0.0042 | 0.0053 | 0.0053 | **0.0042** | 0.0053 | 0.0061 | 0.0054 | 0.0059 | 0.0066 |
| 3 | 0.0178 | 0.1213 | 0.1568 | 0.0564 | 0.0469 | 0.0269 | **0.0132** | 0.0433 | 0.1250 | 0.0496 | 0.0217 |
| 4 | 0.0196 | 0.1002 | 0.1853 | 0.0355 | 0.0298 | 0.0224 | **0.0168** | 0.0624 | 0.0671 | 0.0309 | 0.0454 |
| 5 | 0.0559 | 0.7249 | 1.5152 | 0.4913 | 0.3293 | 0.3542 | **0.0329** | 0.5147 | 0.8353 | 0.0346 | 0.1695 |
| **Goldhill** | 2 | 0.0055 | 0.0054 | **0.0044** | 0.0054 | 0.0054 | 0.0046 | 0.0054 | 0.0065 | 0.0054 | 0.0059 | 0.0065 |
| 3 | 0.0204 | 0.1591 | 0.2248 | 0.0510 | 0.0639 | 0.0346 | **0.0121** | 0.0568 | 0.1088 | 0.0178 | 0.0432 |
| 4 | 0.0278 | 0.0797 | 0.2327 | 0.0353 | 0.0355 | 0.0230 | **0.0207** | 0.1170 | 0.1451 | 0.1060 | 0.0425 |
| 5 | 0.1797 | 0.9458 | 1.5608 | 1.1323 | 0.6627 | 0.8928 | **0.1084** | 1.3667 | 1.4212 | 0.1741 | 0.3315 |
| **Lake** | 2 | 0.0134 | 0.1038 | 0.0300 | 0.0244 | 0.0175 | 0.0147 | **0.0071** | 0.0483 | 0.0308 | 0.0080 | 0.0199 |
| 3 | 0.0084 | 0.0086 | 0.0071 | 0.0083 | 0.0083 | **0.0070** | 0.0086 | 0.0093 | 0.0082 | 0.0086 | 0.0092 |
| 4 | 0.0468 | 0.3141 | 0.8905 | 0.1751 | 0.4727 | 0.1504 | 0.0249 | 0.2158 | 0.3180 | **0.0212** | 0.0710 |
| 5 | 0.0329 | 0.6664 | 0.6274 | 0.1495 | 0.0746 | 0.1055 | **0.0317** | 0.5308 | 0.5476 | 0.1301 | 0.3090 |

Note: the values in bold face represent the best results.

**Table 9.** Success searching ratio of the eleven improved algorithms based on BCV method (%)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***d*** | **P-PSO** | **P-SFLA** | **P-ABCA** | **P-CSO** | **P-FA** | **P-CS** | **P-BA** | **P-GWO** | **P-ALO** | **P-WOA** | **P-MFO** |
| 2 | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** | **100.00** |
| 3 | 88.29 | 99.71 | 93.43 | **100.00** | 88.29 | **100.00** | **100.00** | 99.71 | **100.00** | **100.00** | 99.43 |
| 4 | 85.71 | 92.29 | 76.00 | **100.00** | 85.43 | **100.00** | 98.57 | 97.43 | **100.00** | **100.00** | 98.00 |
| 5 | 62.57 | 63.71 | 46.00 | 80.29 | 64.29 | 88.29 | 66.00 | 78.86 | **95.14** | 86.86 | 66.86 |

Note: the values in bold face represent the best results.

**Table 10.** Ranking of each assessment index of the eleven improved algorithms based on BCV method

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Statistical results** | **Index** | **P-PSO** | **P-SFLA** | **P-ABCA** | **P-CSO** | **P-FA** | **P-CS** | **P-BA** | **P-GWO** | **P-ALO** | **P-WOA** | **P-MFO** |
| \ | **M.V.** | 6 | 10 | 11 | 2 | 7 | 1 | 5 | 8 | 3 | 4 | 9 |
| **Var.** | 5 | 11 | 9 | 1 | 7 | 2 | 5 | 8 | 3 | 4 | 10 |
| **M.C.T.** | 3 | 10 | 8 | 6 | 5 | 2 | 1 | 11 | 9 | 4 | 7 |
| **S.S.R.** | 10 | 8 | 11 | 4 | 9 | 2 | 5 | 6 | 1 | 3 | 7 |
| **Total score** | \ | 24 | 39 | 39 | 13 | 28 | 7 | 16 | 33 | 16 | 15 | 33 |
| **Total ranking** | \ | 6 | 10 | 10 | 2 | 7 | 1 | 4 | 8 | 4 | 3 | 8 |

**Table 11.** Difference of the mean value from the improved algorithms and the original algorithms based on BCV method

| **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000700061 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.050839057 | 27.388963641 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.044292932 | 0.000000000 | 0.000000000 | 0.002366805 |
| 4 | 8.824276693 | 0.633547091 | 49.097652397 | 0.007972757 | 0.000000000 | 0.000000000 | 0.000000000 | 0.220090091 | 0.000000000 | 0.000000000 | 0.015857104 |
| 5 | 18.131333542 | 1.380853394 | 41.066295751 | 0.053858044 | 0.000000000 | 0.007571537 | 0.005682679 | 0.236917254 | 0.004392225 | 0.000000000 | 0.061002535 |
| **Cameraman** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000664599 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.088521202 | 6.798036175 | 0.000423861 | 0.000000000 | 0.000000000 | 0.000000000 | 0.002588733 | 0.000000000 | 0.000000000 | 0.004749222 |
| 4 | 0.000000000 | 0.844789169 | 27.867088865 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.102088174 | 0.000000000 | 0.000484047 | 0.014385995 |
| 5 | 0.160588396 | 2.338368872 | 8.493429388 | 0.001702293 | 0.160189190 | 0.001166109 | **-0.312650089** | 0.192155251 | 0.000120666 | 0.000080444 | 0.318816331 |
| **Aerial** | 2 | 0.000000000 | 0.000000000 | 0.007122328 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.005850796 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 2.951588354 | 0.143053382 | 40.030421328 | 0.000000000 | 7.388235306 | 0.000000000 | 1.475794177 | 0.082267683 | 0.000000000 | 0.000000000 | 0.001523852 |
| 4 | 0.000000000 | 1.231644283 | 10.227481718 | 0.053809457 | 0.000000000 | 0.001033826 | 3.220577533 | 0.273696021 | 0.000000000 | 0.000000000 | 0.011525260 |
| 5 | **-0.992098735** | 2.567620824 | 25.953066711 | 0.254760125 | 0.000000000 | 0.247481815 | **-0.913897488** | 2.963320629 | 0.785190512 | **-0.216741524** | 0.393806538 |
| **Barbara** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.001408965 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.077244241 | 11.133958069 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.087058063 | 0.000000000 | 0.000000000 | 0.000645467 |
| 4 | 0.000000000 | 1.008661230 | 32.173894803 | 0.002246063 | 0.000000000 | 0.000000000 | 0.000000000 | 0.156350763 | 0.000000000 | 0.000206651 | 0.007433599 |
| 5 | 2.777158219 | 2.574666465 | 39.486514133 | 0.011909101 | 0.000000000 | 0.000811712 | 0.000000000 | 0.274038308 | 0.000167231 | 0.000097954 | 0.034805850 |
| **Boat** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.006968468 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.108805396 | 9.782960162 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.013927557 | 0.000000000 | 0.000000000 | 0.004147793 |
| 4 | 0.000000000 | 0.963904068 | 25.307094587 | 0.000119511 | 0.000000000 | 0.000000000 | 0.000000000 | 0.298170343 | 0.000000000 | 0.000029629 | 0.000120506 |
| 5 | 0.000000000 | 2.016506307 | 32.172381386 | 0.015704479 | 0.000153094 | 0.000601741 | **-0.000459283** | 0.437099683 | 0.004674858 | 0.001876757 | 0.025130603 |
| **Goldhill** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.002013746 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.072068534 | 13.761813939 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.114165773 | 0.000000000 | 0.000000000 | 0.002291100 |
| 4 | 0.000000000 | 0.735651100 | 29.334261258 | 0.000516439 | 0.000000000 | 0.000000000 | 0.000000000 | 0.180587353 | 0.000000000 | 0.000000000 | 0.004935308 |
| 5 | 0.000000000 | 2.003238264 | 31.559976637 | 0.032444027 | 0.714918624 | 0.000499803 | 0.000000000 | 1.131621856 | 0.000166200 | 0.000000000 | 0.142918705 |
| **Lake** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.002421573 | 0.000000000 | 0.000000000 | 0.001792107 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.090084494 | 8.921572900 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.061672313 | 0.000000000 | 0.000000000 | 0.000000000 |
| 4 | 0.000000000 | 0.747036316 | 17.339388311 | 0.002370745 | 0.001638539 | 0.000000000 | 0.000000000 | 0.245489687 | 0.000000000 | 0.000000000 | 0.020155050 |
| 5 | 0.000000000 | 1.938726425 | 19.517241447 | 0.015348976 | 0.000000000 | 0.000319408 | 0.000000000 | 1.665029029 | 0.001474552 | 0.716378135 | 0.064932141 |

Note: the values in bold face with negative sign represent the results haven’t been improved.

**Table 12.** Difference of the variance from the original algorithms and the improved algorithms based on BCV method

| **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 2.45e-005 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 5.22e-003 | 4.55e+002 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 1.87e-002 | 0.00e+000 | 0.00e+000 | 1.37e-004 |
| 4 | 4.17e+002 | 1.06e-001 | 7.56e+001 | 2.39e-004 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 4.04e-001 | 0.00e+000 | 0.00e+000 | 7.64e-004 |
| 5 | 1.73e+002 | 5.10e-001 | 1.67e+001 | 1.17e-003 | 0.00e+000 | 5.86e-004 | 5.10e-004 | 3.37e-001 | 5.79e-004 | **-1.08e-019** | 6.08e-003 |
| **Cameraman** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 2.21e-005 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 4.65e-003 | 2.79e+001 | 8.98e-006 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 3.32e-004 | 0.00e+000 | 0.00e+000 | 1.97e-004 |
| 4 | 0.00e+000 | 2.31e-001 | 4.01e+001 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 1.11e-001 | 0.00e+000 | 1.17e-005 | 1.06e-003 |
| 5 | **-1.29e+000** | 9.08e-001 | 7.46e-001 | 2.26e-005 | **-1.28e+000** | 2.24e-005 | 8.83e-001 | 3.61e-001 | **-3.93e-005** | 1.58e-007 | 2.33e+000 |
| **Aerial** | 2 | 0.00e+000 | 0.00e+000 | 3.44e-004 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 6.94e-004 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | **-1.51e+002** | 2.14e-002 | **-3.73e+002** | 0.00e+000 | **-4.11e+002** | 0.00e+000 | 1.09e+002 | 3.34e-002 | 0.00e+000 | 0.00e+000 | 4.72e-005 |
| 4 | 0.00e+000 | 4.53e-001 | 1.64e+000 | 3.43e-003 | 0.00e+000 | 4.95e-005 | 7.41e+001 | **-1.42e+000** | 0.00e+000 | 0.00e+000 | 5.84e-004 |
| 5 | **-4.92e+001** | 1.50e+000 | 7.41e+001 | 1.06e-001 | 0.00e+000 | **-1.06e-001** | 3.63e+000 | 3.44e+001 | 3.46e+000 | **-4.30e-001** | 1.89e+000 |
| **Barbara** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 4.86e-005 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 1.41e-002 | 1.06e+002 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 6.21e-002 | 0.00e+000 | 0.00e+000 | 1.02e-005 |
| 4 | 0.00e+000 | 3.83e-001 | 1.93e+002 | 2.21e-005 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 3.01e-001 | 0.00e+000 | 2.14e-006 | 2.24e-004 |
| 5 | 9.05e+001 | 1.47e+000 | 8.67e+001 | 2.63e-004 | 0.00e+000 | 2.91e-005 | 0.00e+000 | 6.91e-001 | 4.61e-007 | 2.35e-007 | 8.13e-003 |
| **Boat** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 9.11e-004 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 6.42e-003 | 7.82e+001 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 2.80e-003 | 0.00e+000 | 0.00e+000 | 1.93e-004 |
| 4 | 0.00e+000 | 4.70e-001 | 1.05e+002 | 2.60e-007 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 6.33e-001 | 0.00e+000 | 4.39e-008 | 3.56e-007 |
| 5 | 0.00e+000 | 9.67e-001 | 4.99e+001 | 3.56e-004 | 7.17e-008 | 4.25e-006 | 5.02e-007 | 1.09e+000 | 5.52e-005 | 7.88e-005 | 1.34e-003 |
| **Goldhill** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 2.03e-004 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 6.07e-003 | 1.16e+002 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 6.84e-002 | 0.00e+000 | 0.00e+000 | 5.58e-005 |
| 4 | 0.00e+000 | 1.99e-001 | 1.10e+002 | 8.31e-006 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 3.08e-001 | 0.00e+000 | 0.00e+000 | 2.94e-004 |
| 5 | 0.00e+000 | 4.92e-001 | 5.16e+001 | 8.41e-004 | 2.56e+001 | 1.37e-005 | 0.00e+000 | 2.43e+001 | 1.57e-007 | 0.00e+000 | 1.04e-001 |
| **Lake** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 2.93e-004 | 0.00e+000 | 0.00e+000 | 1.61e-004 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 8.28e-003 | 5.73e+001 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 3.27e-002 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 4 | 0.00e+000 | 1.65e-001 | 7.44e+001 | 4.19e-005 | 4.11e-005 | 0.00e+000 | 0.00e+000 | 6.21e-001 | 0.00e+000 | 0.00e+000 | 1.85e-003 |
| 5 | 0.00e+000 | 9.09e-001 | 3.74e+001 | 4.00e-004 | 0.00e+000 | 3.90e-006 | 0.00e+000 | 5.35e+001 | 3.16e-005 | 2.57e+001 | 1.23e-001 |

Note: the values in bold face with negative sign represent the results haven’t been improved.

**Table 13.** Difference of the mean convergence time from the original algorithms and the improved algorithms based on BCV method

| **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.0260 | 0.4662 | 0.3211 | 0.1542 | 0.0624 | 0.0951 | 0.0076 | 0.4687 | 0.3666 | 0.0273 | 0.0712 |
| 3 | 0.0587 | 3.3742 | 0.6063 | 0.5151 | 0.3013 | 0.2337 | 0.0205 | 0.8490 | 0.8157 | 0.0694 | 0.1810 |
| 4 | 0.0961 | 3.0609 | 0.3027 | 1.8999 | 0.6620 | 0.3953 | 0.0533 | 0.8041 | 1.2136 | 0.0946 | 0.1494 |
| 5 | **-0.0303** | 1.8028 | **-1.1012** | 1.3855 | **-0.2469** | **-0.0938** | 0.0746 | **-0.0392** | 0.7674 | 0.0711 | 0.0933 |
| **Cameraman** | 2 | 0.0286 | 0.5563 | 0.2889 | 0.1594 | 0.0617 | 0.1052 | 0.0066 | 0.3780 | 0.3789 | 0.0445 | 0.0787 |
| 3 | 0.0356 | 2.8938 | **-0.0028** | 0.7344 | 0.3728 | 0.1615 | 0.0246 | 0.6666 | 0.4450 | 0.1087 | 0.0893 |
| 4 | 0.1142 | 3.1606 | 0.4838 | 1.3991 | 0.7976 | 0.3900 | 0.0611 | 0.9907 | 1.4114 | 0.3157 | 0.2868 |
| 5 | 0.1393 | 2.6183 | **-0.5476** | 2.0604 | 0.1154 | 0.3059 | 0.0415 | 0.7056 | 1.3259 | 0.3639 | 0.3330 |
| **Aerial** | 2 | 0.0353 | 0.7477 | 0.5253 | 0.2317 | 0.1885 | 0.1182 | 0.0102 | 0.5240 | 0.3716 | 0.0302 | 0.0843 |
| 3 | 0.0305 | 2.9943 | **-0.0705** | 0.7281 | 0.1462 | 0.2316 | **-0.0034** | 0.7576 | 0.7470 | 0.0786 | 0.1545 |
| 4 | 0.0481 | 2.7821 | 0.1727 | 1.6911 | 0.2652 | 0.2967 | 0.0469 | 0.6314 | 1.0078 | 0.1662 | 0.2035 |
| 5 | 0.0818 | 1.6751 | **-0.9116** | 0.9341 | 0.0247 | **-0.0549** | 0.2581 | **-0.1249** | 0.7866 | 0.1376 | **-0.0206** |
| **Barbara** | 2 | 0.0278 | 0.6022 | 0.3012 | 0.1537 | 0.0556 | 0.0963 | 0.0076 | 0.3938 | 0.3733 | 0.0265 | 0.0837 |
| 3 | 0.0534 | 3.3648 | 0.4261 | 0.4813 | 0.1542 | 0.2040 | 0.0165 | 0.7912 | 0.7962 | 0.0997 | 0.1748 |
| 4 | 0.0279 | 2.6916 | **-0.5650** | 1.0507 | 0.0679 | 0.0593 | 0.0215 | 0.0564 | 0.5457 | 0.1599 | 0.0758 |
| 5 | 0.1894 | 3.0232 | **-0.4122** | 2.2567 | 0.6313 | 0.4910 | 0.1510 | 0.7629 | 1.5508 | 0.2577 | 0.2782 |
| **Boat** | 2 | 0.0258 | 0.5547 | 0.3380 | 0.1648 | 0.0548 | 0.0989 | 0.0070 | 0.4202 | 0.3896 | 0.0307 | 0.0828 |
| 3 | 0.0499 | 2.9739 | 0.3764 | 0.5004 | 0.2193 | 0.2195 | 0.0145 | 0.8317 | 0.6864 | 0.0457 | 0.1801 |
| 4 | 0.0942 | 3.1489 | 0.2338 | 1.7600 | 0.3846 | 0.3940 | 0.0478 | 0.9030 | 1.4041 | 0.1972 | 0.2475 |
| 5 | 0.1596 | 2.1917 | **-0.8814** | 2.0919 | 0.3678 | 0.3577 | 0.1438 | 0.6101 | 1.0537 | 0.2922 | 0.2866 |
| **Goldhill** | 2 | 0.0299 | 0.4696 | 0.3416 | 0.1421 | 0.0515 | 0.0954 | 0.0073 | 0.4651 | 0.3805 | 0.0295 | 0.0707 |
| 3 | 0.0418 | 2.9869 | 0.3505 | 0.5181 | 0.2314 | 0.1854 | 0.0174 | 0.7211 | 0.6027 | 0.0484 | 0.1394 |
| 4 | 0.0828 | 3.3394 | 0.3278 | 1.6866 | 0.3660 | 0.3569 | 0.0497 | 0.8982 | 1.2372 | 0.1021 | 0.2433 |
| 5 | 0.0631 | 2.6061 | **-1.0863** | 1.4216 | 0.0329 | **-0.2205** | 0.0735 | -0.2828 | 0.4690 | 0.0925 | 0.1378 |
| **Lake** | 2 | 0.0211 | 0.4931 | 0.3086 | 0.1651 | 0.0355 | 0.0939 | 0.0046 | 0.3484 | 0.3506 | 0.0195 | 0.0697 |
| 3 | 0.0530 | 3.5370 | 0.5508 | 0.4827 | 0.2108 | 0.2281 | 0.0181 | 0.8361 | 0.7692 | 0.0762 | 0.1724 |
| 4 | 0.0570 | 2.7819 | **-0.3136** | 1.7083 | 0.0236 | 0.2326 | 0.0310 | 0.7818 | 1.0008 | 0.1777 | 0.2394 |
| 5 | 0.1492 | 2.2124 | **-0.1252** | 2.2238 | 0.5541 | 0.4856 | 0.1569 | 0.5891 | 1.4406 | 0.2482 | 0.1212 |

Note: the values in bold face with negative sign represent the results haven’t been improved.

**Table 14.** Mean value and variance obtained from eleven original algorithms based on KE method for seven images over 50 runs

| **Algorithms** | ***d*** | **Zebra** | | **Cameraman** | | **Aerial** | | **Barbara** | | **Boat** | | **Goldhill** | | **Lake** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** |
| **PSO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | 15.560544082 | 8.08e-004 |
| 4 | 17.824977310 | 4.33e-002 | **18.395542181** | **3.22e-028** | **18.615899102** | **3.22e-028** | 18.555593147 | 7.12e-005 | 18.608762798 | 1.28e-003 | **18.414375611** | **5.15e-029** | 18.364475124 | 6.74e-005 |
| 5 | 19.931182276 | **2.06e-028** | 21.081413428 | 1.67e-003 | 21.050925557 | 6.49e-003 | **21.245645311** | **2.06e-028** | 21.335677285 | 6.12e-003 | **21.099346930** | **3.22e-028** | **21.024978792** | **7.87e-010** |
| **SFLA** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | 15.080001172 | 4.09e-007 | 15.224933033 | 4.54e-006 | 15.751556097 | 3.27e-007 | 15.746911349 | 6.12e-008 | 15.820390107 | 8.92e-007 | 15.607607556 | 8.13e-008 | 15.566019548 | 4.54e-008 |
| 4 | 17.878271191 | 1.91e-005 | 18.380738807 | 5.73e-005 | 18.602135546 | 9.32e-005 | 18.551950503 | 5.92e-006 | 18.640532068 | 4.28e-005 | 18.410240017 | 4.82e-006 | 18.358635026 | 9.74e-006 |
| 5 | 20.422228594 | 7.84e-005 | 21.102841489 | 3.23e-004 | 21.158873212 | 2.83e-004 | 21.223025816 | 8.04e-005 | 21.352232459 | 3.56e-004 | 21.078326918 | 5.73e-005 | 20.992724755 | 2.39e-004 |
| **ABCA** | 2 | **12.127936570** | **3.22e-030** | 12.168736304 | 5.21e-009 | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | 12.574782530 | 1.23e-008 | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | 14.956107082 | 5.28e-003 | 15.184559018 | 6.55e-004 | 15.718543941 | 3.95e-004 | 15.721924434 | 6.45e-004 | 15.787052640 | 6.28e-004 | 15.575652338 | 5.72e-004 | 15.534914057 | 1.42e-003 |
| 4 | 17.289664167 | 8.12e-003 | 18.289373299 | 2.60e-003 | 18.484402910 | 2.69e-003 | 18.459805885 | 2.44e-003 | 18.513746783 | 2.38e-003 | 18.317446191 | 1.66e-003 | 18.178539640 | 3.95e-003 |
| 5 | 19.664134546 | 1.12e-002 | 20.924596315 | 3.14e-003 | 20.840986592 | 5.10e-003 | 20.989888799 | 5.40e-003 | 21.175068040 | 4.77e-003 | 20.935294859 | 4.92e-003 | 20.798268781 | 3.25e-003 |
| **CSO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | 17.885486746 | 1.04e-007 | **18.395542181** | **3.22e-028** | 18.615831090 | 1.88e-008 | 18.556782180 | 2.57e-010 | **18.655733570** | **1.16e-028** | 18.414235629 | 2.30e-008 | 18.365590931 | 1.29e-008 |
| 5 | **20.447733277** | 5.88e-006 | **21.144590084** | **4.04e-009** | 21.209691676 | **4.64e-007** | 21.245412767 | 5.94e-008 | 21.401564335 | 8.99e-009 | 21.098932567 | 1.38e-007 | 21.024410435 | 1.49e-007 |
| **FA** | 2 | 12.127907124 | 6.49e-009 | 12.168536114 | 1.71e-007 | 12.513251106 | 5.56e-004 | **12.668336540** | **2.90e-029** | 12.572412167 | 1.28e-005 | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | 15.080425383 | **1.58e-028** | 15.227235480 | 5.06e-007 | 15.711823983 | 4.28e-004 | **15.747087798** | **2.06e-028** | 15.814812435 | 7.59e-006 | **15.607851663** | **2.61e-028** | 15.560544082 | 8.08e-004 |
| 4 | **17.816472260** | 4.40e-002 | **18.395542181** | **3.22e-028** | 18.566116492 | 1.08e-004 | 18.555591977 | 7.12e-005 | 18.598789745 | 8.96e-004 | 18.414362251 | 8.92e-009 | **18.365636309** | **3.22e-028** |
| 5 | 19.931182276 | **2.06e-028** | 21.077383947 | 1.10e-003 | 21.041728048 | 3.89e-003 | 21.244280634 | 2.19e-005 | 21.300628068 | 5.38e-003 | **21.099346930** | **3.22e-028** | 21.024911347 | 9.25e-009 |
| **CS** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | **18.395542181** | **3.22e-028** | **18.615899102** | **3.22e-028** | **18.556786861** | **6.31e-028** | 18.655660185 | 3.72e-008 | **18.414375611** | **5.15e-029** | **18.365636309** | **3.22e-028** |
| 5 | 20.405679496 | 2.00e-002 | 21.144526128 | 6.10e-008 | **21.210223306** | 1.64e-006 | **21.245645311** | **2.06e-028** | **21.401598534** | **2.34e-009** | 21.099337239 | 4.70e-009 | **21.024978792** | 7.87e-010 |

Note: the values in bold face represent the best results.

**Table 14 (Continued).** Mean value and variance obtained from eleven original algorithms based on KE method for seven images over 50 runs

| **Algorithms** | ***d*** | **Zebra** | | **Cameraman** | | **Aerial** | | **Barbara** | | **Boat** | | **Goldhill** | | **Lake** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** |
| **BA** | 2 | **12.127936570** | **3.22e-030** | 12.168220048 | 2.47e-007 | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | 12.570464415 | 2.08e-005 | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | 15.215469061 | 7.62e-004 | 15.744506077 | 3.10e-004 | **15.747087798** | **2.06e-028** | 15.816376935 | 8.85e-006 | **15.607851663** | **2.61e-028** | 15.563415414 | 4.12e-004 |
| 4 | 17.881539593 | 8.74e-004 | **18.395542181** | **3.22e-028** | 18.591160242 | 6.35e-004 | 18.550818292 | 3.27e-004 | 18.619141070 | 1.35e-003 | **18.414375611** | **5.15e-029** | 18.356346829 | 4.62e-004 |
| 5 | 20.423016232 | 1.03e-002 | 21.082280854 | 1.33e-003 | 21.106839238 | 7.41e-003 | 21.239821404 | 5.25e-005 | 21.331786916 | 6.08e-003 | **21.099346930** | **3.22e-028** | 21.024974825 | 1.54e-009 |
| **GWO** | 2 | 12.127931859 | 3.55e-010 | 12.168744633 | 1.89e-009 | **12.538208248** | **3.22e-030** | 12.668303394 | 4.88e-008 | 12.574646268 | 2.48e-007 | 12.546470716 | 8.68e-011 | 12.520328978 | 1.24e-008 |
| 3 | 15.079868984 | 2.47e-006 | 15.227158931 | 2.76e-006 | 15.751355257 | 4.64e-006 | 15.747015050 | 1.35e-007 | 15.820258797 | 7.24e-006 | 15.607671527 | 2.99e-007 | 15.566263385 | 1.34e-008 |
| 4 | 17.882985795 | 4.09e-005 | 18.392380230 | 7.08e-005 | 18.614225242 | 4.66e-005 | 18.555808547 | 7.63e-006 | 18.654425533 | 2.75e-005 | 18.413301568 | 1.04e-005 | 18.363778447 | 3.17e-005 |
| 5 | 20.435749213 | 2.73e-004 | 21.131711617 | 5.13e-004 | 21.177572098 | 5.25e-003 | 21.239504607 | 2.40e-004 | 21.399024934 | 8.08e-005 | 21.094582716 | 1.60e-004 | 21.023830593 | 1.75e-005 |
| **ALO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | **18.395542181** | **3.22e-028** | **18.615899102** | **3.22e-028** | **18.556786861** | **6.31e-028** | 18.640948755 | 8.00e-004 | **18.414375611** | **5.15e-029** | 18.364475124 | 6.74e-005 |
| 5 | 20.446575540 | 2.35e-005 | 21.115760942 | 1.99e-003 | 21.174527051 | 5.99e-003 | 21.245621446 | 2.32e-009 | 21.398148474 | 5.01e-004 | 21.099320820 | 1.12e-008 | 21.024886744 | 1.52e-008 |
| **WOA** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | 12.574546819 | 1.37e-007 | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | 15.220995273 | 4.13e-004 | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | 15.820902710 | 1.13e-012 | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | 18.395367476 | 7.47e-007 | 18.615892971 | 9.21e-010 | **18.556786861** | **6.31e-028** | 18.651958566 | 1.91e-004 | **18.414375611** | **5.15e-029** | **18.365636309** | **3.22e-028** |
| 5 | 20.445015465 | 2.20e-005 | 21.136902726 | 6.40e-004 | 21.206259967 | 1.78e-004 | 21.245638151 | 8.19e-010 | 21.395244687 | 8.68e-005 | **21.099346930** | **3.22e-028** | 21.024951021 | 5.40e-009 |
| **MFO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | 15.751877262 | 8.96e-010 | 15.747081361 | 1.86e-009 | 15.820788509 | 3.19e-007 | 15.607830177 | 8.42e-009 | **15.566286745** | **1.58e-028** |
| 4 | 17.885698044 | 2.55e-008 | 18.395518870 | 9.52e-009 | 18.615873752 | 7.44e-009 | 18.556698921 | 6.74e-008 | 18.648286293 | 4.89e-004 | 18.414053980 | 4.79e-007 | 18.365593519 | 4.62e-008 |
| 5 | 20.445304645 | 4.02e-005 | 21.122941753 | 1.52e-003 | 21.200910759 | 1.59e-003 | 21.244885354 | 4.64e-006 | 21.394998475 | 8.72e-004 | 21.097787324 | 3.53e-005 | 21.024133559 | 2.65e-006 |

Note: the values in bold face represent the best results.

**Table 15.** Mean convergence time (in seconds) eleven original algorithms consumed for 50 runs over seven images based on KE method

| **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.1360 | 2.0058 | 1.1667 | 0.6628 | 0.1304 | 0.4419 | **0.0592** | 1.7569 | 0.7458 | 0.2039 | 0.3733 |
| 3 | 0.2651 | 14.7373 | 1.9930 | 1.8387 | 0.7184 | 1.0101 | **0.1555** | 3.0660 | 1.4799 | 0.5496 | 0.7066 |
| 4 | 0.4947 | 13.8032 | 2.0463 | 7.3364 | 1.5536 | 1.8064 | **0.3498** | 3.2572 | 2.3446 | 0.8762 | 1.0516 |
| 5 | **0.6654** | 12.2395 | 1.8763 | 8.5157 | 1.2205 | 2.8515 | 0.9142 | 3.1235 | 3.2329 | 1.3252 | 1.3736 |
| **Cameraman** | 2 | 0.2198 | 3.2328 | 2.0898 | 0.8080 | 0.6165 | 0.6912 | **0.0949** | 1.9587 | 0.7833 | 0.2740 | 0.4101 |
| 3 | 0.2689 | 14.9910 | 2.0129 | 1.3504 | 0.5417 | 1.0795 | **0.2625** | 3.2860 | 1.4281 | 1.3364 | 0.7904 |
| 4 | 0.3677 | 13.9963 | 1.8931 | 4.1004 | 0.4099 | 1.4319 | **0.2663** | 3.4640 | 2.3857 | 1.0102 | 1.0826 |
| 5 | **0.8021** | 13.0607 | 2.1348 | 6.6052 | 1.0210 | 2.8705 | 0.8528 | 3.3129 | 3.0305 | 1.9761 | 1.3779 |
| **Aerial** | 2 | **0.1466** | 2.8716 | 1.2584 | 0.6586 | 0.1567 | 0.5628 | 0.2215 | 1.6290 | 0.8032 | 0.5266 | 0.3830 |
| 3 | 0.2294 | 14.8640 | 2.1016 | 1.5982 | **0.1938** | 0.8898 | 0.3500 | 3.1568 | 1.3917 | 0.5488 | 0.7765 |
| 4 | 0.5602 | 12.1943 | 1.8873 | 6.2897 | 0.5002 | 1.6673 | **0.4837** | 3.5299 | 2.1668 | 1.4019 | 1.1048 |
| 5 | 1.1038 | 11.9802 | 2.0466 | 8.0719 | 1.1620 | 2.8954 | **0.8356** | 3.5918 | 3.0951 | 1.7531 | 1.4706 |
| **Barbara** | 2 | 0.1413 | 1.9497 | 1.0155 | 0.6238 | 0.1692 | 0.4625 | **0.0579** | 1.8545 | 0.7519 | 0.2270 | 0.3684 |
| 3 | 0.2504 | 15.2777 | 1.8095 | 2.0414 | 0.3788 | 0.9358 | **0.0989** | 3.2830 | 1.4375 | 0.3290 | 0.7377 |
| 4 | 0.3756 | 12.3980 | 1.5789 | 5.6904 | 0.7084 | 1.5960 | **0.2436** | 3.3176 | 2.2347 | 0.6947 | 1.2505 |
| 5 | 0.8721 | 11.9489 | 2.0250 | 7.8322 | 1.2271 | 2.1379 | **0.6972** | 3.4232 | 3.0608 | 1.4345 | 1.4955 |
| **Boat** | 2 | 0.2016 | 2.7256 | 1.2259 | 0.6124 | 0.2453 | 0.5377 | **0.1661** | 1.9222 | 0.7571 | 0.8225 | 0.4054 |
| 3 | 0.2864 | 14.7790 | 2.0021 | 1.5474 | 0.3403 | 1.1020 | **0.1741** | 3.1404 | 1.4459 | 1.1162 | 0.8726 |
| 4 | 0.6796 | 13.1269 | 1.9929 | 4.9708 | 0.8209 | 2.4856 | **0.4064** | 3.6185 | 2.3817 | 1.7947 | 1.2150 |
| 5 | 1.0040 | 11.7451 | 1.9525 | 7.4574 | 0.8011 | 2.3981 | **0.6927** | 3.7934 | 3.2279 | 1.9233 | 1.4815 |
| **Goldhill** | 2 | 0.1355 | 2.0144 | 1.1079 | 0.6947 | 0.1297 | 0.4806 | **0.0578** | 1.8562 | 0.7642 | 0.2687 | 0.3437 |
| 3 | 0.2512 | 14.6472 | 2.0799 | 1.9727 | 0.4792 | 0.9199 | **0.1163** | 3.0577 | 1.2769 | 0.5788 | 0.7172 |
| 4 | 0.3614 | 12.5234 | 1.7219 | 6.4058 | 1.0188 | 1.5091 | **0.3517** | 3.2246 | 2.1747 | 0.7054 | 1.1881 |
| 5 | 0.7998 | 11.6871 | 1.7635 | 8.0447 | 1.3381 | 2.1879 | **0.5570** | 3.4584 | 2.9507 | 1.0582 | 1.5200 |
| **Lake** | 2 | 0.1395 | 2.5087 | 1.1344 | 0.6946 | 0.1393 | 0.4818 | **0.0612** | 1.5982 | 0.8068 | 0.1736 | 0.3722 |
| 3 | 0.2330 | 14.5040 | 1.7863 | 1.8363 | 0.3517 | 0.9757 | **0.1112** | 3.3996 | 1.5051 | 0.4342 | 0.7661 |
| 4 | 0.4067 | 12.0764 | 2.2306 | 5.9644 | 0.8470 | 1.5826 | **0.3359** | 3.4347 | 2.2449 | 0.9359 | 1.0788 |
| 5 | **0.9452** | 11.9707 | 1.9581 | 8.7317 | 1.8823 | 2.3485 | 1.2056 | 3.7420 | 3.0989 | 1.3426 | 1.4740 |

Note: the values in bold face represent the best results.

**Table 16.** Success searching ratio of the eleven original algorithms based on KE method (%)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| 2 | **100.00** | **100.00** | 98.86 | **100.00** | 74.86 | **100.00** | 80.57 | 94.57 | **100.00** | 95.43 | **100.00** |
| 3 | 99.43 | 35.43 | 0.29 | **100.00** | 73.14 | **100.00** | 83.14 | 91.14 | **100.00** | 94.29 | 97.43 |
| 4 | 89.14 | 0.00 | 0.00 | 80.57 | 68.57 | **97.14** | 75.71 | 70.29 | 88.57 | 89.43 | 78.86 |
| 5 | 56.29 | 0.00 | 0.00 | 36.00 | 43.43 | **79.14** | 45.43 | 36.86 | 63.71 | 69.71 | 47.71 |

Note: the values in bold face represent the best results.

**Table 17.** Ranking of each assessment index of the eleven original algorithms based on KE method

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Statistical results** | **Index** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| \ | **M.V.** | 5 | 9 | 11 | 3 | 8 | 1 | 7 | 10 | 2 | 4 | 6 |
| **Var.** | 5 | 7 | 11 | 2 | 8 | 1 | 9 | 10 | 4 | 3 | 6 |
| **M.C.T.** | 2 | 11 | 7 | 9 | 3 | 6 | 1 | 10 | 8 | 4 | 5 |
| **S.S.R.** | 3 | 7 | 11 | 4 | 10 | 1 | 7 | 9 | 2 | 6 | 5 |
| **Total score** | \ | 15 | 34 | 40 | 18 | 29 | 9 | 24 | 39 | 16 | 17 | 22 |
| **Total ranking** | \ | 2 | 9 | 11 | 5 | 8 | 1 | 7 | 10 | 3 | 4 | 6 |

**Table 18.** Mean value and variance obtained from eleven improved algorithms based on KE method for seven images over 50 runs

| **Algorithms** | ***d*** | **Zebra** | | **Cameraman** | | **Aerial** | | **Barbara** | | **Boat** | | **Goldhill** | | **Lake** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** |
| **P-PSO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | 15.224412661 | 2.18e-004 | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | 15.554801418 | 1.55e-003 |
| 4 | 17.824977310 | 4.33e-002 | **18.395542181** | **3.22e-028** | 18.614857442 | 5.43e-005 | 18.554399434 | 1.40e-004 | 18.604961457 | 2.08e-003 | **18.414375611** | **5.15e-029** | 18.364475124 | 6.74e-005 |
| 5 | 19.951590479 | 1.02e-002 | 21.082992143 | 1.44e-003 | 21.078836744 | 9.10e-003 | **21.245645311** | **2.06e-028** | 21.345573092 | 5.70e-003 | **21.099346930** | **3.22e-028** | **21.024982760** | **0.00e+000** |
| **P-SFLA** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | 15.227322570 | 2.54e-007 | **15.751881495** | **2.90e-029** | 15.747086412 | 2.26e-011 | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | 18.395334483 | 1.25e-006 | **18.615899102** | **3.22e-028** | 18.556774473 | 3.15e-009 | 18.654804380 | 1.07e-005 | 18.414155964 | 8.85e-008 | **18.365636309** | **3.22e-028** |
| 5 | **20.450338475** | **3.76e-006** | 21.143182917 | 7.04e-006 | 21.208740629 | 6.35e-005 | 21.245402288 | 9.44e-007 | 21.400526052 | 1.19e-005 | 21.099239269 | 2.51e-008 | 21.024915315 | 9.01e-009 |
| **P-ABCA** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | 15.227378062 | 1.25e-008 | 15.751043797 | 3.51e-005 | **15.747087798** | **2.06e-028** | 15.820600044 | 8.54e-007 | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | 17.885672932 | 1.14e-007 | 18.395482009 | 1.81e-007 | 18.591682243 | 5.24e-004 | 18.556784520 | 1.34e-010 | 18.652084719 | 1.34e-004 | 18.414322953 | 2.22e-008 | **18.365636309** | **3.22e-028** |
| 5 | 20.144061654 | 6.39e-002 | 21.139327938 | 1.54e-005 | 21.159903260 | 3.65e-004 | **21.245645311** | **2.06e-028** | 21.401473888 | 9.03e-007 | 21.099286371 | 1.70e-008 | 21.024804228 | 3.61e-009 |
| **P-CSO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | **18.395542181** | **3.22e-028** | **18.615899102** | **3.22e-028** | **18.556786861** | **6.31e-028** | **18.655733570** | **1.16e-028** | **18.414375611** | **5.15e-029** | **18.365636309** | **3.22e-028** |
| 5 | 20.450305635 | 3.78e-006 | 21.144608578 | 1.22e-010 | 21.210430135 | 7.62e-009 | **21.245645311** | **2.06e-028** | **21.401608305** | **5.15e-029** | 21.099333472 | 4.44e-009 | 21.024962923 | 3.61e-009 |
| **P-FA** | 2 | 12.127926754 | 2.36e-009 | 12.168733656 | 1.95e-008 | 12.518163848 | 5.32e-004 | **12.668336540** | **2.90e-029** | 12.573341922 | 8.21e-006 | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | 15.725151150 | 5.90e-004 | **15.747087798** | **2.06e-028** | 15.817651672 | 8.69e-006 | **15.607851663** | **2.61e-028** | 15.560544082 | 8.08e-004 |
| 4 | 17.824738839 | 4.33e-002 | **18.395542181** | **3.22e-028** | 18.568026594 | 1.60e-004 | 18.555593147 | 7.12e-005 | 18.604726095 | 9.18e-004 | 18.414362251 | 8.92e-009 | **18.365636309** | **3.22e-028** |
| 5 | 19.951314120 | 9.93e-003 | 21.081799088 | 1.28e-003 | 21.056334091 | 4.57e-003 | **21.245645311** | **2.06e-028** | 21.340761640 | 5.35e-003 | **21.099346930** | **3.22e-028** | 21.024947053 | 5.93e-009 |
| **P-CS** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | **18.395542181** | **3.22e-028** | **18.615899102** | **3.22e-028** | **18.556786861** | **6.31e-028** | 18.655723798 | 2.34e-009 | **18.414375611** | **5.15e-029** | **18.365636309** | **3.22e-028** |
| 5 | 20.396761567 | 2.46e-002 | **21.144610141** | **1.16e-028** | **21.210448970** | **1.04e-009** | **21.245645311** | **2.06e-028** | **21.401608305** | **5.15e-029** | **21.099346930** | **3.22e-028** | **21.024982760** | **0.00e+000** |

Note: the values in bold face represent the best results.

**Table 18 (Continued).** Mean value and variance obtained from eleven improved algorithms based on KE method for seven images over 50 runs

| **Algorithms** | ***d*** | **Zebra** | | **Cameraman** | | **Aerial** | | **Barbara** | | **Boat** | | **Goldhill** | | **Lake** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** | **M.V.** | **Var.** |
| **P-BA** | 2 | **12.127936570** | **3.22e-030** | 12.168654639 | 8.96e-008 | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | 12.570460308 | 1.94e-005 | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | 15.215469061 | 7.62e-004 | 15.744977437 | 2.71e-004 | **15.747087798** | **2.06e-028** | 15.817012672 | 1.21e-005 | **15.607851663** | **2.61e-028** | 15.560544082 | 8.08e-004 |
| 4 | 17.877367915 | 1.71e-003 | 18.395482009 | 1.81e-007 | 18.579919083 | 5.27e-004 | 18.553205720 | 2.05e-004 | 18.600848171 | 1.85e-003 | **18.414375611** | **5.15e-029** | 18.357508014 | 4.14e-004 |
| 5 | 20.429983279 | 5.31e-003 | 21.081829164 | 1.35e-003 | 21.076058984 | 8.51e-003 | 21.241985116 | 3.28e-005 | 21.348329248 | 5.37e-003 | **21.099346930** | **3.22e-028** | 21.018929274 | 1.82e-003 |
| **P-GWO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | 15.820731558 | 4.69e-007 | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | 18.395334483 | 1.25e-006 | **18.615899102** | **3.22e-028** | 18.556783350 | 1.97e-010 | 18.655667208 | 8.16e-008 | 18.414306548 | 5.41e-008 | **18.365636309** | **3.22e-028** |
| 5 | 20.449764167 | 1.00e-005 | 21.143787182 | 4.47e-006 | 21.187284048 | 3.83e-003 | 21.245638151 | 8.19e-010 | 21.401569730 | 3.13e-008 | 21.099320015 | 8.50e-009 | 21.024978792 | 7.87e-010 |
| **P-ALO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | **18.395542181** | **3.22e-028** | **18.615899102** | **3.22e-028** | **18.556786861** | **6.31e-028** | 18.642929686 | 7.47e-004 | **18.414375611** | **5.15e-029** | **18.365636309** | **3.22e-028** |
| 5 | 20.449070098 | 1.52e-005 | 21.117909909 | 1.73e-003 | 21.180487195 | 4.87e-003 | **21.245645311** | **2.06e-028** | 21.398592871 | 4.49e-004 | 21.099340201 | 2.26e-009 | **21.024982760** | **0.00e+000** |
| **P-WOA** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | 12.574609675 | 1.15e-007 | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | 15.215284776 | 7.58e-004 | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | **15.820902860** | **8.05e-029** | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | 18.395334483 | 7.15e-007 | **18.615899102** | **3.22e-028** | **18.556786861** | **6.31e-028** | 18.653024300 | 1.38e-004 | **18.414375611** | **5.15e-029** | **18.365636309** | **3.22e-028** |
| 5 | 20.446343270 | 2.32e-005 | 21.128580163 | 1.22e-003 | 21.209424336 | 5.22e-005 | **21.245645311** | **2.06e-028** | 21.398494971 | 4.43e-005 | **21.099346930** | **3.22e-028** | 21.024907380 | 9.46e-009 |
| **P-MFO** | 2 | **12.127936570** | **3.22e-030** | **12.168753410** | **3.22e-030** | **12.538208248** | **3.22e-030** | **12.668336540** | **2.90e-029** | **12.574798244** | **2.06e-028** | **12.546472034** | **1.29e-029** | **12.520359742** | **2.90e-029** |
| 3 | **15.080425383** | **1.58e-028** | **15.227393861** | **2.06e-028** | **15.751881495** | **2.90e-029** | **15.747087798** | **2.06e-028** | 15.820845759 | 1.63e-007 | **15.607851663** | **2.61e-028** | **15.566286745** | **1.58e-028** |
| 4 | **17.885720606** | **6.31e-028** | **18.395542181** | **3.22e-028** | **18.615899102** | **3.22e-028** | 18.556785691 | 6.85e-011 | 18.650116463 | 3.15e-004 | 18.414256873 | 6.33e-008 | **18.365636309** | **3.22e-028** |
| 5 | 20.448460780 | 2.16e-005 | 21.135715233 | 6.17e-004 | 21.206442933 | 7.95e-004 | 21.245640538 | 5.58e-010 | 21.396682006 | 8.73e-004 | 21.099317053 | 1.08e-008 | 21.024948937 | 6.32e-009 |

**Table 19.** Mean convergence time (in seconds) eleven improved algorithms consumed for 50 runs over seven images based on KE method

| **Images** | ***d*** | **P-PSO** | **P-SFLA** | **P-ABCA** | **P-CSO** | **P-FA** | **P-CS** | **P-BA** | **P-GWO** | **P-ALO** | **P-WOA** | **P-MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.0530 | 0.2830 | 0.1389 | 0.1550 | 0.0501 | 0.1011 | **0.0427** | 0.1540 | 0.1624 | 0.0474 | 0.0887 |
| 3 | 0.0538 | 0.0593 | 0.2221 | 0.0720 | 0.0553 | 0.0466 | 0.0528 | **0.0455** | 0.0491 | 0.0524 | 0.0796 |
| 4 | 0.1758 | 0.7794 | 3.2904 | 0.2884 | 0.2901 | 0.3067 | 0.2131 | 0.5518 | 0.4442 | **0.1151** | 0.3386 |
| 5 | **0.1603** | 4.4981 | 4.4672 | 5.0918 | 0.4246 | 2.3054 | 0.5948 | 2.2798 | 0.8476 | 0.5279 | 0.6841 |
| **Cameraman** | 2 | 0.0422 | 0.0680 | 0.0733 | 0.0826 | 0.0441 | **0.0401** | 0.0625 | 0.1111 | 0.0885 | 0.0502 | 0.0668 |
| 3 | **0.1121** | 2.6897 | 2.2465 | 0.4489 | 0.3115 | 0.4602 | 0.2495 | 1.3317 | 0.8295 | 0.9497 | 0.5334 |
| 4 | **0.0896** | 2.3001 | 1.2483 | 0.3307 | 0.1880 | 0.2703 | 0.2331 | 1.4465 | 0.7870 | 0.5676 | 0.4462 |
| 5 | **0.2804** | 8.4137 | 5.2638 | 3.4922 | 0.7181 | 2.7142 | 0.5510 | 4.7651 | 1.8896 | 1.8197 | 1.3612 |
| **Aerial** | 2 | 0.1005 | 0.8632 | 0.5896 | 0.3012 | **0.0554** | 0.3363 | 0.1798 | 0.4988 | 0.5153 | 0.1846 | 0.1874 |
| 3 | 0.0735 | 0.3512 | 1.1955 | 0.1688 | **0.0499** | 0.1807 | 0.3882 | 0.2351 | 0.4848 | 0.1991 | 0.2256 |
| 4 | 0.3993 | 2.6717 | 3.7793 | 0.7413 | **0.1348** | 0.7355 | 0.3941 | 1.0936 | 1.0488 | 0.4066 | 0.6213 |
| 5 | 0.3584 | 7.9481 | 5.5952 | 4.0487 | **0.3249** | 2.4190 | 0.5771 | 4.6470 | 1.7823 | 0.8378 | 1.6336 |
| **Barbara** | 2 | 0.0279 | 0.0283 | 0.0214 | 0.0276 | 0.0273 | **0.0212** | 0.0273 | 0.0285 | 0.0275 | 0.0311 | 0.0315 |
| 3 | 0.0844 | 1.0917 | 1.1546 | 0.3797 | 0.1492 | 0.2167 | **0.0546** | 0.7488 | 0.4403 | 0.0759 | 0.1885 |
| 4 | **0.1648** | 1.2279 | 2.3471 | 0.6200 | 0.4490 | 0.5348 | 0.2051 | 1.1626 | 0.8685 | 0.2468 | 0.6185 |
| 5 | **0.1181** | 1.0140 | 1.5867 | 0.4408 | 0.1429 | 0.3233 | 0.1603 | 1.4306 | 0.9349 | 0.4056 | 1.4206 |
| **Boat** | 2 | 0.1306 | 1.3720 | 0.7804 | 0.3263 | 0.1125 | 0.3166 | **0.0816** | 0.7901 | 0.5310 | 0.7230 | 0.3196 |
| 3 | 0.1679 | 2.3088 | 1.9219 | 0.4187 | **0.0983** | 0.5253 | 0.1088 | 1.3238 | 0.6838 | 0.4746 | 0.3959 |
| 4 | 0.4457 | 4.5194 | 5.0593 | 1.9149 | 0.2804 | 1.4349 | **0.1399** | 3.3852 | 1.4294 | 1.3211 | 0.9899 |
| 5 | 0.1730 | 4.4879 | 2.7863 | 0.6179 | **0.1126** | 0.6240 | 0.2600 | 2.6395 | 1.5377 | 1.2711 | 1.4242 |
| **Goldhill** | 2 | 0.0270 | 0.0271 | **0.0214** | 0.0268 | 0.0268 | 0.0216 | 0.0268 | 0.0283 | 0.0270 | 0.0281 | 0.0287 |
| 3 | 0.0663 | 0.3319 | 0.4227 | 0.1681 | 0.0878 | 0.1025 | **0.0441** | 0.5776 | 0.3570 | 0.0598 | 0.1708 |
| 4 | **0.2121** | 3.5903 | 3.5377 | 1.1076 | 0.6726 | 0.8378 | 0.2551 | 3.2735 | 1.6056 | 0.3710 | 1.2304 |
| 5 | 0.2818 | 1.8204 | 4.7227 | 2.7037 | 0.7707 | 1.1882 | **0.1924** | 3.0901 | 1.9587 | 0.2987 | 1.2900 |
| **Lake** | 2 | 0.0345 | 0.1409 | **0.0251** | 0.0458 | 0.0346 | 0.0263 | 0.0303 | 0.0726 | 0.1157 | 0.0338 | 0.0453 |
| 3 | 0.0387 | 0.0703 | 0.3904 | 0.0488 | **0.0380** | 0.0583 | 0.0448 | 0.0430 | 0.0498 | 0.0569 | 0.0818 |
| 4 | 0.0814 | 0.0893 | 0.2788 | 0.0892 | **0.0609** | 0.0764 | 0.1405 | 0.0667 | 0.1434 | 0.0728 | 0.1416 |
| 5 | **0.3951** | 3.0797 | 0.5448 | 1.8556 | 1.7006 | 2.2993 | 0.3988 | 5.1606 | 1.9962 | 0.7219 | 1.3803 |

Note: the values in bold face represent the best results.

**Table 20.** Success searching ratio of the eleven improved algorithms based on KE method(%)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***d*** | **P-PSO** | **P-SFLA** | **P-ABCA** | **P-CSO** | **P-FA** | **P-CS** | **P-BA** | **P-GWO** | **P-ALO** | **P-WOA** | **P-MFO** |
| 2 | **100.00** | **100.00** | **100.00** | **100.00** | 82.57 | **100.00** | 87.71 | **100.00** | **100.00** | 96.57 | **100.00** |
| 3 | 98.29 | 98.57 | 98.00 | **100.00** | 82.00 | **100.00** | 85.14 | 99.14 | **100.00** | 96.00 | 99.71 |
| 4 | 88.29 | 84.57 | 82.00 | **100.00** | 70.86 | 99.43 | 72.00 | 95.71 | 95.71 | 89.71 | 92.86 |
| 5 | 60.00 | 68.29 | 43.71 | 90.86 | 49.71 | **92.29** | 44.86 | 86.86 | 86.29 | 71.71 | 82.29 |

Note: the values in bold face represent the best result.

**Table 21.** Ranking of each assessment index of the eleven improved algorithms based on KE method

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Statistical results** | **Index** | **P-PSO** | **P-SFLA** | **P-ABCA** | **P-CSO** | **P-FA** | **P-CS** | **P-BA** | **P-GWO** | **P-ALO** | **P-WOA** | **P-MFO** |
| \ | **M.V.** | 8 | 7 | 9 | 2 | 10 | 1 | 11 | 5 | 3 | 4 | 6 |
| **Var.** | 9 | 7 | 8 | 2 | 10 | 1 | 11 | 5 | 3 | 4 | 6 |
| **M.C.T.** | 1 | 11 | 10 | 6 | 2 | 5 | 3 | 9 | 8 | 4 | 6 |
| **S.S.R.** | 7 | 6 | 8 | 1 | 11 | 1 | 10 | 4 | 3 | 9 | 5 |
| **Total score** | \ | 25 | 31 | 35 | 11 | 33 | 8 | 35 | 23 | 17 | 21 | 23 |
| **Total ranking** | \ | 7 | 8 | 10 | 2 | 9 | 1 | 10 | 5 | 3 | 4 | 5 |

**Table 22.** Difference of the mean value from the improved algorithms and the original algorithms based on KE method

| **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000019630 | 0.000000000 | 0.000000000 | 0.000004711 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.000424211 | 0.124318301 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000556399 | 0.000000000 | 0.000000000 | 0.000000000 |
| 4 | 0.000000000 | 0.007449415 | 0.596008765 | 0.000233860 | 0.008266578 | 0.000000000 | **-0.004171677** | 0.002734811 | 0.000000000 | 0.000000000 | 0.000022562 |
| 5 | 0.020408203 | 0.028109881 | 0.479927108 | 0.002572358 | 0.020131844 | **-0.008917929** | 0.006967047 | 0.014014953 | 0.002494558 | 0.001327805 | 0.003156134 |
| **Cameraman** | 2 | 0.000000000 | 0.000000000 | 0.000017106 | 0.000000000 | 0.000197541 | 0.000000000 | 0.000434591 | 0.000008777 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | **-0.002981200** | 0.002389537 | 0.042819044 | 0.000000000 | 0.000158381 | 0.000000000 | 0.000000000 | 0.000234929 | 0.000000000 | **-0.005710497** | 0.000000000 |
| 4 | 0.000000000 | 0.014595676 | 0.106108710 | 0.000000000 | 0.000000000 | 0.000000000 | **-0.000060173** | 0.002954253 | 0.000000000 | **-0.000032993** | 0.000023311 |
| 5 | 0.001578716 | 0.040341428 | 0.214731623 | 0.000018493 | 0.004415141 | 0.000084014 | **-0.000451689** | 0.012075566 | 0.002148967 | **-0.008322563** | 0.012773481 |
| **Aerial** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.004912741 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.000325398 | 0.032499856 | 0.000000000 | 0.013327167 | 0.000000000 | 0.000471360 | 0.000526238 | 0.000000000 | 0.000000000 | 0.000004233 |
| 4 | **-0.001041661** | 0.013763556 | 0.107279332 | 0.000068012 | 0.001910101 | 0.000000000 | **-0.011241159** | 0.001673860 | 0.000000000 | 0.000006132 | 0.000025350 |
| 5 | 0.027911187 | 0.049867417 | 0.318916669 | 0.000738460 | 0.014606043 | 0.000225664 | **-0.030780254** | 0.009711951 | 0.005960144 | 0.003164369 | 0.005532174 |
| **Barbara** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000033146 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.000175062 | 0.025163365 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000072749 | 0.000000000 | 0.000000000 | 0.000006438 |
| 4 | **-0.001193714** | 0.004823971 | 0.096978636 | 0.000004682 | 0.000001170 | 0.000000000 | 0.002387427 | 0.000974803 | 0.000000000 | 0.000000000 | 0.000086770 |
| 5 | 0.000000000 | 0.022376473 | 0.255756512 | 0.000232544 | 0.001364677 | 0.000000000 | 0.002163712 | 0.006133544 | 0.000023865 | 0.000007159 | 0.000755184 |
| **Boat** | 2 | 0.000000000 | 0.000000000 | 0.000015714 | 0.000000000 | 0.000929755 | 0.000000000 | **-0.000004107** | 0.000151976 | 0.000000000 | 0.000062856 | 0.000000000 |
| 3 | 0.000000000 | 0.000512753 | 0.033547404 | 0.000000000 | 0.002839237 | 0.000000000 | 0.000635737 | 0.000472761 | 0.000000000 | 0.000000150 | 0.000057251 |
| 4 | **-0.003801340** | 0.014272311 | 0.138337936 | 0.000000000 | 0.005936350 | 0.000063613 | **-0.018292899** | 0.001241675 | 0.001980931 | 0.001065734 | 0.001830170 |
| 5 | 0.009895807 | 0.048293594 | 0.226405848 | 0.000043971 | 0.040133571 | 0.000009771 | 0.016542332 | 0.002544796 | 0.000444397 | 0.003250284 | 0.001683530 |
| **Goldhill** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000001318 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | 0.000000000 | 0.000244107 | 0.032199325 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000180135 | 0.000000000 | 0.000000000 | 0.000021486 |
| 4 | 0.000000000 | 0.003915947 | 0.096876762 | 0.000139983 | 0.000000000 | 0.000000000 | 0.000000000 | 0.001004980 | 0.000000000 | 0.000000000 | 0.000202893 |
| 5 | 0.000000000 | 0.020912351 | 0.163991512 | 0.000400905 | 0.000000000 | 0.000009691 | 0.000000000 | 0.004737299 | 0.000019382 | 0.000000000 | 0.001529729 |
| **Lake** | 2 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000000000 | 0.000030764 | 0.000000000 | 0.000000000 | 0.000000000 |
| 3 | **-0.005742663** | 0.000267197 | 0.031372689 | 0.000000000 | 0.000000000 | 0.000000000 | **-0.002871332** | 0.000023360 | 0.000000000 | 0.000000000 | 0.000000000 |
| 4 | 0.000000000 | 0.007001283 | 0.187096669 | 0.000045378 | 0.000000000 | 0.000000000 | 0.001161185 | 0.001857862 | 0.001161185 | 0.000000000 | 0.000042790 |
| 5 | 0.000003967 | 0.032190560 | 0.226535447 | 0.000552488 | 0.000035706 | 0.000003967 | **-0.006045551** | 0.001148200 | 0.000096015 | **-0.000043641** | 0.000815379 |

Note: the values in bold face with negative sign represent the results haven’t been improved.

**Table 23.** Difference of the variance from the original algorithms and the improved algorithms based on KE method

| **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 4.13e-009 | 0.00e+000 | 0.00e+000 | 3.55e-010 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 4.09e-007 | 5.28e-003 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 2.47e-006 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 4 | 0.00e+000 | 1.91e-005 | 8.12e-003 | 1.04e-007 | 6.83e-004 | 0.00e+000 | **-8.35e-004** | 4.09e-005 | 0.00e+000 | 0.00e+000 | 2.55e-008 |
| 5 | **-1.02e-002** | 7.46e-005 | **-5.27e-002** | 2.10e-006 | **-9.93e-003** | **-4.60e-003** | 5.02e-003 | 2.63e-004 | 8.36e-006 | **-1.28e-006** | 1.86e-005 |
| **Cameraman** | 2 | 0.00e+000 | 0.00e+000 | 5.21e-009 | 0.00e+000 | 1.51e-007 | 0.00e+000 | 1.58e-007 | 1.89e-009 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | **-2.18e-004** | 4.29e-006 | 6.55e-004 | 0.00e+000 | 5.06e-007 | 0.00e+000 | 0.00e+000 | 2.76e-006 | 0.00e+000 | **-3.45e-004** | 0.00e+000 |
| 4 | 0.00e+000 | 5.60e-005 | 2.60e-003 | 0.00e+000 | 0.00e+000 | 0.00e+000 | **-1.81e-007** | 6.95e-005 | 0.00e+000 | 3.27e-008 | 9.52e-009 |
| 5 | 2.28e-004 | 3.16e-004 | 3.12e-003 | 3.91e-009 | **-1.82e-004** | 6.10e-008 | **-2.20e-005** | 5.09e-004 | 2.58e-004 | **-5.78e-004** | 9.03e-004 |
| **Aerial** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 2.37e-005 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 3.27e-007 | 3.60e-004 | 0.00e+000 | **-1.62e-004** | 0.00e+000 | 3.82e-005 | 4.64e-006 | 0.00e+000 | 0.00e+000 | 8.96e-010 |
| 4 | **-5.43e-005** | 9.32e-005 | 2.16e-003 | 1.88e-008 | **-5.23e-005** | 0.00e+000 | 1.08e-004 | 4.66e-005 | 0.00e+000 | 9.21e-010 | 7.44e-009 |
| 5 | **-2.61e-003** | 2.19e-004 | 4.73e-003 | 4.57e-007 | **-6.75e-004** | 1.64e-006 | **-1.10e-003** | 1.41e-003 | 1.12e-003 | 1.25e-004 | 7.92e-004 |
| **Barbara** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 4.88e-008 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 6.12e-008 | 6.45e-004 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 1.35e-007 | 0.00e+000 | 0.00e+000 | 1.86e-009 |
| 4 | **-6.83e-005** | 5.92e-006 | 2.44e-003 | 2.57e-010 | **-2.78e-009** | 0.00e+000 | 1.22e-004 | 7.63e-006 | 0.00e+000 | 0.00e+000 | 6.74e-008 |
| 5 | 0.00e+000 | 7.95e-005 | 5.40e-003 | 5.94e-008 | 2.19e-005 | 0.00e+000 | 1.98e-005 | 2.40e-004 | 2.32e-009 | 8.19e-010 | 4.64e-006 |
| **Boat** | 2 | 0.00e+000 | 0.00e+000 | 1.23e-008 | 0.00e+000 | 4.59e-006 | 0.00e+000 | 1.45e-006 | 2.48e-007 | 0.00e+000 | 2.22e-008 | 0.00e+000 |
| 3 | 0.00e+000 | 8.92e-007 | 6.27e-004 | 0.00e+000 | **-1.10e-006** | 0.00e+000 | **-3.25e-006** | 6.77e-006 | 0.00e+000 | 1.13e-012 | 1.56e-007 |
| 4 | **-7.95e-004** | 3.21e-005 | 2.25e-003 | 0.00e+000 | **-2.25e-005** | 3.49e-008 | **-5.04e-004** | 2.74e-005 | 5.27e-005 | 5.29e-005 | 1.74e-004 |
| 5 | 4.22e-004 | 3.44e-004 | 4.77e-003 | 8.99e-009 | 2.33e-005 | 2.34e-009 | 7.15e-004 | 8.08e-005 | 5.26e-005 | 4.26e-005 | **-4.02e-007** |
| **Goldhill** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 8.68e-011 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | 0.00e+000 | 8.13e-008 | 5.72e-004 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 2.99e-007 | 0.00e+000 | 0.00e+000 | 8.42e-009 |
| 4 | 0.00e+000 | 4.74e-006 | 1.66e-003 | 2.30e-008 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 1.04e-005 | 0.00e+000 | 0.00e+000 | 4.15e-007 |
| 5 | 0.00e+000 | 5.73e-005 | 4.92e-003 | 1.34e-007 | 0.00e+000 | 4.70e-009 | 0.00e+000 | 1.60e-004 | 8.93e-009 | 0.00e+000 | 3.52e-005 |
| **Lake** | 2 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 0.00e+000 | 1.24e-008 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 3 | **-7.40e-004** | 4.54e-008 | 1.42e-003 | 0.00e+000 | 0.00e+000 | 0.00e+000 | **-3.95e-004** | 1.34e-008 | 0.00e+000 | 0.00e+000 | 0.00e+000 |
| 4 | 0.00e+000 | 9.74e-006 | 3.95e-003 | 1.29e-008 | 0.00e+000 | 0.00e+000 | 4.82e-005 | 3.17e-005 | 6.74e-005 | 0.00e+000 | 4.62e-008 |
| 5 | 7.87e-010 | 2.39e-004 | 3.25e-003 | 1.45e-007 | 3.32e-009 | 7.87e-010 | **-1.82e-003** | 1.75e-005 | 1.52e-008 | **-4.06e-009** | 2.65e-006 |

Note: the values in bold face with negative sign represent the results haven’t been improved.**Table 24.** Difference of the mean convergence time from the original algorithms and the improved algorithms based on KE method

| **Images** | ***d*** | **PSO** | **SFLA** | **ABCA** | **CSO** | **FA** | **CS** | **BA** | **GWO** | **ALO** | **WOA** | **MFO** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Zebra** | 2 | 0.0830 | 1.7228 | 1.0278 | 0.5079 | 0.0803 | 0.3408 | 0.0166 | 1.6029 | 0.5834 | 0.1565 | 0.2846 |
| 3 | 0.2113 | 14.6781 | 1.7708 | 1.7667 | 0.6631 | 0.9635 | 0.1027 | 3.0205 | 1.4308 | 0.4972 | 0.6270 |
| 4 | 0.3189 | 13.0238 | **-1.2441** | 7.0480 | 1.2634 | 1.4997 | 0.1368 | 2.7054 | 1.9004 | 0.7611 | 0.7129 |
| 5 | 0.5051 | 7.7414 | **-2.5909** | 3.4239 | 0.7959 | 0.5462 | 0.3194 | 0.8437 | 2.3853 | 0.7972 | 0.6895 |
| **Cameraman** | 2 | 0.1776 | 3.1648 | 2.0165 | 0.7254 | 0.5724 | 0.6511 | 0.0325 | 1.8477 | 0.6948 | 0.2238 | 0.3432 |
| 3 | 0.1568 | 12.3014 | **-0.2336** | 0.9016 | 0.2302 | 0.6194 | 0.0130 | 1.9543 | 0.5986 | 0.3867 | 0.2571 |
| 4 | 0.2780 | 11.6963 | 0.6449 | 3.7696 | 0.2219 | 1.1616 | 0.0332 | 2.0175 | 1.5987 | 0.4426 | 0.6364 |
| 5 | 0.5216 | 4.6470 | **-3.1290** | 3.1129 | 0.3029 | 0.1563 | 0.3018 | **-1.4522** | 1.1409 | 0.1564 | 0.0167 |
| **Aerial** | 2 | 0.0461 | 2.0084 | 0.6688 | 0.3574 | 0.1012 | 0.2265 | 0.0418 | 1.1302 | 0.2879 | 0.3420 | 0.1956 |
| 3 | 0.1559 | 14.5128 | 0.9061 | 1.4294 | 0.1439 | 0.7091 | **-0.0383** | 2.9217 | 0.9070 | 0.3497 | 0.5509 |
| 4 | 0.1609 | 9.5226 | **-1.8919** | 5.5483 | 0.3655 | 0.9318 | 0.0895 | 2.4362 | 1.1179 | 0.9952 | 0.4835 |
| 5 | 0.7454 | 4.0321 | **-3.5487** | 4.0232 | 0.8371 | 0.4764 | 0.2585 | **-1.0552** | 1.3128 | 0.9152 | **-0.1630** |
| **Barbara** | 2 | 0.1135 | 1.9214 | 0.9941 | 0.5962 | 0.1419 | 0.4413 | 0.0306 | 1.8259 | 0.7244 | 0.1959 | 0.3369 |
| 3 | 0.1660 | 14.1860 | 0.6548 | 1.6617 | 0.2296 | 0.7191 | 0.0443 | 2.5343 | 0.9972 | 0.2531 | 0.5492 |
| 4 | 0.2109 | 11.1702 | **-0.7682** | 5.0704 | 0.2593 | 1.0612 | 0.0385 | 2.1550 | 1.3662 | 0.4479 | 0.6319 |
| 5 | 0.7541 | 10.9349 | 0.4382 | 7.3914 | 1.0842 | 1.8146 | 0.5369 | 1.9927 | 2.1259 | 1.0289 | 0.0749 |
| **Boat** | 2 | 0.0710 | 1.3536 | 0.4455 | 0.2861 | 0.1327 | 0.2211 | 0.0845 | 1.1321 | 0.2260 | 0.0996 | 0.0857 |
| 3 | 0.1185 | 12.4702 | 0.0802 | 1.1288 | 0.2419 | 0.5767 | 0.0654 | 1.8166 | 0.7621 | 0.6416 | 0.4767 |
| 4 | 0.2340 | 8.6075 | **-3.0664** | 3.0559 | 0.5404 | 1.0507 | 0.2665 | 0.2333 | 0.9522 | 0.4735 | 0.2251 |
| 5 | 0.8310 | 7.2573 | **-0.8338** | 6.8395 | 0.6885 | 1.7741 | 0.4327 | 1.1538 | 1.6902 | 0.6523 | 0.0573 |
| **Goldhill** | 2 | 0.1086 | 1.9873 | 1.0865 | 0.6679 | 0.1029 | 0.4590 | 0.0310 | 1.8279 | 0.7372 | 0.2405 | 0.3149 |
| 3 | 0.1850 | 14.3154 | 1.6572 | 1.8045 | 0.3914 | 0.8174 | 0.0722 | 2.4801 | 0.9199 | 0.5189 | 0.5464 |
| 4 | 0.1493 | 8.9331 | **-1.8158** | 5.2982 | 0.3463 | 0.6713 | 0.0965 | **-0.0489** | 0.5691 | 0.3344 | **-0.0423** |
| 5 | 0.5180 | 9.8666 | **-2.9592** | 5.3410 | 0.5674 | 0.9997 | 0.3645 | 0.3683 | 0.9920 | 0.7594 | 0.2300 |
| **Lake** | 2 | 0.1050 | 2.3678 | 1.1093 | 0.6488 | 0.1047 | 0.4554 | 0.0309 | 1.5257 | 0.6911 | 0.1398 | 0.3269 |
| 3 | 0.1944 | 14.4337 | 1.3958 | 1.7876 | 0.3137 | 0.9174 | 0.0664 | 3.3567 | 1.4554 | 0.3773 | 0.6843 |
| 4 | 0.3252 | 11.9870 | 1.9518 | 5.8753 | 0.7861 | 1.5062 | 0.1954 | 3.3681 | 2.1015 | 0.8632 | 0.9373 |
| 5 | 0.5500 | 8.8911 | 1.4133 | 6.8761 | 0.1818 | 0.0493 | 0.8068 | **-1.4187** | 1.1027 | 0.6206 | 0.0937 |

Note: the values in bold face with negative sign represent the results haven’t been improved.

**Figures**

**Fig.1.** Classification of meta-heuristic algorithms



**Fig.2.** Framework of applying SI algorithms to optimize BCV and KE method

**Algorithm 1** (Cuckoo Search applied to multilevel thresholding).

**Parameters:** set *N*, **, , , *MaxT*, =;

**Begin**

//Initialize population and calculate fitness

**for** *i* = 1 to *N* **do**

Randomize nest position ***X****i* ;

Calculate corresponding fitness, ;

**end** //for

*iter* = 1;

// Main iterations of the CS algorithm

**while** *iter* <= *MaxT* **do**

Update ***Xg***, ***Xg*** = { ***X****i*: *min*() };

// Exploration phase: Levy flight

**for** *i* = 1 to *N* **do**

Get new nest through Levy flight by equation (6) and equation (7);

Calculate corresponding fitness, ;

**if**  **do**

=;

;

**end** //if

**end** //for

// Exploitation phase: abandoned fraction  of worse nests

**for** *i* = 1 to *N* **do**

Get new nestby equation (8) to equation (10);

Calculate corresponding fitness, ;

**if**  **do**

=;

;

**end** //if

**end** //for

*iter*++ ;

**end** //while

// Output ***Xg*** : as the approximate optimal thresholds to the optimal thresholdsof

**end** //Begin

**Fig.3.** Pseudo-code of the CS algorithm applied to BCV-based multilevel thresholding



**Fig.4.** Demonstration of exploratory move and pattern move



**Fig.5.** Demonstration of the execution of pattern search algorithm



**Fig.6.** Flowchart of improved SI algorithms applied to multilevel thresholding

**Algorithm 2** (Improved Cuckoo Search applied to multilevel thresholding).

**Parameters:** set *N*, **, , , *MaxT*, =, , , , ;

**Begin**

//Initialize population and calculate fitness

**for** *i* = 1 to *N* **do**

Randomize nest position ***X****i* ;

Calculate corresponding fitness, ;

**end** //for

*iter* = 1;

// Main iterations of the CS algorithm

**while** *iter* <= *MaxT* **do**

Update ***Xg***, ***Xg*** = { ***X****i*: *min*() };

// Plug in pattern search algorithm, where ***parameters*** = [, , , ]

***Xg*** := PatternSearch(***Xg***, ***parameters***);

// Exploration phase: Levy flight

**for** *i* = 1 to *N* **do**

Get new nest through Levy flight by equation (6) and equation (7);

Calculate corresponding fitness, ;

**if**  **do**

=;

;

**end** //if

**end** //for

// Exploitation phase: abandoned fraction  of worse nests

**for** *i* = 1 to *N* **do**

Get new nestby equation (8) to equation (10);

Calculate corresponding fitness, ;

**if**  **do**

=;

;

**end** //if

**end** //for

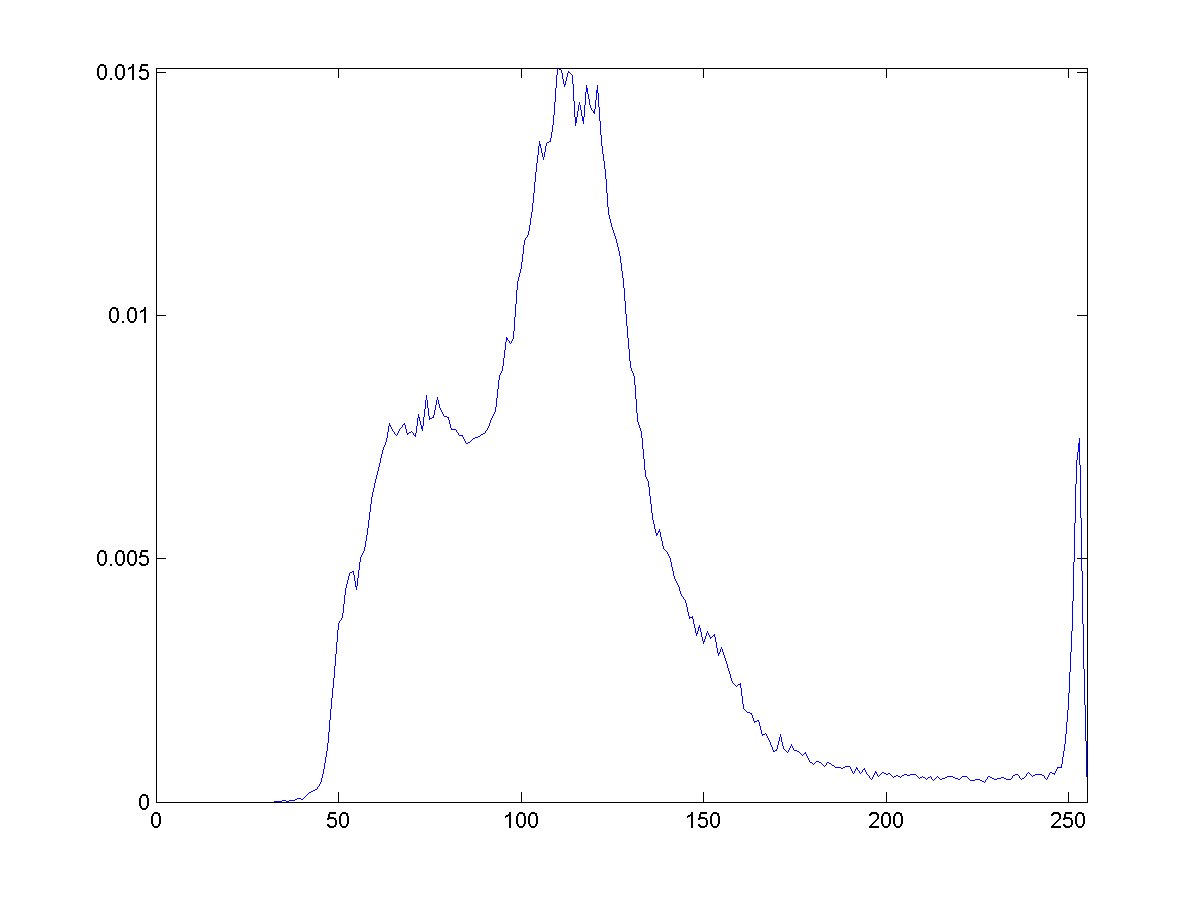
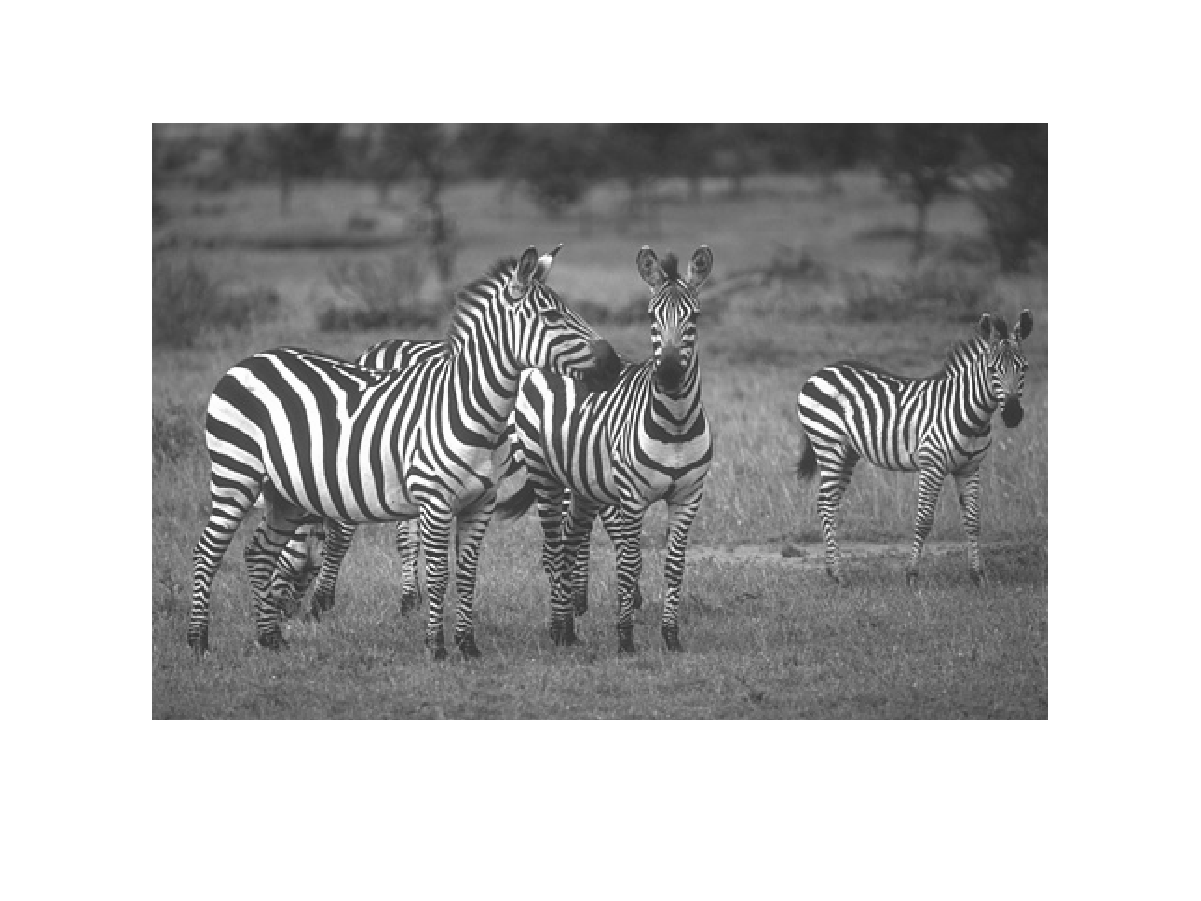
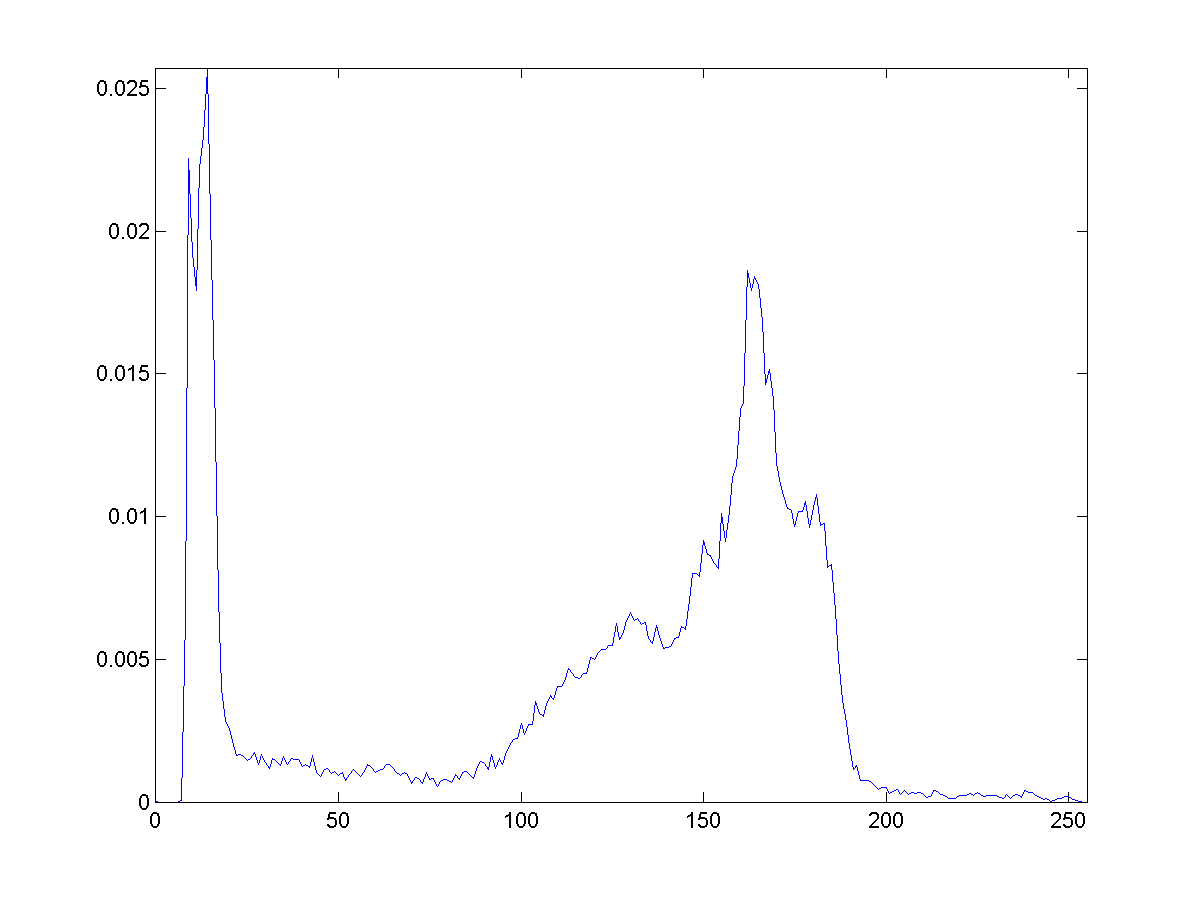
*iter*++ ;

**end** //while

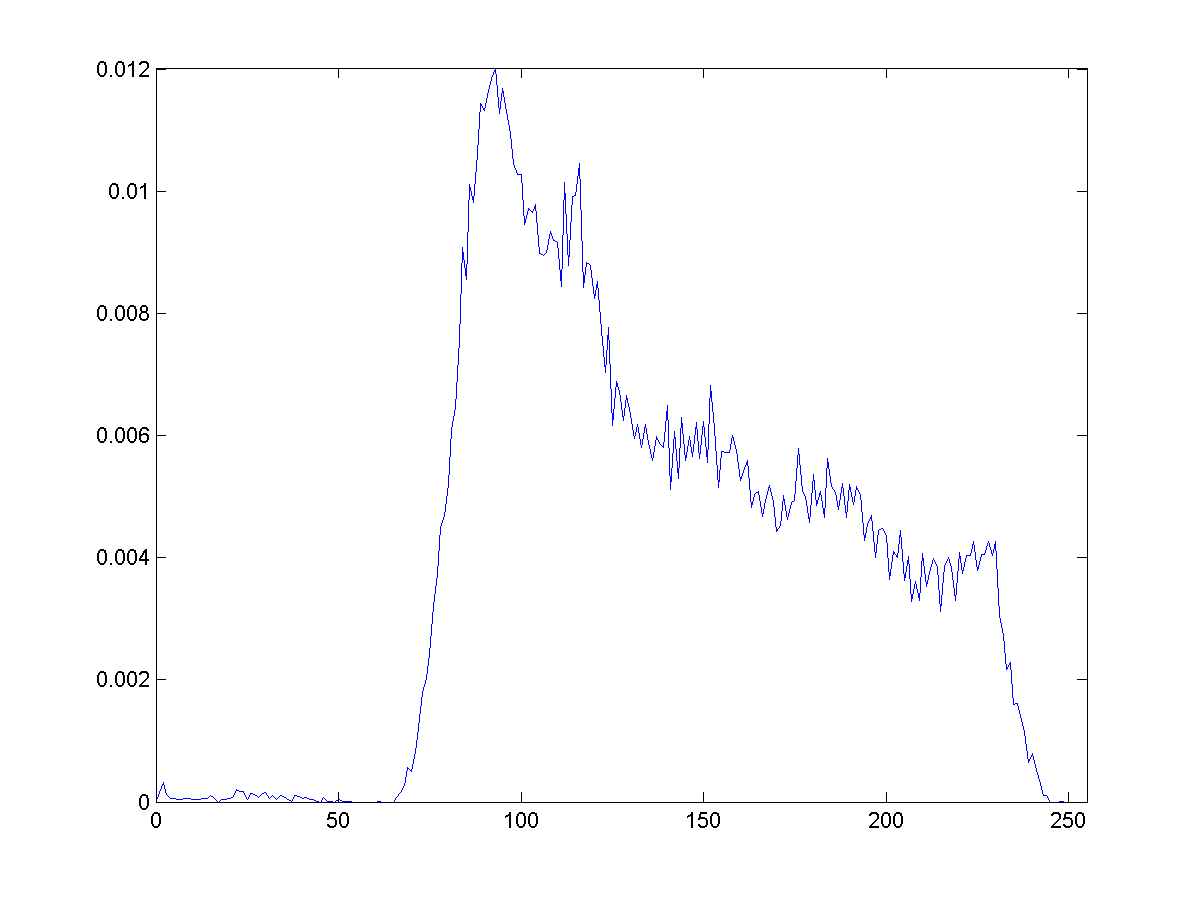
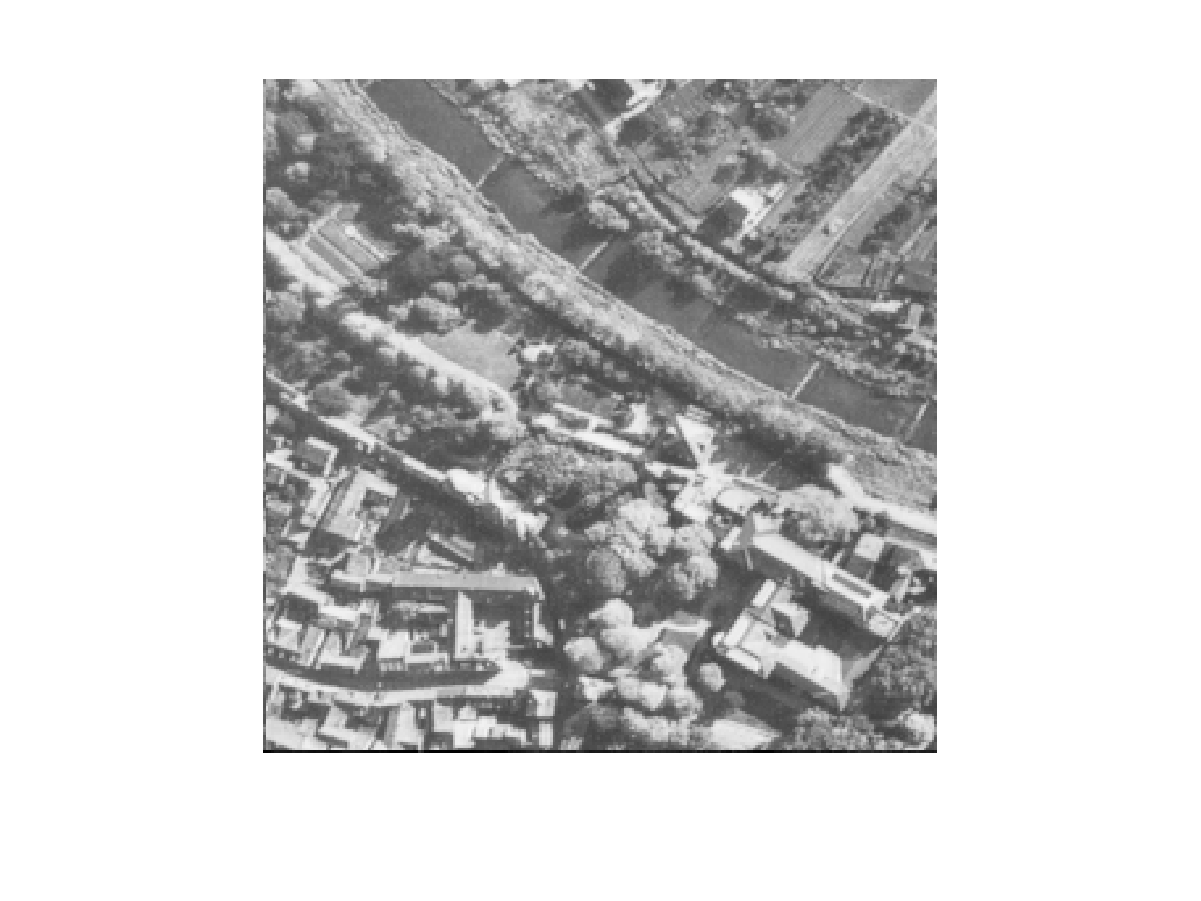
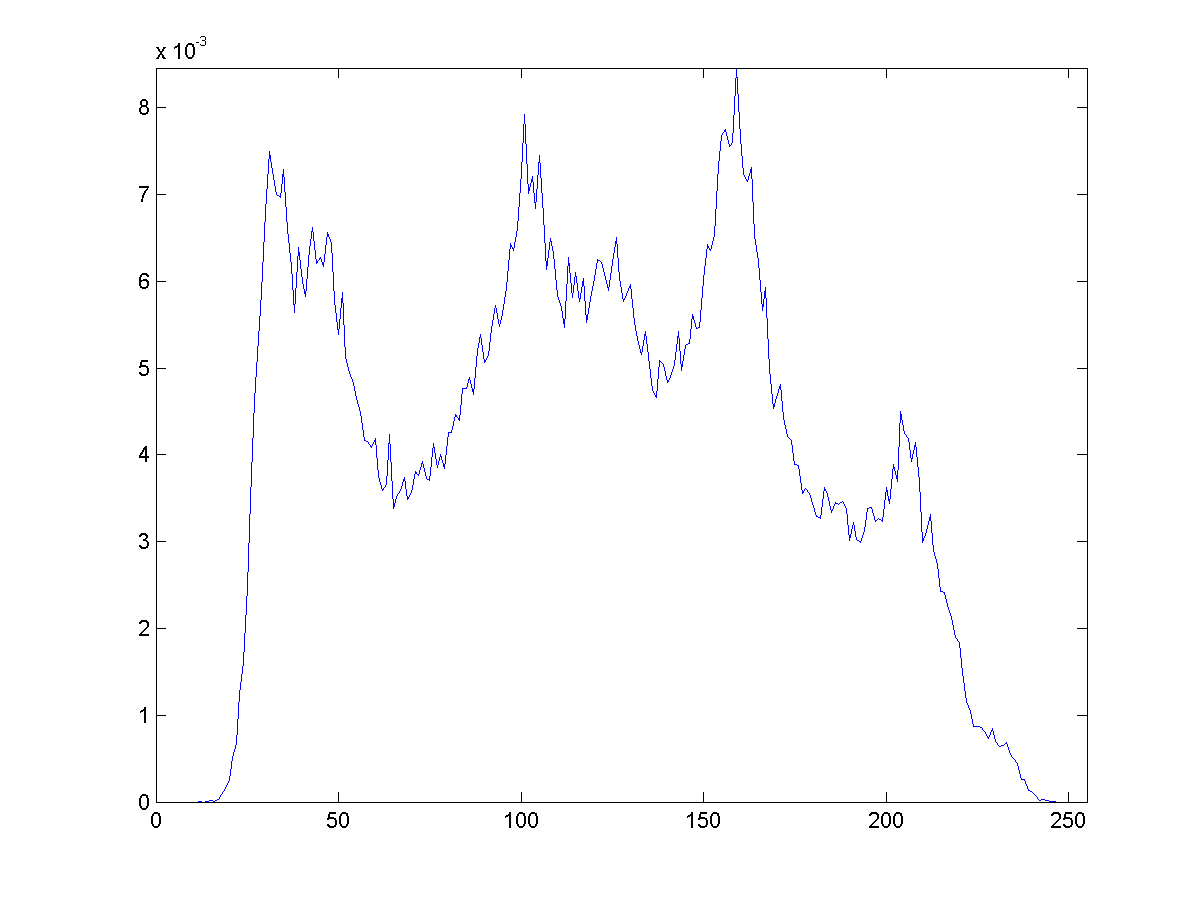
// Output ***Xg*** : as the approximate optimal thresholds to the optimal thresholdsof

**end** //Begin

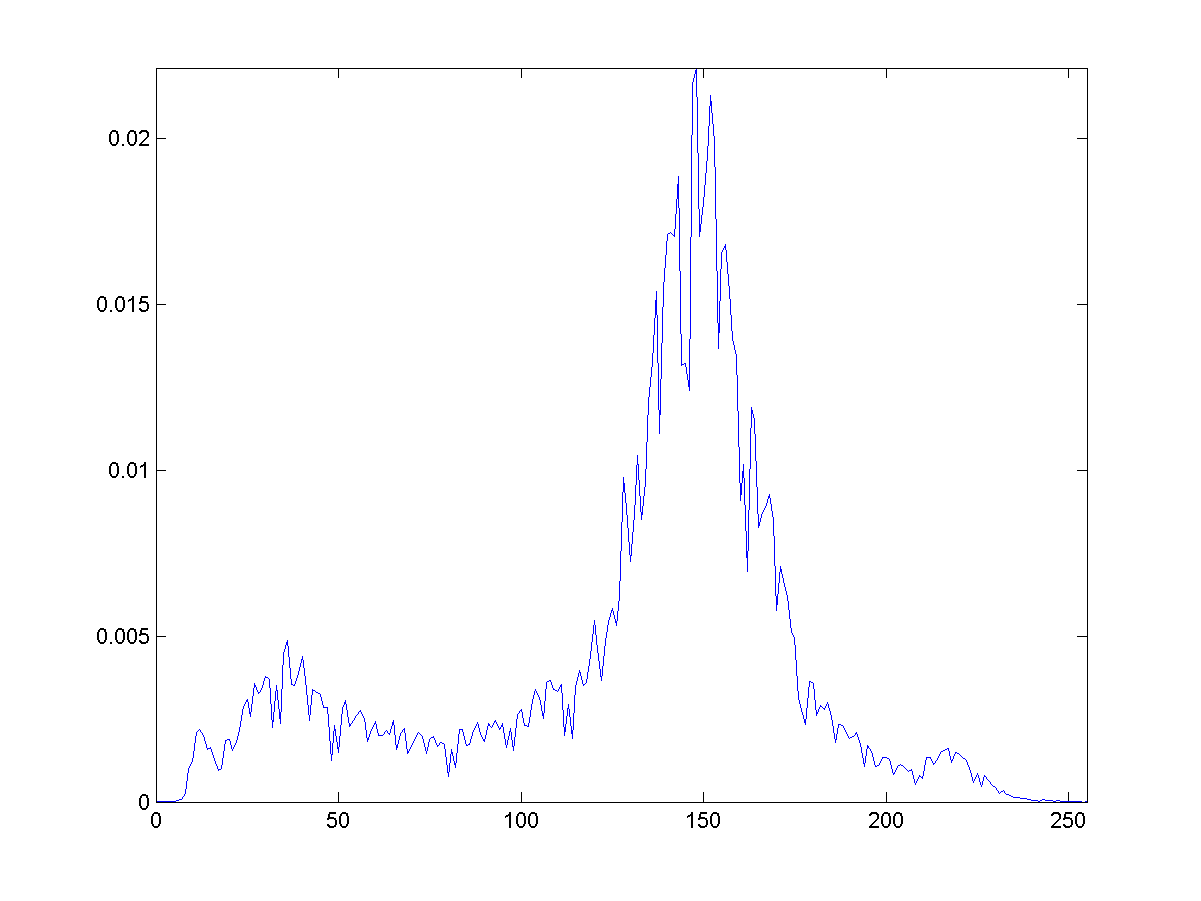
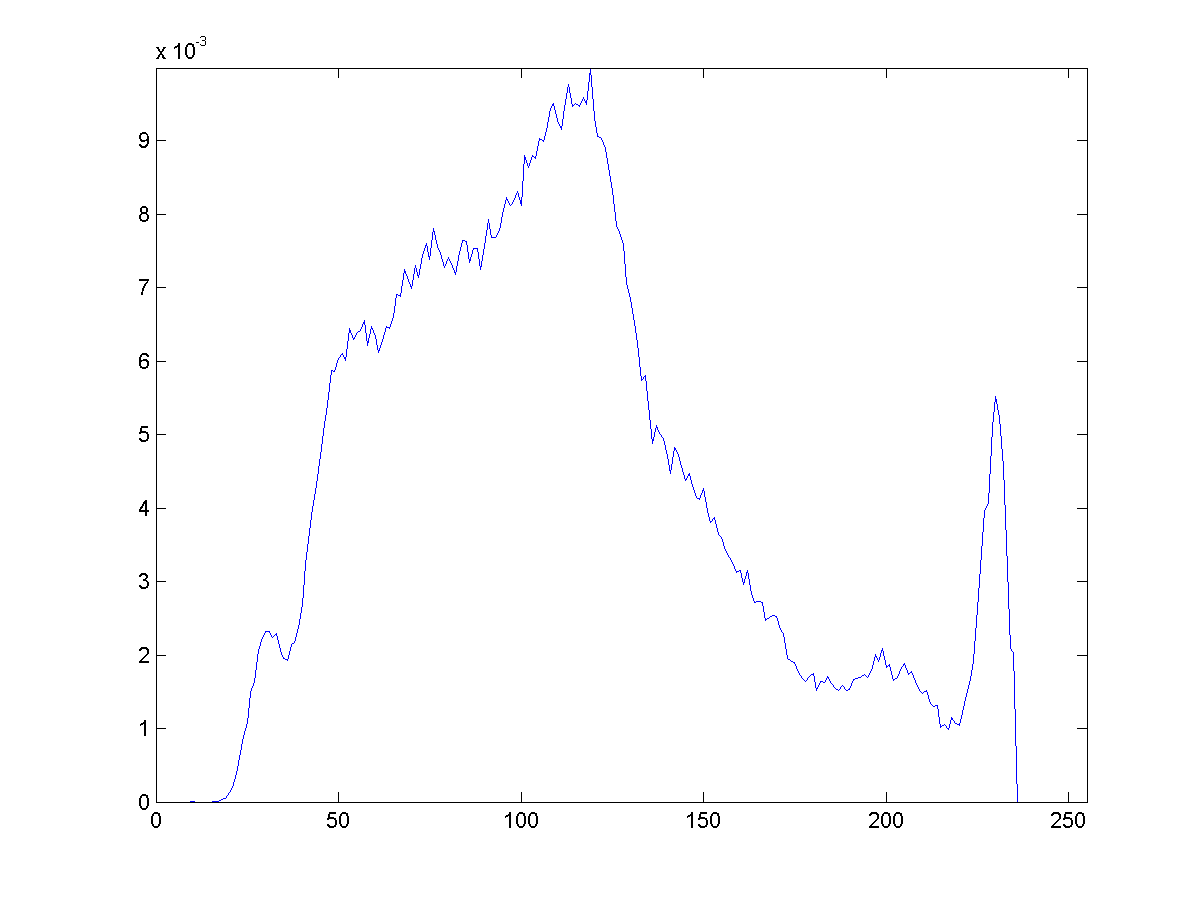
**Fig.7.** Pseudo-code of P-CS algorithm applying to BCV-based multilevel thresholding

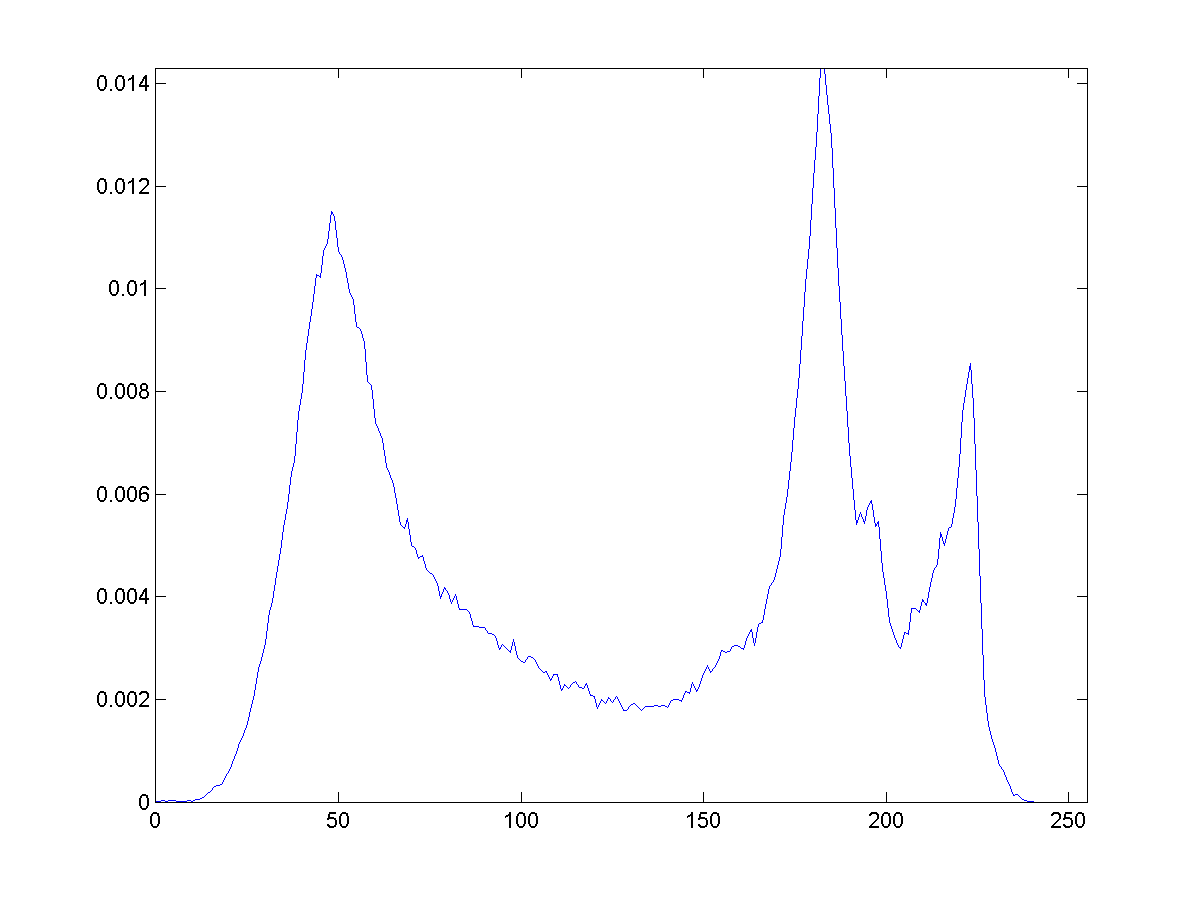
(I) Zebra (I’) Histogram of I (II) Cameraman (II’) Histogram of II

(III) Aerial (III’) Histogram of III (IV) Barbara (IV’) Histogram of IV

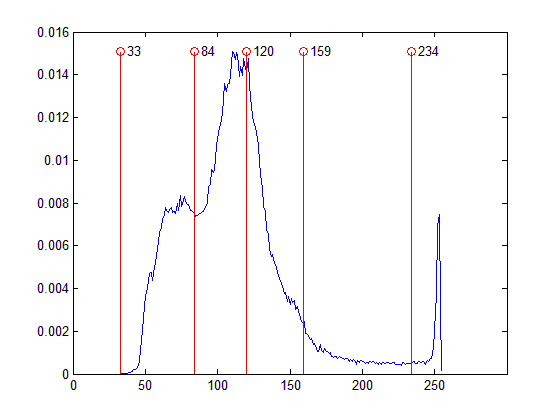
 

(V) Boat (V’) Histogram of V (VI) Goldhill (VI’) Histogram of VI

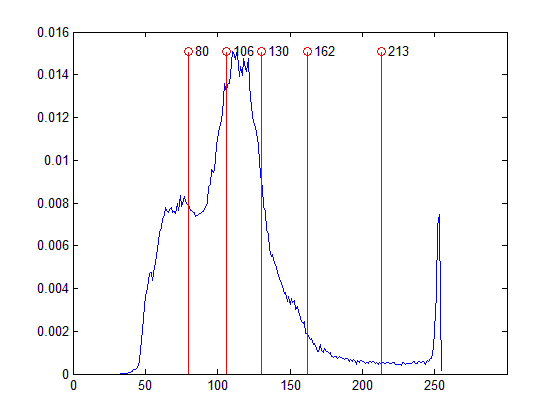
 

(VII) Lake (VII’) Histogram of VII

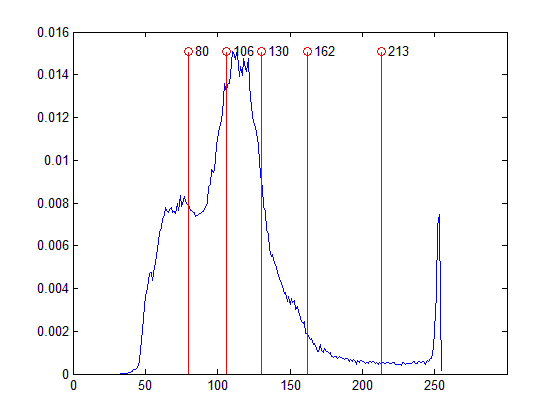
**Fig.8.** Testing images and normalized gray histograms



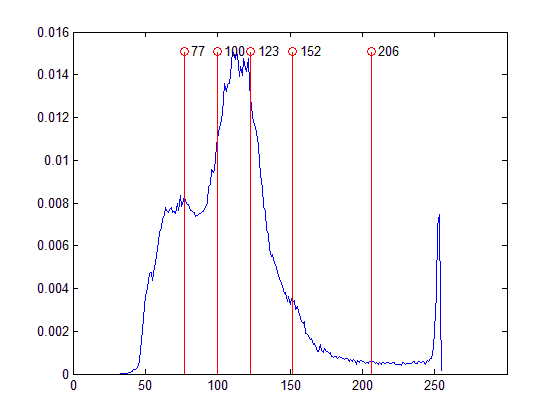
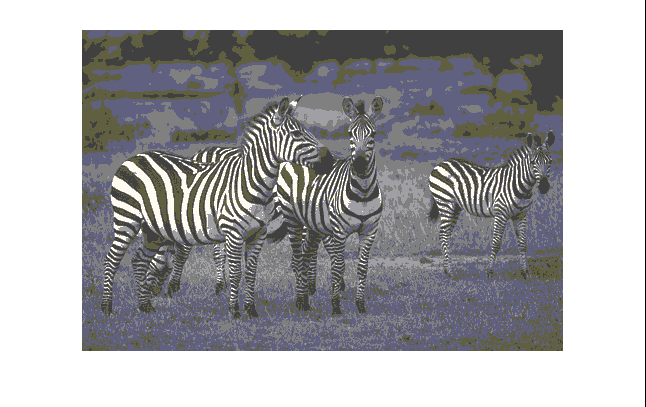
1. ABC



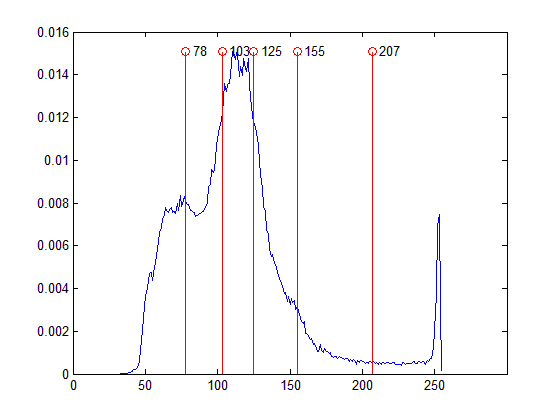
1. ALO



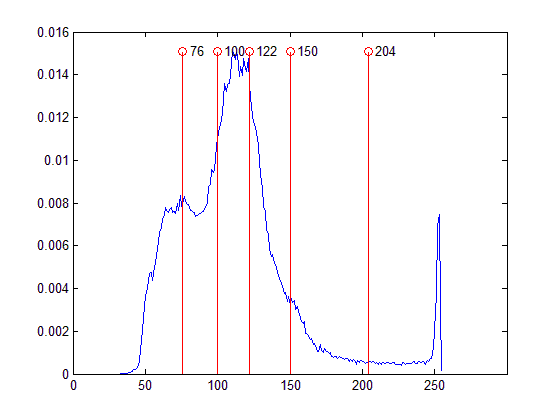
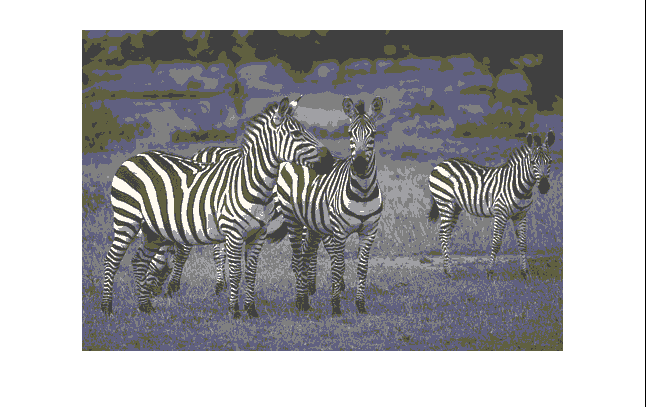
1. BA



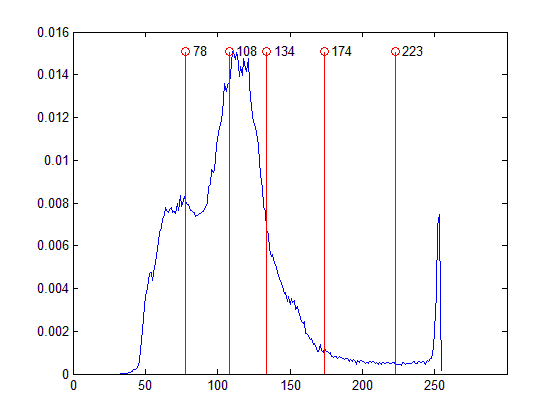
1. CS



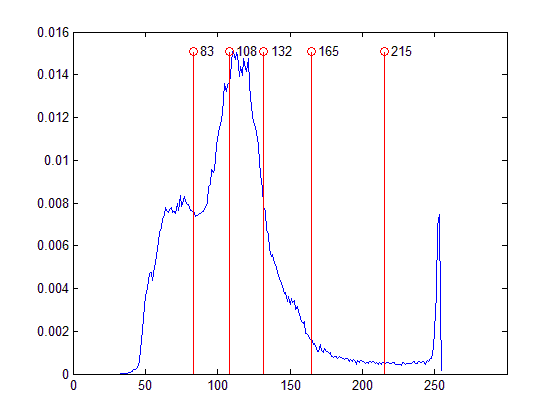
1. CSO



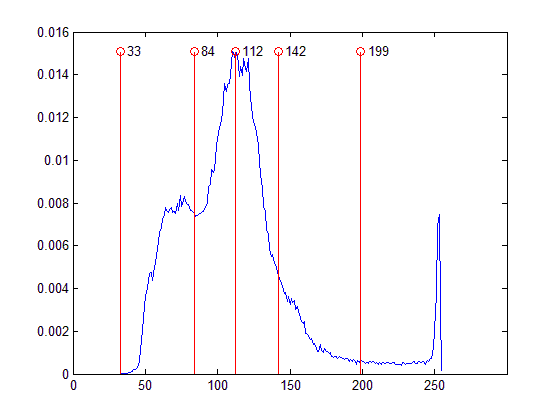
1. FA



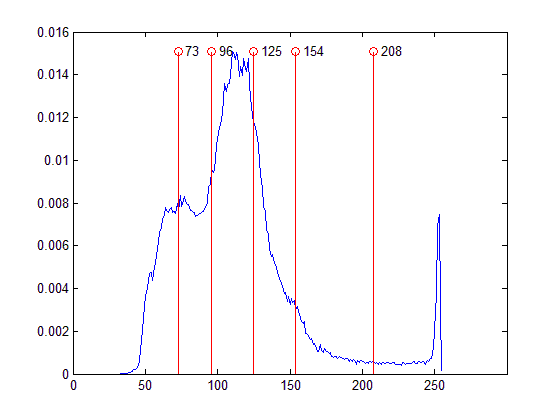
1. GWO



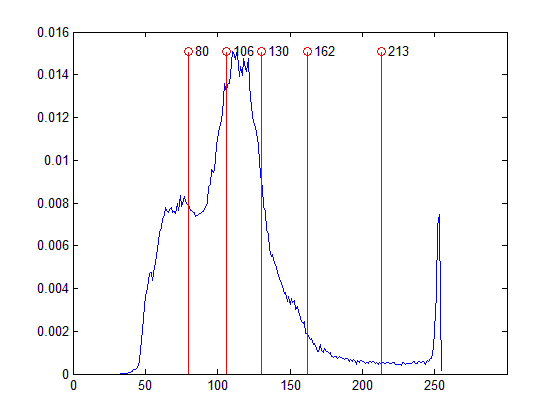
1. MFO



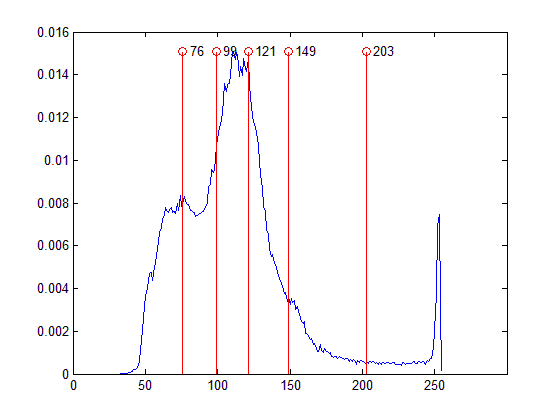
1. PSO



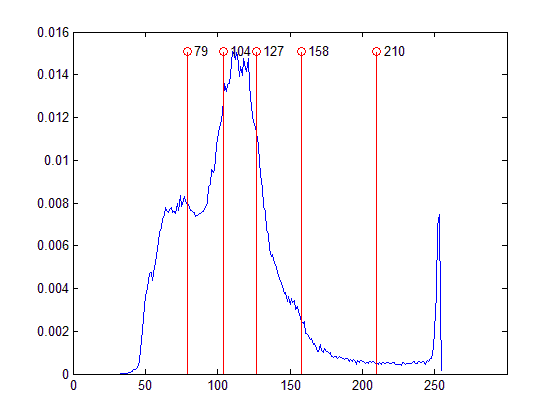
1. SFLA



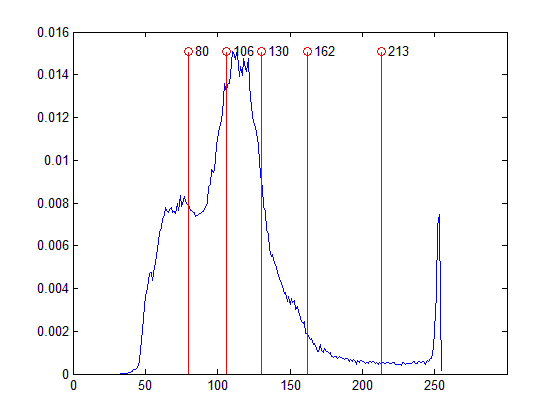
1. WOA



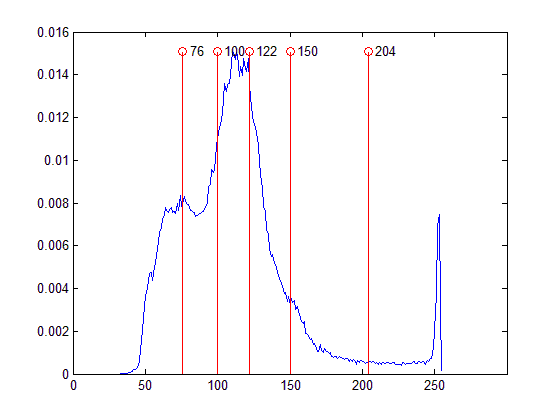
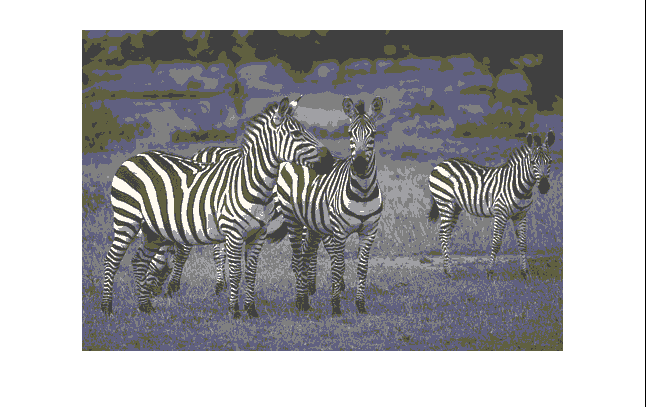
1. P-ABC



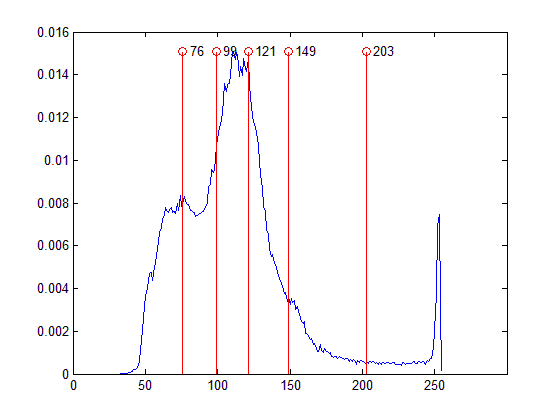
1. P-ALO



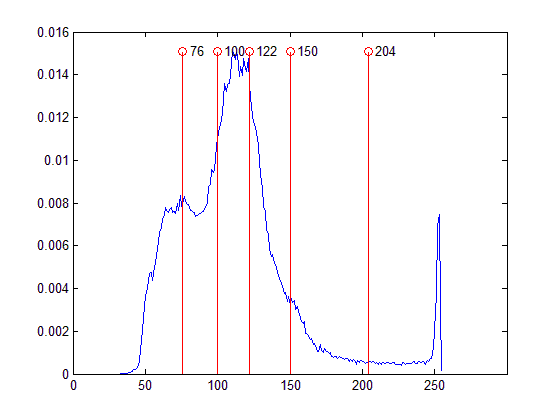
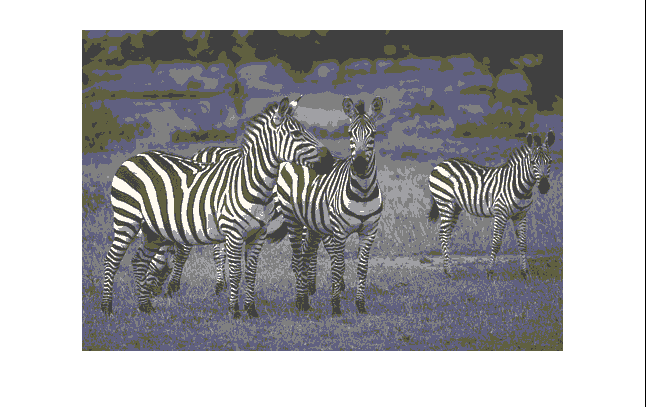
1. P-BA



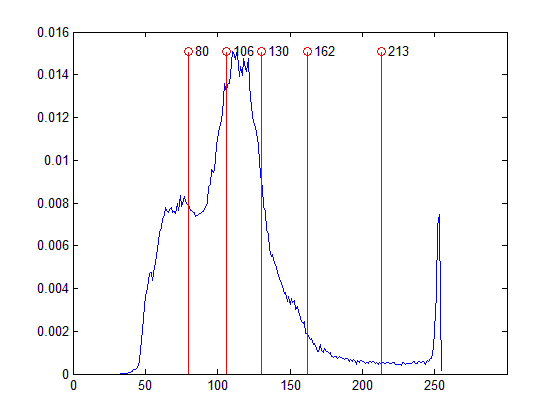
1. P-CS



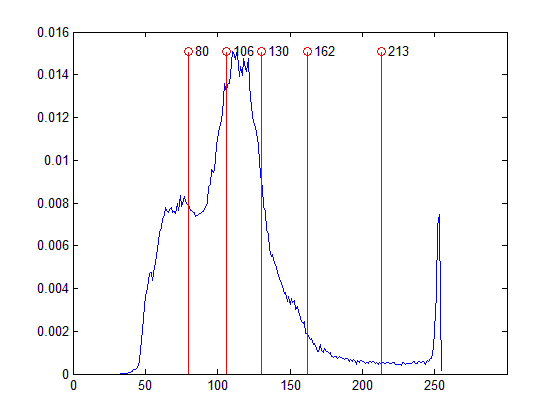
1. P-CSO



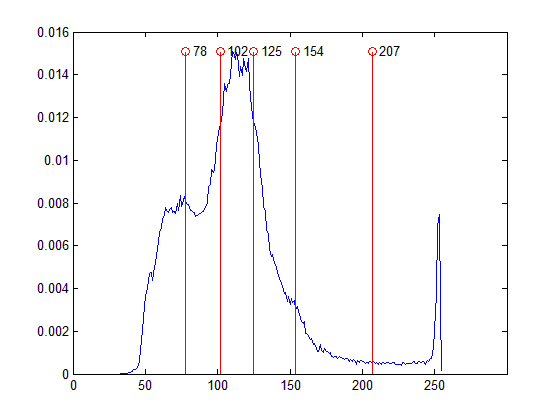
1. P-FA



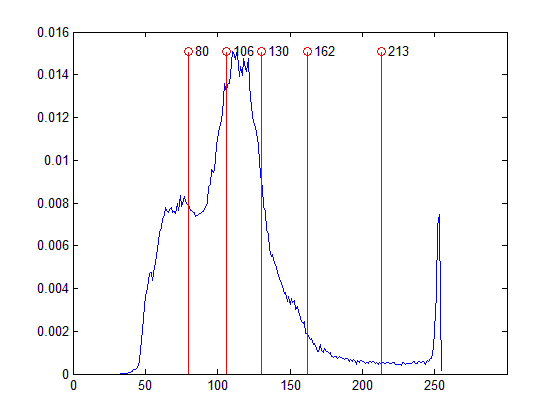
1. P-GWO



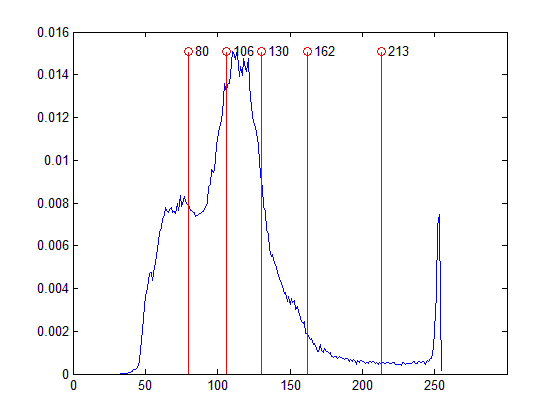
1. P-MFO



1. P-PSO

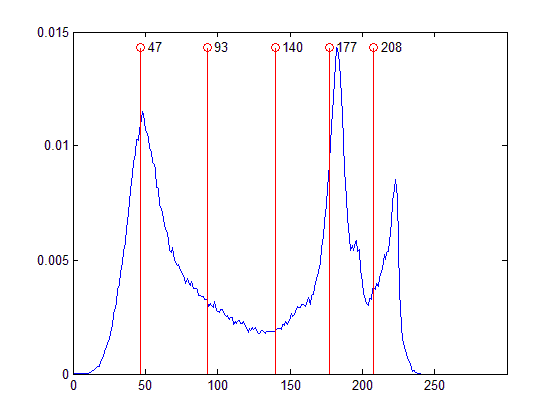


1. P-SFLA

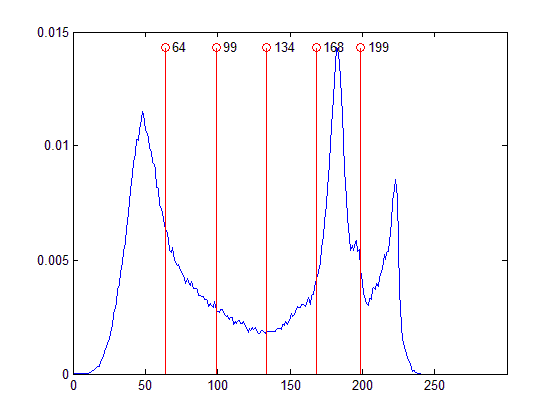


1. P-WOA

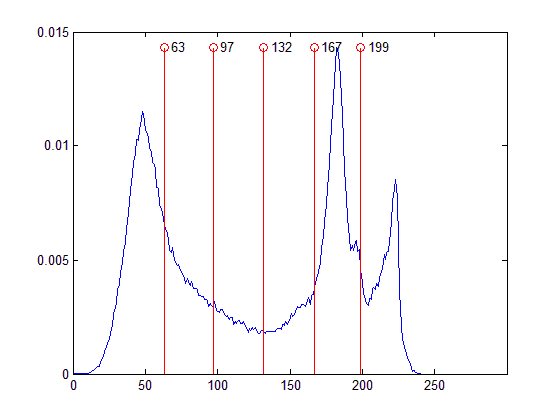
**Fig.9.** (a)-(v) are the segmented ‘Zebra’ images and the normalized histogram of the image labeled with segment thresholds when threshold number is 5 under different swarm intelligent algorithms based on BCV method.



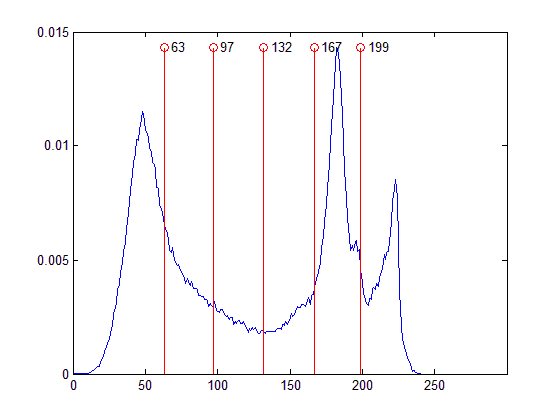
1. ABC



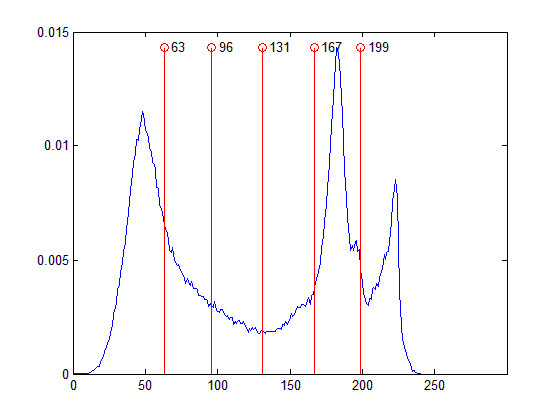
1. ALO



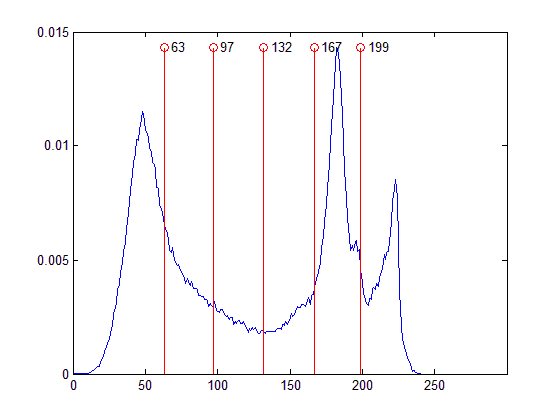
1. BA



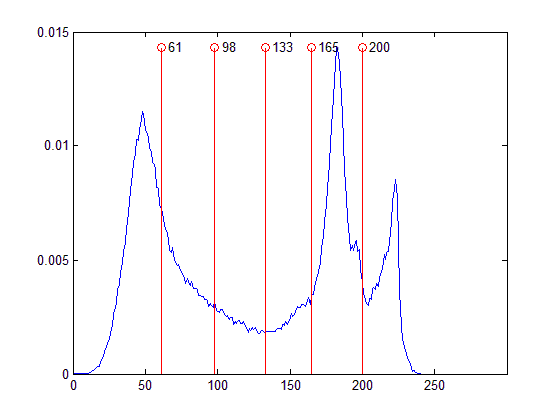
1. CS



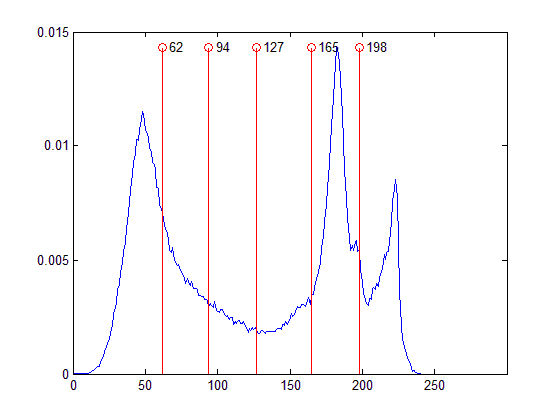
1. CSO



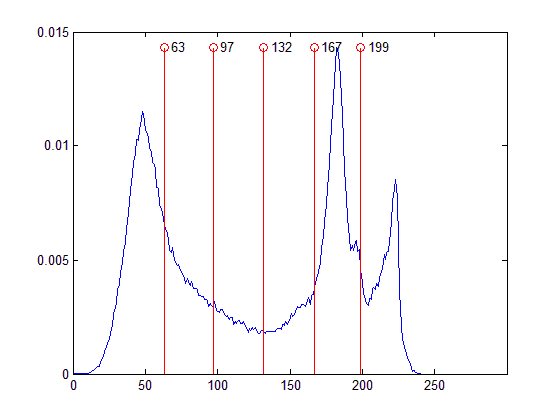
1. FA



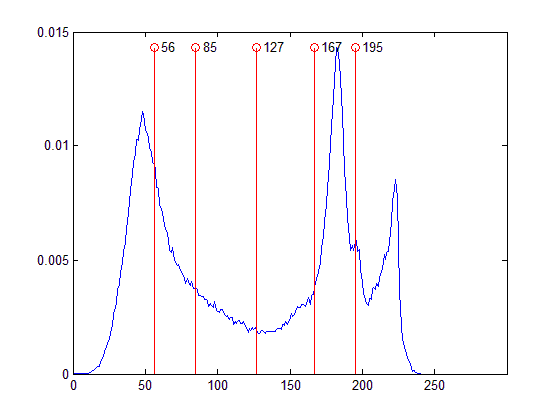
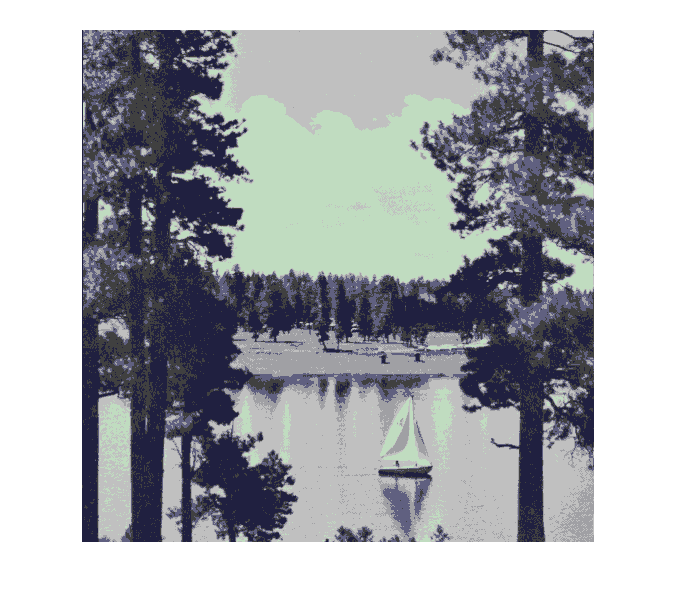
1. GWO



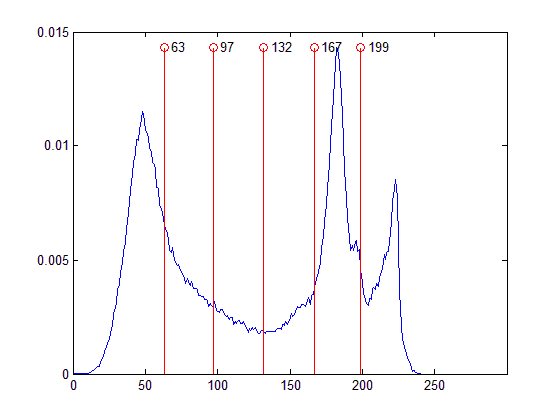
1. MFO



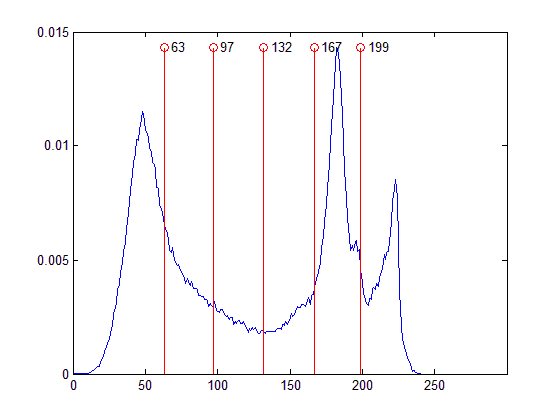
1. PSO



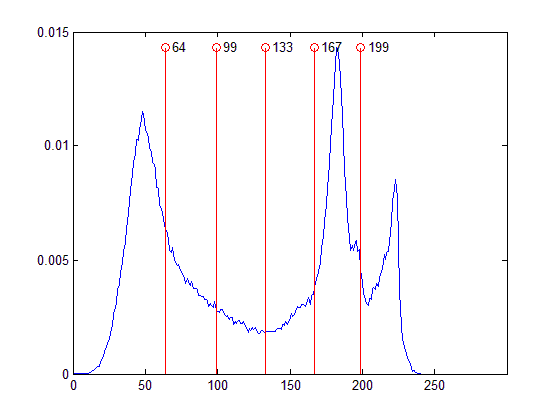
1. SFLA



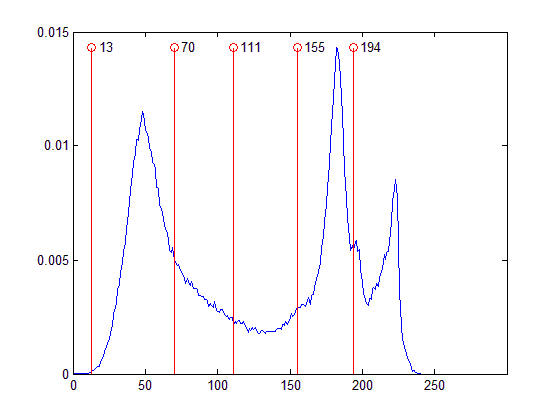
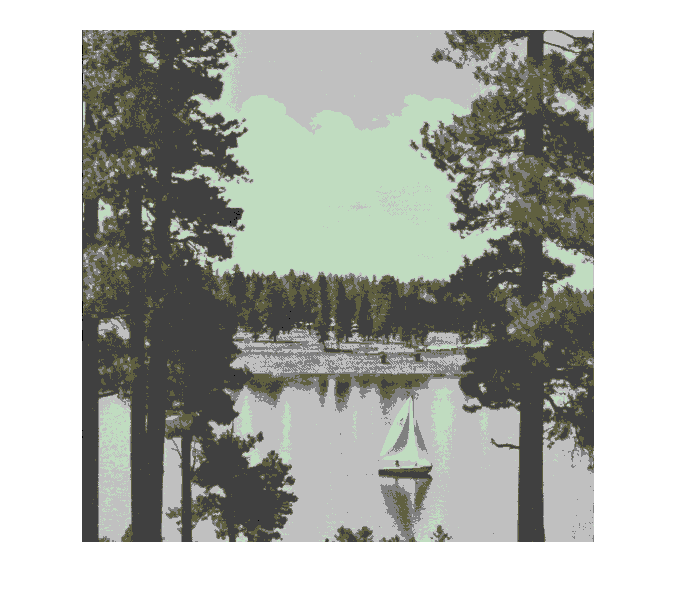
1. WOA



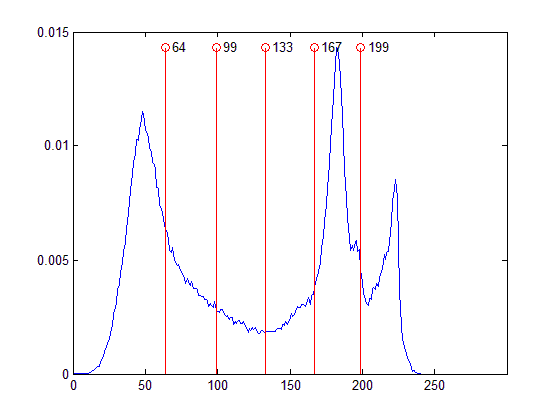
1. P-ABC



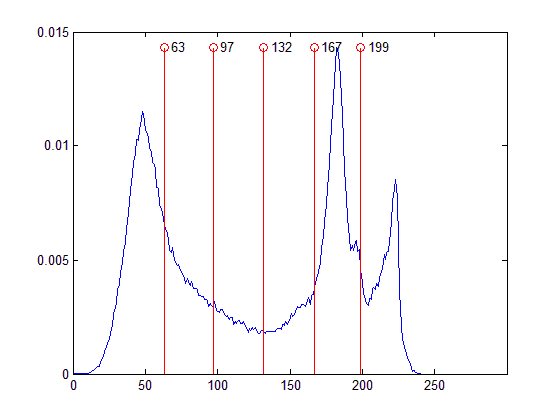
1. P-ALO



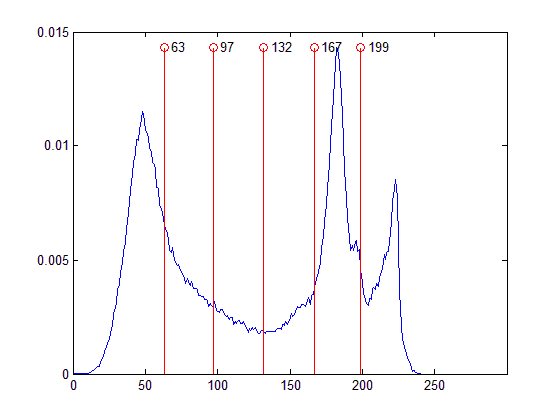
1. P-BA



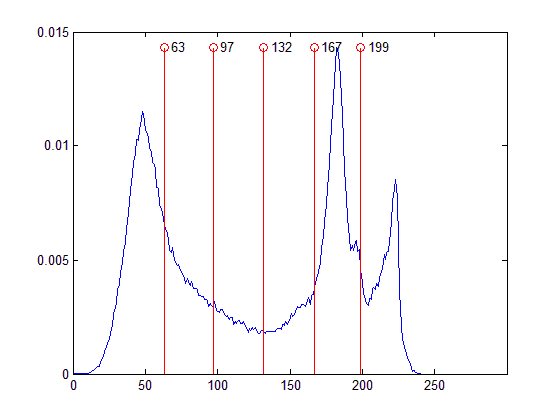
1. P-CS



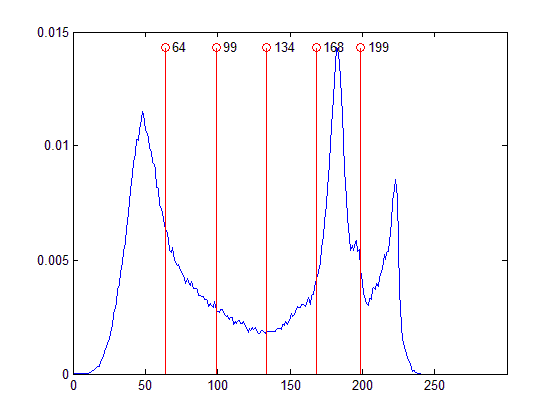
1. P-CSO



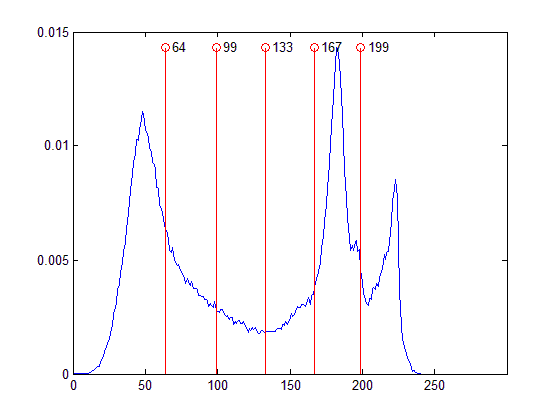
1. P-FA



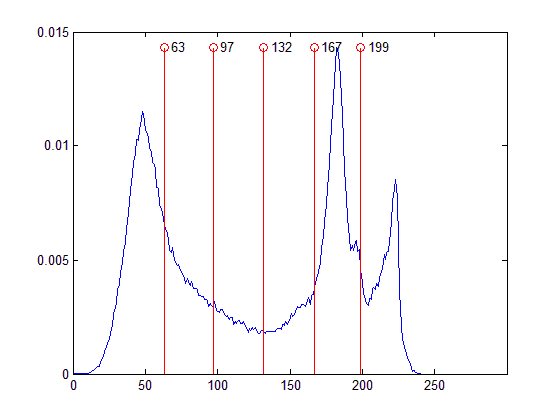
1. P-GWO



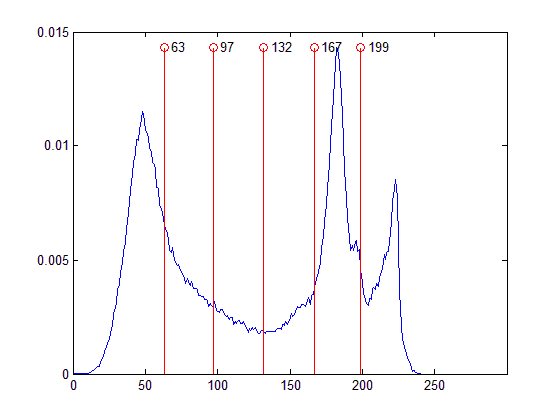
1. P-MFO



1. P-PSO



1. P-SFLA



1. P-WOA

**Fig.10.** (a)-(v) are the segmented ‘Boat’ images and the normalized histogram of the image labeled with segment thresholds when threshold number is 5 under different swarm intelligent algorithms based on KE method.

**Abbreviations**

| **Abbreviations** | **Meaning of the abbreviations** |
| --- | --- |
| ABCA | Artificial Bee Colony Algorithm |
| ABCA/best/1 | Artificial Bee Colony Algorithm with Differential Evolution strategy1 |
| ABCA/best/2 | Artificial Bee Colony Algorithm with Differential Evolution strategy2 |
| ABF | Amended Bacterial Foraging algorithm |
| ACO | Ant Colony Optimization |
| AFSA | Artificial Fish Swarm Algorithm |
| AgPSO | Aggressive Particle Swarm Optimization |
| ALO | Ant Lion Optimization |
| BA | Bat algorithm |
| BCV | Maximum between class variance |
| BD | Brownian Distribution |
| BF | Bacterial Foraging algorithm |
| BPSO | Bayesian Particle Swarm Optimization |
| CABC | Artificial Bee Colony with Candidate strategy |
| CDC | Counts of Dimension to Change (parameter in Cat Swarm Optimization) |
| CHBMA | Cooperative Honey Bee Mating-based Algorithm |
| CoDE | Composite Differential Evolution |
| CPU | Central Processing Unit |
| CS | Cuckoo Search |
| CSO | Cat Swarm Optimization |
| DE | Differential Evolution |
| DEO | Dolphin Echolocation Optimization |
| ELR-CS | Egg-Laying Radius based Cuckoo Search |
| FA | Firefly Algorithm |
| FP | Flower Pollination algorithm |
| FPSO | Fibonacci Particle Swarm Optimization |
| GA | Genetic Algorithm |
| GRPSO | Golden Ratio Particle Swarm Optimization |
| GWO | Grey Wolf Optimizer |
| HBMO | Honey Bee Mating Optimization |
| HCOCLPSO | Hybrid COoperative learning and Comprehensive Learning Particle Swarm Optimization |
| IABCA/best/1 | Improved Artificial Bee Colony Algorithm with DE strategy1 |
| IABCA/rand/1 | Improved Artificial Bee Colony Algorithm with DE strategy2 |
| IBA | Improved Bat Algorithm |
| IDPSO | Intermediate Disturbance Particle Swarm Optimization |
| IFA | Improved Firefly Algorithm |
| KE | Kapur’s entropy method |
| KH | Krill Herd algorithm |
| LF | Levy Flight |
| M.C.T | Mean Convergence Time (statistical index of the experimental results) |
| M.V. | Mean Value (statistical index of the experimental results) |
| MABCA | Modified Artificial Bee Colony Algorithm |
| MATLAB | The MATLAB 7.12 software platform |
| MBF | Modified Bacterial Foraging algorithm |
| MCET | Minimum Cross Entropy Thresholding |
| MEHBMOT | Honey Bee Mating Optimization based Maximum Entropy Thresholding |
| MET | Maximum entropy thresholding |
| MFO | Moth Flame Optimization |
| MPSO | Modified Particle Swarm Optimization |
| MR | Mixture Ratio (parameter in Cat Swarm Optimization) |
| MRI | Magnetic Resonance Imaging |
| MS | Monkey Search |
| MSFLA | Modified Shuffled Frog Leaping Algorithm |
| MTET | Maximum Tsallis entropy thresholding |
| NFL | No free lunch theorem |
| P-ABCA | Pattern search based Artificial Colony Algorithm (introduced in this paper) |
| P-ALO | Pattern search based Ant Lion Optimization (introduced in this paper) |
| P-BA | Pattern search based Bat Algorithm (introduced in this paper) |
| PC | Personal Computer |
| P-CS | Pattern search based Cuckoo Search (introduced in this paper) |
| P-CSO | Pattern search based Cat Swarm Optimization (introduced in this paper) |
| P-FA | Pattern search based Firefly Algorithm (introduced in this paper) |
| P-GWO | Pattern search based Grey Wolf Optimization (introduced in this paper) |
| PID | Proportion Integration Differentiation |
| PLBA | Patch-Levy-based Bees Algorithm |
| P-MFO | Pattern search based Month Flame Optimization (introduced in this paper) |
| P-PSO | Pattern search based Particle Swarm Optimization (introduced in this paper) |
| PS | Pattern Search algorithm |
| P-SFLA | Patter search based Shuffled Foraging Leaping Algorithm (introduced in this paper) |
| PSNR | Peak Signal to Noise Ratio |
| PSO | Particle swarm optimization |
| P-WOA | Pattern search based Whale Optimization Algorithm (introduced in this paper) |
| QDE | Quantum Differential Evolution |
| QPSO | Quantum Particle Swarm Optimization |
| RABC | Refined Artificial Bee Colony algorithm |
| RAM | Random Access Memory |
| RiCE | Rician Classifier |
| S.S.R | Success Searching Ratio (statistical index of the experimental results) |
| SAR | Synthetic Aperture Radar |
| SFLA | Shuffled Frog Leaping Algorithm |
| SI | Swam Intelligent |
| SMP | Seeking Memory Pool (parameter in Cat Swarm Optimization) |
| SRD | Seeking Range of the Dimension (parameter in Cat Swarm Optimization) |
| SSIM | Structural SIMilarity Index |
| SSO | Social Spider Optimization |
| TM | Trade Mark |
| TSMO | Two-Stage Multi-threshold Otsu method |
| Var. | Variance (statistical index of the experimental results) |
| WDO | Wind Driven Optimization |
| WOA | Whale Optimization Algorithm |

**References**

1. Sezgin M，Sankur B L．Survey over image thresholding techniques and quantitative performance evaluation[M]．Journal of Electronic Imaging，2004：146－168．
2. Gonzalez R C，Woods R E．Digital Image Processing[M]．Third Edition．BeiJing：Publishing House of Electronics Industry，2010．
3. Haralick R M，Shapiro L G．Survey：image segmentation techniques[J]．Computer Vision Graphics & Image Processing．1985，29(1)：100－132．
4. Fu K S，Mui J K．A survey on image segmentation[J]．Pattern Recognition．1981，13(1)：3－16．
5. Bhandari A K，Singh V K，Kumar A，et al．Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur’s entropy[J]．Expert Systems with Applications．2014，41(7)：3538－3560．
6. Wu Y，Meng T，Wu S．Research Progress of Image Thresholding Methods in Recent 20 Years(1994-2014)[J]．Journal of Data Acquisition & Processing．2015．
7. Kapur B J N，Sahoo P　K，Wong A　K　C．A New Method for Gray-level Picture Thresholding Using the Entropy[C]．2010．
8. Otsu N．A threshold selection method from level histograms[J]．1979．
9. Nie F，Zhang P．Image Segmentation Based on Framework of Two-dimensional Histogram and Class Variance Criterion[J]．International Journal of Signal Processing Image Processing & Pattern Recognition．2015，8．
10. Sheng B．An Three-Dimensional Threshold Image Segmentation Method Based on Mind Evolutionary Algorithm[J]．Electronics Quality．2016．
11. Sarkar S，Das S．Multilevel Image Thresholding Based on 2D Histogram and Maximum Tsallis Entropy-A Differential Evolution Approach[J]．IEEE Transactions on Image Processing．2013，22(12)：4788－4797．
12. Pan Z，Wu Y Q．Method of thresholding using two-dimensional exponent entropy and its fast algorithm[J]．Journal of Computer Applications．2007，27(4)：982－985．
13. Wu Y Q，Yin．Multi-threshold Selection Using Maximum Reciprocal Entropy/Reciprocal Gray Entropy[J]．Journal of Signal Processing．2013．
14. Wu Y Q，Cao P X，Wang K，et al．Two-dimensional Arimoto gray entropy thresholding[J]．Yingyong Kexue Xuebao/journal of Applied Sciences．2014，32(4)：331－340．
15. Bhandari A K，Kumar A，Singh G K．Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur’s，Otsu and Tsallis functions[J]．Expert Systems with Applications．2015，42(3)：1573－1601．
16. Zhang Y J，Yin Y，Hou X D，et al．Threshold Segmentation of Road Images Based on the Combination of P-tile and Histogram-Based Fuzzy C-Means Clustering[J]．Computer Engineering & Science．2008，30(10)：27－28．
17. Rosenfeld A，De L T P．Histogram concavity analysis as an aid in threshold selection[J]． IEEE Transactions on Systems Man & Cybernetics．1983，SMC－13(2)：231－235．
18. Hammouche K，Diaf M，Siarry P．A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem[J]．Engineering Applications of Artificial Intelligence．2010，23(5)：676－688．
19. Katsaggelos A K．Framework for efficient optimal multilevel image thresholding[J]．Journal of Electronic Imaging．2009，18(1)：427－432．
20. Zhu Q D，Jing L Q，Rong-Sheng B I，et al．Improvement Algorithm of Minimum-error Thresholding Segmentation Method[J]．Opto-Electronic Engineering．2010，37(7)：107－113．
21. Quan W Y，Jie Z X，Hua W S．Two-dimensional Cross Entropy Thresholding Based on Chaotic Resilient Particle Swarm Optimization or Decomposition[J]．Shanghai Jiaotong Daxue Xuebao/journal of Shanghai Jiaotong University．2011，45(3)：301－307．
22. Yin P Y，Chen L H．A fast iterative scheme for multilevel thresholding methods[J]．Signal Processing．1997，60(3)：305－313．
23. Liao P S，Chen T S，Chung P C．A Fast Algorithm for Multilevel Thresholding．[J]．Journal of Information Science & Engineering．2001，17(5)：713－727．
24. Boussaïd I，Lepagnot J，Siarry P．A survey on optimization metaheuristics[J]．Information Sciences．2013，237(237)：82－117．
25. Binitha S，Sathya S S．A Survey of Bio Inspired Optimization Algorithm[J]．International Journal of Soft Computing & Engineering．2012，2(2)．
26. Wolpert D H，Macready W G．No free lunch theorems for optimization[J]．IEEE Transactions on Evolutionary Computation．1997，1(1)：67－82．
27. Kennedy J，Eberhart R．Particle swarm optimization[C]．1995．
28. http://www.particleswarm.info/index.html[Z]．2016．
29. Gao H，Kwong S，Yang J，et al．Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation[J]．Information Sciences．2013，250(11)：82－112．
30. Eusuff M M，Lansey K E．Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm[J]．American Society of Civil Engineers．2003，129(3)：210－225．
31. Elbeltagi E，Hegazy T，Grierson D．A modified shuffled frog-leaping optimization algorithm：applications to project management[J]．Structure & Infrastructure Engineering．2007，3(1)：53－60．
32. Huynh T H．A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers[C]．2008．
33. Fang Y，Yu J．Application of Shuffled Frog-Leaping Algorithm in Web’s Text Cluster Technology[M]．Springer Berlin Heidelberg，2011：363－368．
34. Xu Y，Wang L，Wang S．An effective shuffled frog-leaping algorithm for the flexible job-shop scheduling problem[C]．2013．
35. Karaboga D．An Idea Based on Honey Bee Swarm for Numerical Optimization，Technical Report-TR06[J]．2005．
36. Karaboga D，Gorkemli B，Ozturk C，et al．A comprehensive survey：artificial bee colony (ABC) algorithm and applications[J]．Artificial Intelligence Review．2014，42(1)：21－57．
37. Karaboga D，Akay B．A comparative study of Artificial Bee Colony algorithm[J]．Applied Mathematics & Computation．2009，214(1)：108－132．
38. Mernik M，Liu S H，Karaboga D，et al．On clarifying misconceptions when comparing variants of the Artificial Bee Colony Algorithm by offering a new implementation[J]．Information Sciences．2015，291(C)：115－127．
39. Chu S C，Tsai P，Pan J S．Cat Swarm Optimization[J]．Lecture Notes in Computer Science．2006，6：854－858．
40. Saha S K，Ghoshal S P，Mandal D，et al．Cat Swarm Optimization algorithm for optimal linear phase FIR filter design．[J]．Isa Transactions．2013，52(6)：781－794．
41. Ma B，Ye C．The research of flow-shop scheduling problem based on cat swarm optimization[J]．Modern Manufacturing Engineering．2014．
42. Yang X S．Nature－Inspired Metaheuristic Algorithms[M]．Luniver Press，2010．
43. Yang X S，He X．Firefly Algorithm：Recent Advances and Applications[J]．International Journal of Swarm Intelligence．2013，1(1)．
44. Yang X S，Deb S．Cuckoo Search via Levy Flights[C]．2010．
45. Fister I，Yang X S，Fister D，et al．Cuckoo Search：A Brief Literature Review[J]．Physics．2014，516(2)：49－62．
46. Yang X S，He X．Bat algorithm：literature review and applications[J]．International Journal of Bio-Inspired Computation．2013，5(3)：141－149．
47. Gandomi A H，Yang X S．Chaotic bat algorithm[J]．Journal of Computational Science．2014，5(2)：224－232．
48. Bahmani-Firouzi B，Azizipanah-Abarghooee R．Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm[J]．International Journal of Electrical Power & Energy Systems．2014，56(3)：42－54．
49. Zhu B，Zhu W，Liu Z，et al．A Novel Quantum-Behaved Bat Algorithm with Mean Best Position Directed for Numerical Optimization[J]．Computational Intelligence & Neuroscience．2016，2016(2)：1－17．
50. Passino K M．Bacterial Foraging Optimization[J]．Innovations & Developments of Swarm Intelligence Applications．2010，1(1)：1－16．
51. Zhiwei Y，Qinyun L，Mengdi Z，et al．Image Segmentation Using Thresholding and Artificial Fish－Swarm Algorithm[J]．Journal of Software．2012，7(5)．
52. Dorigo M，Caro G D．The ant colony optimization meta-heuristic[C]．1999．
53. Mucherino A，Seref O．Monkey search：a novel metaheuristic search for global optimization[C]．2007．
54. Gandomi A H，Alavi A H．Krill herd：A new bio-inspired optimization algorithm[J]．Communications in Nonlinear Science & Numerical Simulation．2012，17(12)：4831－4845．
55. Gholizadeh S，Poorhoseini H．Optimum design of steel frame structures by a modified Dolphin echolocation algorithm[J]．Structural Engineering & Mechanics．2015，55(3)：535－554．
56. Mirjalili S，Mirjalili S M，Lewis A．Grey Wolf Optimizer[J]．Advances in Engineering Software．2014，69(3)：46－61．
57. Mirjalili S．How effective is the Grey Wolf optimizer in training multi-layer perceptrons[J]．Applied Intelligence．2015，43(1)：150－161．
58. Sulaiman M H，Mustaffa Z，Mohamed M R，et al．Using the gray wolf optimizer for solving optimal reactive power dispatch problem[J]．Applied Soft Computing．2015，32(C)：286－292．
59. Mirjalili S．The Ant Lion Optimizer[J]．Advances in Engineering Software．2015，83(6)：80－98．
60. Rebecca N，Shin M，M．H．S，et al．Ant Lion Optimizer for Optimal Reactive Power Dispatch Solution[J]．2015．
61. Trivedi I N，Parmar S A，Bhesdadiya R H，et al．Voltage stability enhancement and Voltage Deviation Minimization using ant-lion optimizer algorithm[C]．2016．
62. Mirjalili S．Moth－flame optimization algorithm：A novel nature-inspired heuristic paradigm[J]．Knowledge-Based Systems．2015，89：228－249．
63. Soliman G M A，Khorshid M M H，Abou-El-Enien T H M．MODIFIED MOTH－FLAME OPTIMIZATION ALGORITHMS FOR TERRORISM PREDICTION[J]．2016．
64. Zawbaa H M，Emary E，Parv B，et al．Feature Selection Approach based on Moth-Flame Optimization Algorithm[C]．2016．
65. Mirjalili S，Lewis A．The Whale Optimization Algorithm[J]．Advances in Engineering Software．2016，95：51－67．
66. Kaveh A，Ghazaan M I．Enhanced Whale Optimization Algorithm for Sizing Optimization of Skeletal Structures[J]．2016．
67. Jangir P，Trivedi I N，Jangir N，et al．A Novel Adaptive Whale Optimization Algorithm for Global Optimization[J]．2016．
68. Pare S，Kumar A，Bajaj V，et al．A multilevel color image segmentation technique based on cuckoo search algorithm and energy curve[J]．Applied Soft Computing．2016，47：76－102．
69. Bhandari A K，Kumar A，Singh G K．Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms[J]．Expert Systems with Applications．2015，42(22)：8707－8730．
70. Tuba M．Multilevel image thresholding by nature-inspired algorithms：A short review[J]．Computer Science Journal of Moldova．2014．
71. Yin P Y．Multilevel minimum cross entropy threshold selection based on particle swarm optimization[J]．Applied Mathematics & Computation．2007，184(2)：503－513．
72. Maitra M，Chatterjee A．A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding[J]．Expert Systems with Applications An International Journal．2008，34(2)：1341－1350．
73. Gao H，Xu W，Sun J，et al．Multilevel Thresholding for Image Segmentation Through an Improved Quantum-Behaved Particle Swarm Algorithm[J]．IEEE Transactions on Instrumentation & Measurement．2010，59(4)：934－946．
74. Akbar H，Suryana N，Sahib S．Multilevel Thresholding Method Based on Aggressive Particle Swarm Optimization[C]．2011．
75. Shen Y，Wu Y，Ji Y．Multilevel Thresholding Based on Exponent Gray Entropy and Niche Chaotic Particle Swarm Optimization[M]．2011：617－624．
76. Jiang Y，Hao Z，Yuan G，et al．Multilevel thresholding for image segmentation through Bayesian particle swarm optimisation[J]．International Journal of Modelling Identification & Control．2012，15(4)：267－276．
77. Apoorva N，Ramesh D，Manikantan K，et al．Optimal multilevel thresholding based on Tsallis entropy using Fibonacci Particle Swarm Optimization for improved Image Segmentation[C]．2012．
78. Manikantan K，Arun B V，Yaradoni D K S．Optimal Multilevel Thresholds based on Tsallis Entropy Method using Golden Ratio Particle Swarm Optimization for Improved Image Segmentation[J]．Procedia Engineering．2012，30(1)：364－371．
79. Dey S，Bhattacharyya S，Maulik U．Quantum inspired meta-heuristic algorithms for multi－level thresholding for true colour images[C]．2013．
80. Liu Y，Mu C，Kou W，et al．Modified particle swarm optimization-based multilevel thresholding for image segmentation[J]．Soft Computing．2015，19(5)：1311－1327．
81. Li Y，Jiao L，Shang R，et al．Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation[J]．Information Sciences．2015，294：408－422．
82. Zhao X，Turk M，Li W，et al．A multilevel image thresholding segmentation algorithm based on two-dimensional K&L divergence and modified particle swarm optimization[J]．Applied Soft Computing．2016：151－159．
83. Horng M H．Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization[J]．Expert Systems with Applications．2009，37(6)：4580－4592．
84. Horng M H．A multilevel image thresholding using the honey bee mating optimization[J]．Applied Mathematics & Computation．2010，215(9)：3302－3310．
85. Ahmed L J，Jeyakumar A E．Maximum tsallis entropy thresholding for image segmentation using a refined artificial bee colony optimization[J]．International Review on Computers & Software．2013，8(8)：1923－1930．
86. Charansiriphaisan K，Chiewchanwattana S，Sunat K．A Comparative Study of Improved Artificial Bee Colony Algorithms Applied to Multilevel Image Thresholding[J]．Mathematical Problems in Engineering．2013，2013(6)：1－17．
87. Cuevas E，Sención F，Zaldivar D，et al．A Multi-threshold Segmentation Approach Based on Artificial Bee Colony Optimization[J]．Applied Intelligence．2014，37(3)：321－336．
88. Li J Y，Zhao Y D，Li J H，et al．Artificial Bee Colony Optimizer with Bee-to-Bee Communication and Multi population Coevolution for Multilevel Threshold Image Segmentation[J]．Mathematical Problems in Engineering．2015，2015：1－23．
89. Hussein W A，Sahran S，Abdullah S N H S．A fast scheme for multilevel thresholding based on a modified Bees Algorithm[J]．Knowledge-Based Systems．2016，101：114－134．
90. Jiang Y，Yeh W C，Hao Z，et al．A cooperative honey bee mating algorithm and its application in multi-threshold image segmentation[J]．Information Sciences．2016，369：171－183．
91. Sathya P D，Kayalvizhi R．Optimal multilevel thresholding using bacterial foraging algorithm[J]．Expert Systems with Applications An International Journal．2011，38(12)：15549－15564．
92. Sanyal N，Chatterjee A，Munshi S．An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation[J]．Expert Systems with Applications．2011，38(12)：15489－15498．
93. Bakhshali M A，Shamsi M．Segmentation of color lip images by optimal thresholding using bacterial foraging optimization (BFO)[J]．Journal of Computational Science．2013，5(2)：251－257．
94. Sathya P D，Kayalvizhi R．Modified bacterial foraging algorithm based multilevel thresholding for image segmentation[J]．Engineering Applications of Artificial Intelligence．2011，24(4)：595－615．
95. Sathya P D，Kayalvizhi R．Amended bacterial foraging algorithm for multilevel thresholding of magnetic resonance brain images[J]．Measurement．2011，44(10)：1828－1848．
96. Wang N，Li X，Chen X H．Fast three-dimensional Otsu thresholding with shuffled frog－leaping algorithm[J]．Pattern Recognition Letters．2010，31(13)：1809－1815．
97. Horng M H．Multilevel minimum cross entropy threshold selection with shuffled frog－leaping algorithm[J]．Advances in Information Sciences & Service Sciences．2012．
98. Ladgham A，Hamdaoui F，Sakly A，et al．Fast MR brain image segmentation based on modified Shuffled Frog Leaping Algorithm[J]．Signal Image & Video Processing．2015，9(5)：1113－1120．
99. Horng M H，Liou R J．Multilevel minimum cross entropy threshold selection based on the firefly algorithm[J]．Expert Systems with Applications．2011，38(12)：14805－14811．
100. Raja N，Rajinikanth V，Latha K．Otsu Based Optimal Multilevel Image Thresholding Using Firefly Algorithm[J]．Modelling & Simulation in Engineering．2014，2014(2)：1－17．
101. Chen K，Zhou Y，Zhang Z，et al．Multilevel Image Segmentation Based on an Improved Firefly Algorithm[J]．Mathematical Problems in Engineering．2016，2016(285–296)：1－12．
102. Alihodzic A，Tuba M．Improved bat algorithm applied to multilevel image thresholding[J]．Scientific World Journal．2014，2014：176718．
103. Ye Z W，Wang M W，Liu W，et al．Fuzzy entropy based optimal thresholding using bat algorithm[J]．Applied Soft Computing．2015，31(C)：381－395．
104. Panda R，Agrawal S，Bhuyan S．Edge Magnitude based Multilevel Thresholding using Cuckoo Search Technique[J]．Expert Systems with Applications．2013，40(18)：7617－7628．
105. Agrawal S，Panda R，Bhuyan S，et al．Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm[J]．Swarm & Evolutionary Computation．2013，11：16－30．
106. Brajevic I，Tuba M．Cuckoo Search and Firefly Algorithm Applied to Multilevel Image Thresholding[M]．Springer International Publishing，2014：115－139．
107. Ouadfel S，Taleb－Ahmed A．Social spiders optimization and flower pollination algorithm for multilevel image thresholding：A performance study[J]．Expert Systems with Applications An International Journal．2016，55(C)：566－584．
108. Agarwal P，Singh R，Kumar S，et al．Social Spider Algorithm Employed Multi-level Thresholding Segmentation Approach[M]．Springer International Publishing，2016．
109. Dey S，Bhattacharyya S，Maulik U．New Quantum Inspired Meta-heuristic Techniques for Multi-level Colour Image Thresholding[J]．Applied Soft Computing．2016，46：677－702．
110. Akay B．A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding[J]．Applied Soft Computing．2013，13(6)：3066－3091．
111. Jaynes E T．On the rationale of maximum-entropy methods[J]．Proceedings of the IEEE．1982，70(9)：939－952．
112. Hooke R，Jeeves T A．Direct search" solution of numerical and statistical problems[J]．Journal of the Acm．1961，8(2)：212－229．
113. Chen B L．Optimization：Theory and Algorithm[M]．Beijing：Tsinghua Press，2013．
114. Awadallah M A，Al-Betar M A，Khader A T，et al．Hybridization of harmony search with hill climbing for highly constrained nurse rostering problem[J]．Neural Computing & Applications．2015：1－20．
115. Wang H，Wang D，Yang S．A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems[J]．Soft Computing．2009，13(13)：763－780．
116. Renders J M，Bersini H．Hybridizing genetic algorithms with hill－climbing methods for global optimization：two possible ways[C]．1994．
117. Shi Y，Eberhart R．Modified particle swarm optimizer[C]．1998．
118. Yang X．A new metaheuristic bat-inspired algorithm[J]．2010．