Applied Soft Computing 49 (2016) 146–163

Contents lists available at ScienceDirect

Applied Soft Computing

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

Effective optic disc detection method based on swarm intelligence techniques and novel pre-processing steps

Sa'ed Abed*, Suood Abdulaziz Al-Roomi, Mohammad Al-Shayeji

Computer Engineering Department, College of Computing Sciences and Engineering, Kuwait University, P O Box 5969, Safat, 13060, Kuwait

ARTICLE INFO

Article history:
Received 7 April 2016
Received in revised form 5 July 2016
Accepted 5 August 2016
Available online 11 August 2016

Keywords:
Diabetic retinopathy
Swarm intelligence techniques
Optic disc detection
Pre-processing
Speed and accuracy

ABSTRACT

Diabetic retinopathy affects the vision of a significant fraction of the population worldwide. Retinal fundus images are used to detect the condition before vision loss develops to enable medical interventions. Optic disc detection is an essential step for the automatic detection of the disease. Several techniques have been introduced in the literature to detect the optic disc with different performance characteristics such as speed, accuracy and consistency. For optic disc detection, a nature-inspired algorithm called swarm intelligence has been shown to have clear superiority in terms of speed and accuracy compared to traditional detection algorithms. We therefore further investigated and compared several swarm intelligence techniques. Our study focused on five popular swarm intelligence algorithms: artificial bee colony, particle swarm optimization, bat algorithm, cuckoo search and firefly algorithm. This work also featured a novel pre-processing scheme that enhances the detection accuracy of the swarm techniques by making the optic disc region the highest grayscale value in the image. The pre-processing involves multiple stages of background subtraction, median filtering and mean filtering and is named Background Subtraction-based Optic Disc Detection (BSODD). The best result was obtained by combining our pre-processing technique, firefly algorithm and the parameters used for the algorithm. The obtained accuracy was superior to the other tested algorithms and published results in the literature. The accuracy of the firefly algorithm was 100%, 100%, 98.82% and 95% when using the DRIVE, DiaRetDB1, DMED and STARE databases, respectively.

1. Introduction

More than 171 million people suffer from Diabetes worldwide [1]. Diabetic retinopathy is a micro-vascular complication of diabetes and it affects 10–50% of patients who have diabetes [1]. Research shows that 4.4% of the US population who are 40 years or older suffers from vision threatening diabetic retinopathy [2]. One of the main obstacle in diagnosing diabetic retinopathy is that it has no external symptoms until vision loss develops. Hence, periodic screening is necessary to detect the condition in diabetic patients. An effective measure to detect the existence of the disease is to assess fundus images of diabetic patients. A fundus image is an image of the interior of the eye that is captured using a process called fundus photography. An example of a fundus image is shown in Fig. 1. Such images can be used to diagnose diseases affecting the retina, including diabetic retinopathy.

The first step in detecting the disease is to correctly highlight the optic disc. The optic disc is a bright yellow disc in the retina from which blood vessels emerge. The optic disc is marked in the retinal fundus image shown in Fig. 1. The optic disc is similar in color and contrast to lesions due to diabetic retinopathy (DR). Therefore, the disc must be masked from the fundus image before further processing for DR detection. Segmentation of blood vessels from the fundus image also requires optic disc detection because the disc is used as a reference location for other features in the image. Manual examination of the disease using fundus images is very tedious, time consuming and expensive because the services of a trained ophthalmologist are required. Consequently, various image-processing techniques have been proposed to automate the detection of DR [3–5].

For optic disc detection in particular, a variety of proposals are available in the literature [6–17]. Sinthayothin et al. [13] proposed an approach using a region of maximum variance in the image for detecting the optic disc. Morphological filtering and watershed transformation was used by Walter et al. [14]. A set of techniques including pyramidal decomposition, edge detection and entropy filter was used by Qureshi et al. to locate the optic disc [15]. Hsiao

* Corresponding author.
E-mail addresses: s.abed@ku.edu.kw (S. Abed), eng.saoud.kw@gmail.com (S.A. Al-Roomi), m.shayeji@ku.edu.kw (M. Al-Shayeji).

http://dx.doi.org/10.1016/j.asoc.2016.08.015
1568-4946/

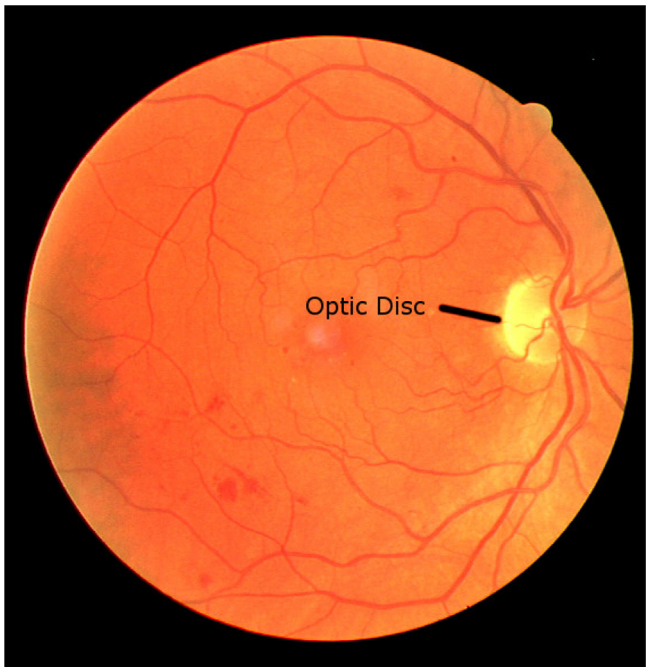

Fig. 1. An example of a fundus image: The optic disc is marked in the image.

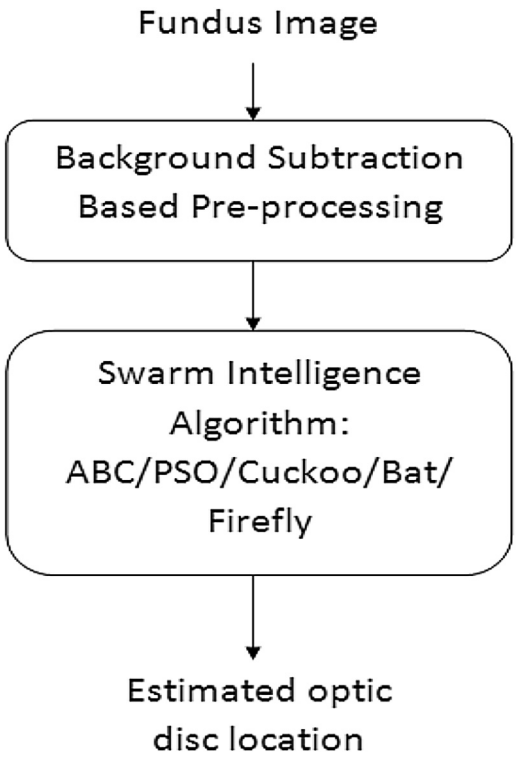

Fig. 2. Algorithm flowchart of the Background Subtraction based Optic Disc Detection (proposed method).

et al. [16] used an illumination correction method to locate the optic disc.

In addition to the above techniques, a set of interesting optic detection techniques inspired by nature, called Swarm techniques, has yielded very high accuracy and speed results. Pereira et al. [12] proposed a method based on anisotropic diffusion followed by edge detection using the ant colony algorithm for this problem. Another approach proposed by Rahebi et al. [11] involves the use of median filtering followed by the firefly algorithm.

The superior results from the two swarm intelligence-based algorithms inspired us to further investigate other nature-inspired techniques for optic disc detection. In this paper, we applied five different swarm intelligence algorithms and a novel pre-processing method to further improve the accuracy of such techniques over the state of the art in the literature. The problem also provides a good opportunity to compare the five algorithms. A similar comparison of nature-inspired algorithms was reported previously for multi-thresholding based segmentation of blood micrographs [18]. In that paper, the authors compared three algorithms, i.e., differential evolution, particle swarm optimization and artificial bee colony algorithms, and determined that differential evolution exhibited superior performance compared to the other two for this problem.

The five swarm intelligence algorithms that were used in this work are artificial bee colony [19], particle swarm optimization [20], bat algorithm [21], cuckoo search [22] and firefly algorithm [23]. Karaboga et al. [19] proposed the artificial bee colony algorithm inspired by the swarming behaviour of bees around their hives. Particle swarm optimization [20] is a powerful optimization method inspired by the flocking behaviour of birds that was proposed by Eberhart and Kennedy. The echolocation behaviour of bats inspired the bat algorithm [21] proposed by Yang. Ramin Rajabioun proposed the cuckoo search algorithm [22] based on the egg laying and breeding behaviour of a bird species called the cuckoo. The flashing behaviour of fireflies inspired another powerful algorithm known as the firefly algorithm [23] proposed by Yang.

In this paper, we propose an algorithm named Background Subtraction-based Optic Disc Detection (BSODD). BSODD first performs extensive pre-processing to make the optic disc the most accessible peak in the processed image and then uses an optimization algorithm to find the optic disc. The peak is the pixel with the highest grayscale value in an image. An overview of the proposed method is given in Fig. 2. First, the retinal fundus image is pre-processed using a series of filtering, transformation and background subtraction operations to eliminate false peaks and illumination variations in the background. Subsequent masking of the resultant image eliminates the regions of the image where the optic disc is unlikely to occur. Finally, one of the five optimization algorithms is applied to the image to find the peak, which is selected as the optic disc location.

The algorithm was tested on four publicly available databases and yielded superior results compared to other methods reported in the literature. Of the five algorithms used, the firefly algorithm exhibited the best performance. The detection accuracy for the firefly algorithm was 100%, 100%, 98.82% and 95% when the databases DRIVE, DiaRetDB1, DMED and STARE, respectively, were used. The order of the algorithms in terms of detection accuracy from best to worst was firefly algorithm, artificial bee colony algorithm, cuckoo search, bat algorithm and particle swarm optimization. A detailed study of the variation impact of the parameters of the swarm intelligence algorithm was carried out with the corresponding plots are also included in this work. The main contributions of this paper are as follows:

- Proposal of a novel pre-processing technique to improve optic disc detection.
- Proposal of the use of swarm intelligence algorithms for optic disc detection, including algorithms that have not been used previously in the literature for optic disc detection (artificial bee colony, particle swarm optimization, bat algorithm, and cuckoo search).
- Comparison of five swarm algorithms in the context of detection accuracy in optic disc detection.

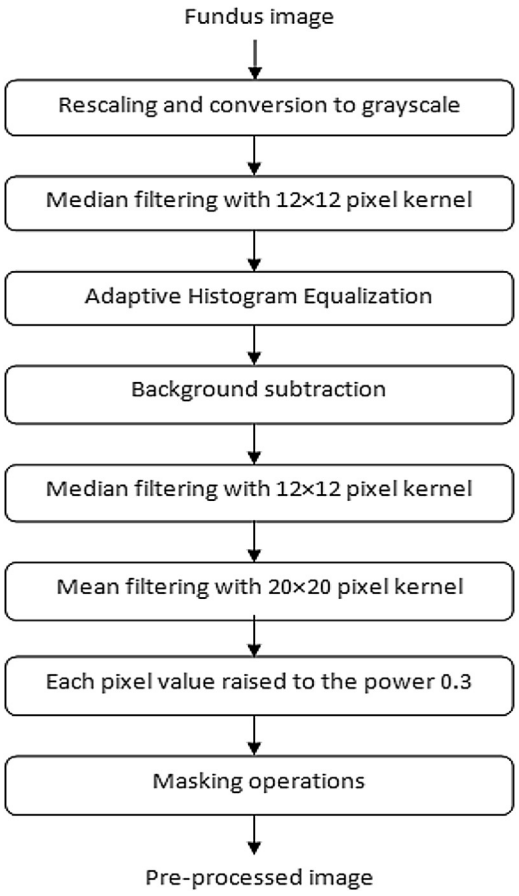

Fig. 3. Flow diagram of the pre-processing steps.

- Comparison of running time and consistency between the five swarm intelligence algorithms in the context of optic disc detection,
- Providing optimal parameters for each of the five studied swarm intelligence algorithms that netted suppirior for optic disc detection accuracy.
- A performance study of the impact of varying the swarm intelligence algorithm parameters on optic disc detection performance.

The reminder of this paper is organized as follows. Section 2 gives a description of the pre-processing steps used in the proposed approach. Section 3 describes the five swarm intelligence algorithms used in the work. Section 4 gives the results of the work and Section 5 gives a discussion of the results. Section 6 concludes the paper.

2. Pre-processing steps

The fundus image is pre-processed extensively to remove false peaks and to smoothen the image using a series of filtering, background subtraction and masking operations. The goal of the filtering operations is to eliminate the false peaks and to smoothen the image, whereas the masking operations eliminate the regions of the image where the optic disc is unlikely to occur. These pre-processing steps are essential and experimentally confirmed to increase the accuracy of optic disc detection. Fig. 3 presents the flow diagram of the pre-processing steps.

First, the fundus image is submitted and tested. If the height of the fundus image is greater than 400 pixels, it is rescaled to half the size. If the height is greater than 2000 pixels, it is rescaled to one-eighth the size. The rescaling maintains the image at a moderate size to facilitate the location of the optic disc by the optimization algorithms. After rescaling, the image is converted to a grayscale image.

The noise is removed from the resultant image by passing the image through median filtering with a 12×12 pixel kernel. The process involves replacing the grayscale values of the current pixel with the median of that pixel and its neighbors (in this case 12×12 neighbors). This process also helps remove some of the false peaks from the image. The value 12×12 is selected such that the kernel is sufficiently large to filter out the lesions without filtering out the optic disc. Next, the image is subjected to Adaptive Histogram Equalization (AHE) [24] to improve the contrast of the image.

After applying AHE, the background is estimated and subtracted from the image. The background is estimated using median filtering using a 110×110 pixel kernel. The kernel is selected such that it is sufficiently large to exclude the optic disc from the background image. Then, the resultant image is again median filtered using a 12×12 pixel kernel. The median filtering operation suppresses any false peaks that may appear after background subtraction. For further smoothening, the image is subjected to mean filtering with a 20×20 pixel kernel. The optimization algorithms can locate the optic disc more accurately when the image is smooth, i.e., it does not have a large number of local maxima and minima. This kernel size of 20×20 was experimentally determined to give the best performance. Finally, the grayscale value of each pixel is raised to the power 0.3, which was determined to yield the best result experimentally. This transformation resulted in the more accurate location of the optic disc by the optimization algorithms. Note that this operation does not change the order of grayscale values of the pixels. Hence, the pixel with the largest grayscale retains this status after the transformation.

The next step of the pre-processing includes masking off the regions of the image where the optic disc is unlikely to occur. All pixels that are outside the region of interest (ROI) are made zero. Here, ROI is the circular area with some portion at the edge clipped. The regions outside the ROI are the dark regions at the four corners, as shown in Fig. 1. Next, some pixels inside the ROI at the edge are also removed. Pixels that are within $10 \times W/128$ pixels (where W is the width of the image in pixels) from the edge of the ROI are removed. For a 128×128 pixel image, this operation will remove pixels that are within 10 pixels from the edge of the ROI. Pixels that are within $5 \times W/128$ pixels from the center of the image are also masked. For a 128×128 image, this will remove a circular region with a radius of 5 pixels from the center of the image. Further, rectangular regions of height $25 \times H/128$ pixels (where H is the height of the image in pixels) and the width of the image are masked from the top and bottom of the image. For a 128×128 image, this operation will remove a rectangular region with a height of 25 pixels from the top and bottom of the image. The authors determined experimentally that these masks produced the best results.

An example of a pre-processed image is shown on the right in Fig. 4. The original image is shown on the left. Notice that the pre-processed image is cropped on the top, bottom, left and right edges of the ROI. In addition, a circular region is removed from the center of the image as a result of the masking step.

With the above steps, the pre-processing stage is complete. This pre-processed image is a two dimensional function, with the function value being the intensity at each pixel location. This two dimensional function is considered as the objective function. The optimum point of the image, which is the point with the highest intensity, is found using one of the optimization algorithms discussed in the next section, and the result given by the optimization algorithm is taken as the optic disc location. For instance, the 3D surface plot of a fundus image after the above pre-processing is

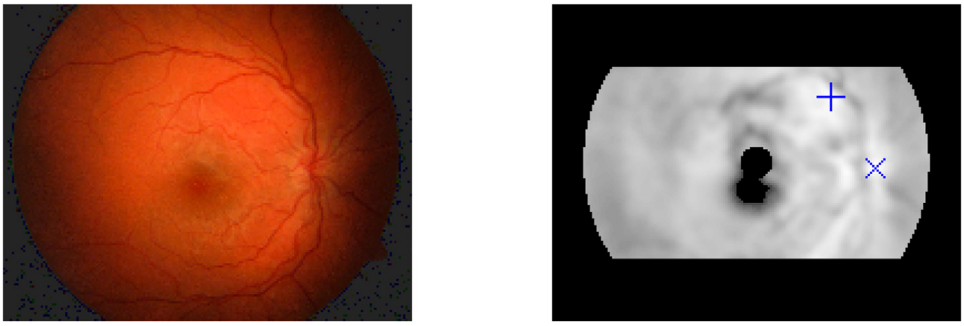

Fig. 4. Example of pre-processing: The original fundus image on left and the pre-processed image on right. Note the existence of a false peak even after pre-processing, where the pixel with the highest grayscale value is shown by '+' while the correct location of the optic disc found by the firefly algorithm is marked by 'X'.

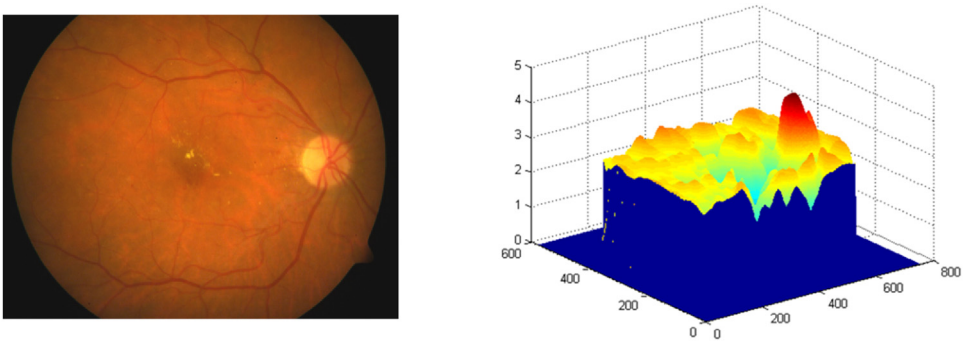

Fig. 5. (left) A fundus image from DiaRet dataset and (right) 3D surface plot of the pre-processed fundus image. The peak shown in red in the surface plot is the optic disc location. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

shown in Fig. 5. The role of the nature inspired algorithm is to find the highest peak in this image that is shown in red.

3. Swarm intelligence algorithms

This section describes the five swarm intelligence algorithms used in this paper: artificial bee colony, particle swarm optimization, bat algorithm, cuckoo search and firefly algorithm. All five swarm intelligence techniques are minimization algorithms. The function given as an input to the optimization algorithm for minimization is named the objective function. To find the optic disc, the algorithm locates the maximum grayscale value in the image. Then, to find the maximum of the function using these minimization algorithms, the objective function is defined as the negative of the grayscale value of the pixel of the input image.

3.1. Artificial bee colony (ABC) algorithm

The artificial bee colony algorithm [19] was proposed by Karaboga et al. The algorithm simulates the foraging behavior of bees to find good solutions to optimization problems. The artificial bees search for potential solutions that are named food sources. There are three types of bees: employed bees, onlooker bees and scouts. The employed bees fly to the food sources in their memory and search for new food sources around them. The onlooker bees choose a food source of one of the employed bees at random and find a food source around it. Some of the food sources are abandoned, and replacement food sources are found at random by scouts. The pseudocode of the artificial bee colony algorithm is shown in Algorithm 1 in Appendix A.

The parameters used in the ABC algorithm are given in Table 1. These parameters were determined based on experimental testing and analysis for better results.

Table 1
Parameters used in the ABC algorithm.

Parameter	Value
Number of food sources, *numFoodSources*	40
Limit to the trials, *limit*	200
Maximum number of iterations maxIter	5000

Table 2
Parameters used in the PSO algorithm.

Parameter	Value
Swarm size, S	256
maxIter	3000
Inertia, ω	1.3
Correction factor, ϕ_p	2.0

3.2. Particle swarm optimization

Particle swarm optimization (PSO) was proposed by Kennedy and Eberhart [20] and simulates the swarming behavior of birds and fish. Birds and fish tend to follow the leader to form a swarm while flying or swimming. In the PSO algorithm, the leader is the potential solution with the best value of the objective function, and the swarm is the set of all potential solutions. The potential solutions are known as particles, and they move through the space following the particles that have the current best value of the objective function. Each particle is influenced by the local best but also moves toward the best known location in search space. Each particle is assigned a velocity based on how far it is from its previous best position and from the global best position. The locations of the particles are updated based on their velocity. The pseudocode of the algorithm is given in Algorithm 2 in the Appendix A. The parameters used in the PSO algorithm were experimentally evaluated to give better results as shown in Table 2.

Table 3
Parameters used in the bat algorithm.

Parameter	Value
Population size	200
Loudness, A	0.5
Pulse rate, r	0.5
Max number of iterations	1500
f_{min}	0
f_{max}	2

Table 4
Parameters used in the cuckoo search algorithm.

Parameter	Value
Number of nests, n	400
Levy distribution parameter β	3/2
Levy distribution parameter σ_v	1
Maximum number of iterations *maxIter*	100
pa	0.5
a	0.01

3.3. Bat algorithm

The bat algorithm [21] proposed by Yang is inspired by the behavior of microbats, which use a type of sonar called echolocation to detect prey and avoid obstacles. The virtual microbats used in the algorithm find prey using sound waves at different wavelengths and amplitudes. The virtual bats move randomly based on the frequencies and velocities and are attracted toward locations with low values of the objective function. The pseudocode of the bat algorithm is shown in Algorithm 3 in Appendix A.

Table 3 presents the parameters used in the bat algorithm. We experimentally specified these parameters to obtain good results.

3.4. Cuckoo search algorithm

The cuckoo search algorithm [22] was proposed by Ramin Rajabioun and is inspired by the breeding behavior of cuckoo birds. The virtual cuckoos make random flights to find new solutions called nests. The new nests are kept if the objective function value is better than the current one, as long as the host bird does not discover the new nest. Discovery occurs with a pre-determined probability. If the host bird discovers the nest, it is discarded. The pseudocode of the algorithm is given in Algorithm 4 in Appendix A.

The parameters used in the cuckoo PSO algorithm are shown in Table 4. These parameters were experimentally calculated to obtain better results.

3.5. Firefly algorithm

Yan proposed the firefly algorithm [23] based on the flashing behavior of fireflies. Fireflies are attracted to each other based on the intensity of their flash. In this algorithm, the intensity of the flash depends on the value of the objective function at that location. Therefore, fireflies located near the minimum have more intense flashes, which attracts other fireflies toward regions that have lower values of the objective function. The pseudocode of the algorithm is given in Algorithm 5 in Appendix A.

The parameters used in the firefly algorithm are shown in Table 5. We experimentally specified these parameters to obtain good results.

Table 5
Parameters used in the firefly algorithm.

Parameter	Value
MaxGeneration	100
Population size, N	300
Initial value of α	0.9
δ	0.98
β_0	0.6
γ	0.00007

4. Evaluation results

The methods described earlier were tested and evaluated using four publicly available datasets described in Subsection 4.1. The results were then compared with each other and with the ones available in literature as explained in Subsection 4.2.

4.1. Datasets

The four public datasets that were used to evaluate the performance of the methods described were Digital Retinal Images for Vessel Extraction (DRIVE) [25], Standard Diabetic Retinopathy Database Calibration level 1 (DiaRetDB1) [26], Diabetic Macular Edema Database (DMED) [27], and STructured Analysis of the Retina (STARE) [28].

The DRIVE dataset consists of 40 images obtained from a diabetic screening program conducted in the Netherlands. Of these 40 images, 33 do not have any symptoms of DR, and the remaining 7 have mild symptoms of the disease.

The DiaRetDB1 dataset consists of 89 images; 84 of these images contain symptoms of DR, and the remaining 5 do not have any symptoms of the disease. The DMED dataset includes 169 images, of which 54 are pathological and thus contain lesions that indicate DR.

The STARE dataset contains 20 images, of which 10 are pathological and the remainder images are non-pathological. One of the 10 non-pathological images in this dataset does not contain the optic disc. We added this image to the set of images in which the optic disc is correctly detected while running all five algorithms.

4.2. Experiment

To determine whether the algorithm correctly finds the location of the optic disc, we first manually found the optic center (C) and radius (R) locations for each image in the four datasets by viewing the images using image-editing software and reading the pixel number when the pointer points to the center and rim of the optic disc. The automatic optic disc detection algorithms described earlier were subsequently executed, and the results of the algorithms were compared to the manual results. The optic disc was considered detected when the automatically found location of the optic disc was within a distance of R pixels from C. The experiments using all 5 algorithms were repeated four times, and the mean was determined. Four repeats were sufficient because the variation of results in each run was minimal, on the order of 1%.

The algorithms were executed using MATLAB on a computer with an Intel Core i7-4770k processor, 16 GB of RAM, and 64 bit Windows 7 operating system. The mean of the results was compared to other results available in the literature as shown in Table 6.

The mean running times of the algorithms for different datasets are given in Table 7. For both tables, the best value(s) are highlighted in green, and the worst are highlighted in red.

The individual simulation results obtained by running the algorithms using the four datasets are shown in Fig. 6.

Fig. 6 represents the normalized error in optic disc localization, which is found as shown in Eq. (1). The normalized error in optic disc localization is the distance between the actual optic disc location and the location found by the swarm intelligence algorithm

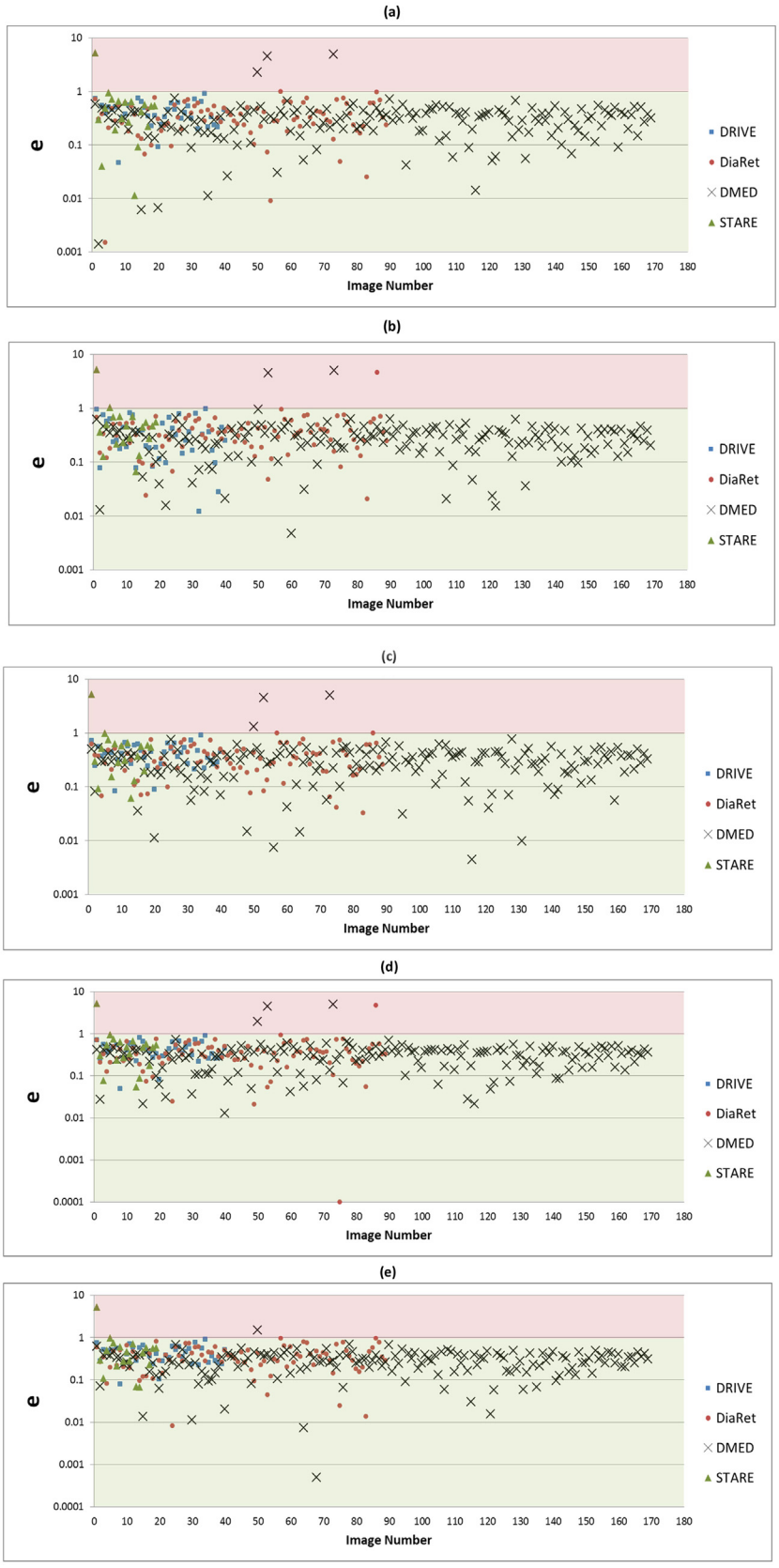

Fig. 6. Simulation results obtained by running the BSODD algorithm using (a) ABC, (b) PSO, (c) Bat (d) Cuckoo and (e) Firefly. The letter e in the y-axis is the normalized error in optic disc localization as shown in Eq. (1). The shapes represent the following: + for DRIVE, X for DiaRetDB1, square for DMED and diamond for STARE. Points in the red area indicate the images for which optic discs are not localized correctly. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Detection accuracy results compared with previously published work. The algorithm with the highest accuracy for each dataset is highlighted in green.

		Is it swarm Intelligence technique?	Detection Accuracy (%)			
			DRIVE	DiaRetDB1	DMED	STARE
Previously	Sinthanayothin et al. [13]	No	60	–	–	
	Walter et al. [14]	No	80	–	–	
	Qureshi et al. [15]	No	100	94.02	–	
	Hsiao et al. [16]	No	100	–	–	
	Pereira et al. [12]	Yes, Ant Colony	100	93.25	–	
	Rahebi et al. [11]	Yes, firefly	100	94.38	–	95
Background Subtraction-based Optic Disc Detection (proposed work)	Artificial Bee Colony		100	100	98.22	90
	Particle Swarm Optimization		100	98.60	98.08	91.25
	Bat Algorithm		100	98.88	97.93	93.75
	Cuckoo Search		100	99.44	98.22	92.50
	Firefly Algorithm		100	100	98.82	95

Table 7
The running time of the five swarm intelligence algorithms tested. The fastest algorithm for each dataset is highlighted in green, and the slowest is highlighted in red.

Algorithm	Running time (s)			
	DRIVE	DiaRetDB1	DMED	STARE
Artificial Bee Colony	10.02	10.10	9.70	9.74
Particle Swarm Optimization	1.86	1.86	1.85	1.87
Bat Algorithm	5.75	4.51	5.43	5.44
Cuckoo Search	1.37	1.49	1.09	1.13
Firefly Algorithm	1.35	1.44	1.00	1.09

divided by the actual radius of the optic disc.

$$e = Normalized\ \ error\ \ in\ \ optic\ \ disc\ \ localization = \frac{||Actual\ \ Optic\ \ Disc\ Location - Swarm\ \ Intelligence\ \ Algorithm\ \ Estimated\ \ Optic\ \ Disc\ \ Location||}{Actual\ \ Radius\ \ of\ \ the\ \ Optic\ Discinpixels} \quad (1)$$

where $||x-y||$ is the distance between points x and y in pixels. In this equation, *ActualOpticDiscLocation* represents the manually located center of the optic disc and *Swarm Intelligence Algorithm Estimated Optic Disc Location* is the estimate of the optic disc location found by the swarm intelligence algorithm. The difference between the location found manually and through any of the swarm intelligence algorithm represents the error in the estimate of the optic disc location from the swarm intelligence algorithm in numbers of pixels. This value is then divided by the *Actual Radius of the Optic Discinpixels*, which is the manually found radius of the optic disc.

A value less than one in the graphs indicates correct detection, while a value greater than one indicates a missed detection. The aim is to find if the point estimated by the swarm intelligence algorithm is within the manually found optic disc, in which case the value of normalized error will be less than one. A detailed discussion of the results is provided in the next section.

5. Discussion

In this work, optic disc detection was achieved by first pre-processing the fundus image and then using one of the five optimization algorithms described earlier. An elaborate set of pre-processing steps were performed on the image to eliminate small lesions and background illumination variations from the fundus image. For most images, after the pre-processing step, the most identifiable peak was inside the optic disc. Although there were still some false peaks inside some of the images, the optimization algorithms were able to correctly identify the optic disc in most cases, as these were the most accessible maxima. An example of a false peak is shown in Fig. 4. The original fundus image is shown on the left, and the pre-processed image is shown on the right. In the pre-processed image, the pixel with the highest grayscale value is indicated by '+', and the correct location of the optic disc found by the firefly algorithm is marked by an 'X'.

5.1. Swarm intelligence detection evaluation

A set of samples from each of the datasets used in which all of the algorithms performed very well is given in Fig. 7. An 'X' marks the location of the optic disc found by each algorithm. The algorithms worked well, even in the presence of variation in background illumination.

An image in which none of the algorithms gave correct results is shown in Fig. 8. The failure to correctly recognize the optic disc occurred mainly because the lesions in the image are very bright and the optic disc is relatively dark.

Another fundus image from the DiaRetDB1 dataset in which some algorithms gave the correct result and some other algorithms failed is shown in Fig. 9. In this image as well, the optic disc is not very bright, and the background is bright, confusing some of the algorithms. Here, the ABC algorithm and firefly algorithm found the correct location, whereas the PSO, bat algorithm and cuckoo search failed.

5.2. Swarm intelligence accuracy, running time and consistency evaluation

Our BSODD algorithm, in which one of the five tested swarm intelligence algorithms was preceded by our proposed pre-processing technique, yielded superior accuracy and running time compared to previously published work of the swarm type. The superiority of swarm intelligence algorithms may be due to the ability of such algorithms to find the global minimum even in functions that contain multiple local minima. The innovative pre-processing techniques also helped the swarm intelligence algorithms achieve good results.

In terms of detection accuracy, the firefly algorithm had the highest accuracy, followed by the ABC algorithm and cuckoo search, both of which achieved the same detection performance. The bat algorithm was slightly inferior, and PSO had the worst detection performance.

For running time, the firefly algorithm was the fastest of the 5 swarm intelligence technique. The cuckoo search was slightly slower than the firefly algorithm and was followed in order by the PSO algorithm, the bat algorithm, and ABC algorithm.

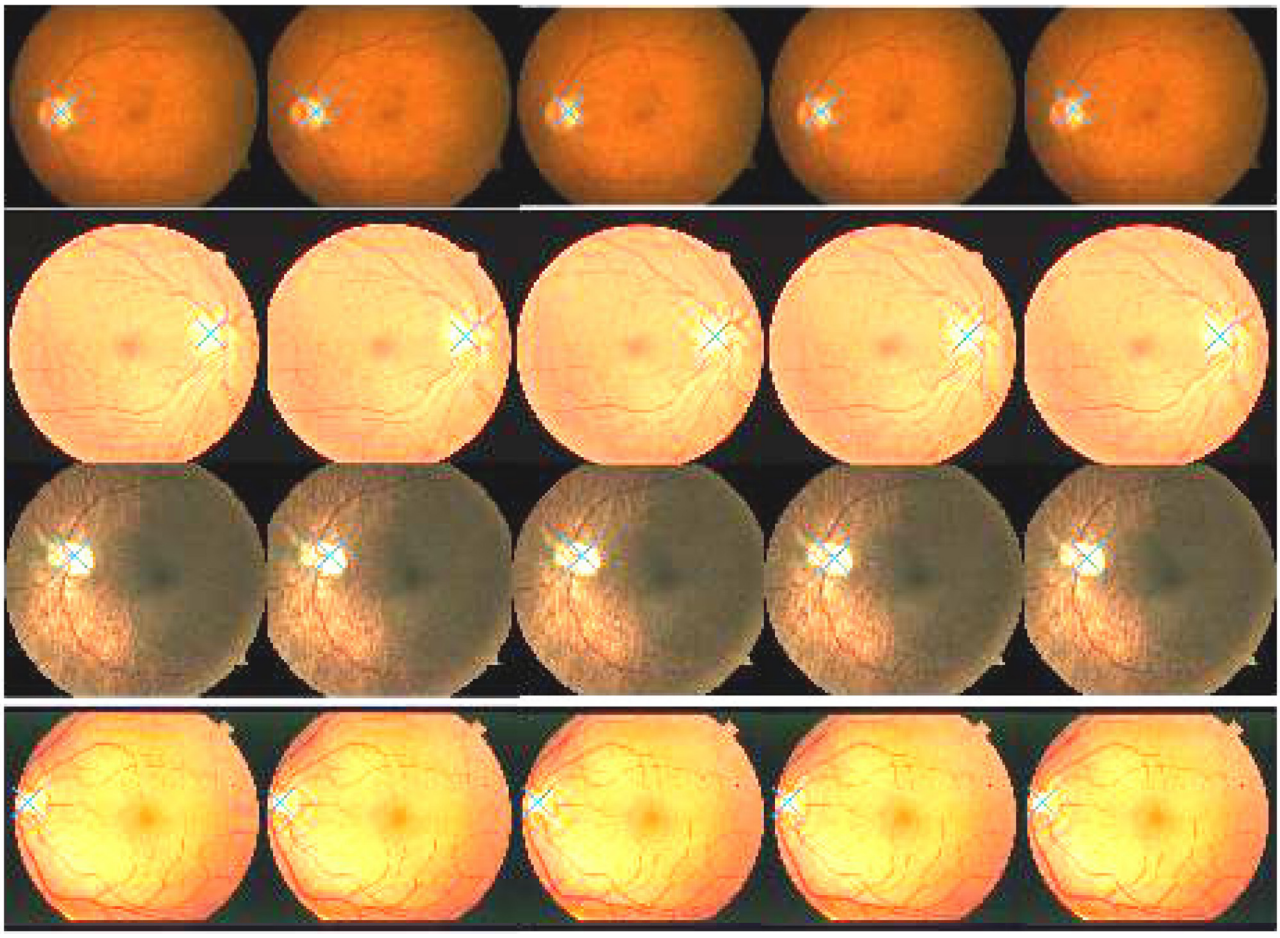

Fig. 7. A set of images in which the proposed algorithms worked well: Samples from the datasets (from top to bottom) DiaRetDB1, DRIVE, DMED and STARE. The optic disc was detected using the algorithms (from left to right) ABC, PSO, Bat, Cuckoo and Firefly. A cross marks the detected location of the optic disc.

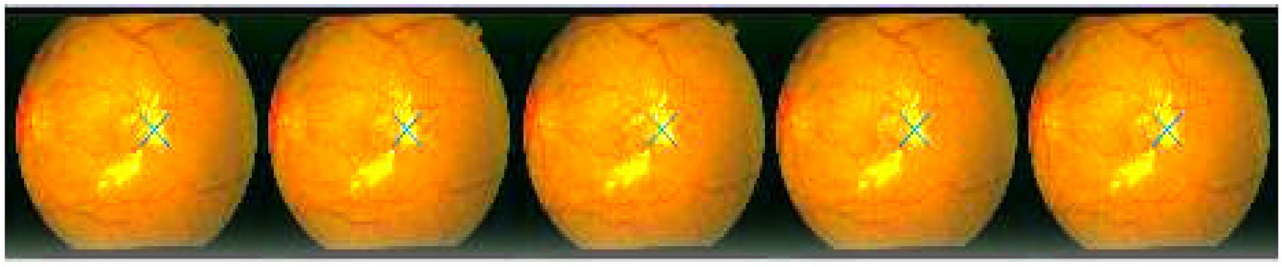

Fig. 8. Images from STARE dataset for which none of the BSODD algorithms found the correct location of the optic disc: The algorithms detected the locations of the lesions. The algorithms used were (from left to right) ABC, PSO, Bat, Cuckoo and Firefly. A cross marks the detected location of the optic disc.

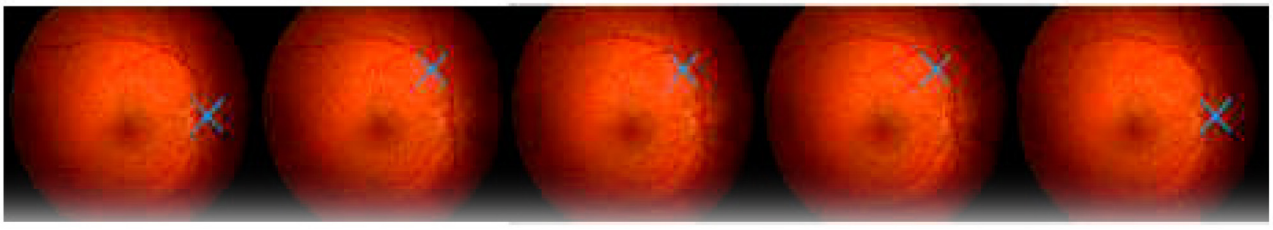

Fig. 9. An image from the DiaRetDB1 dataset in which some of the algorithms detected the optic disc correctly and some failed. Algorithms used (from left to right): ABC, PSO, Bat, Cuckoo and Firefly. A cross marks the detected location of the optic disc. The ABC algorithm and Firefly algorithm gave the correct result, whereas the other algorithms failed.

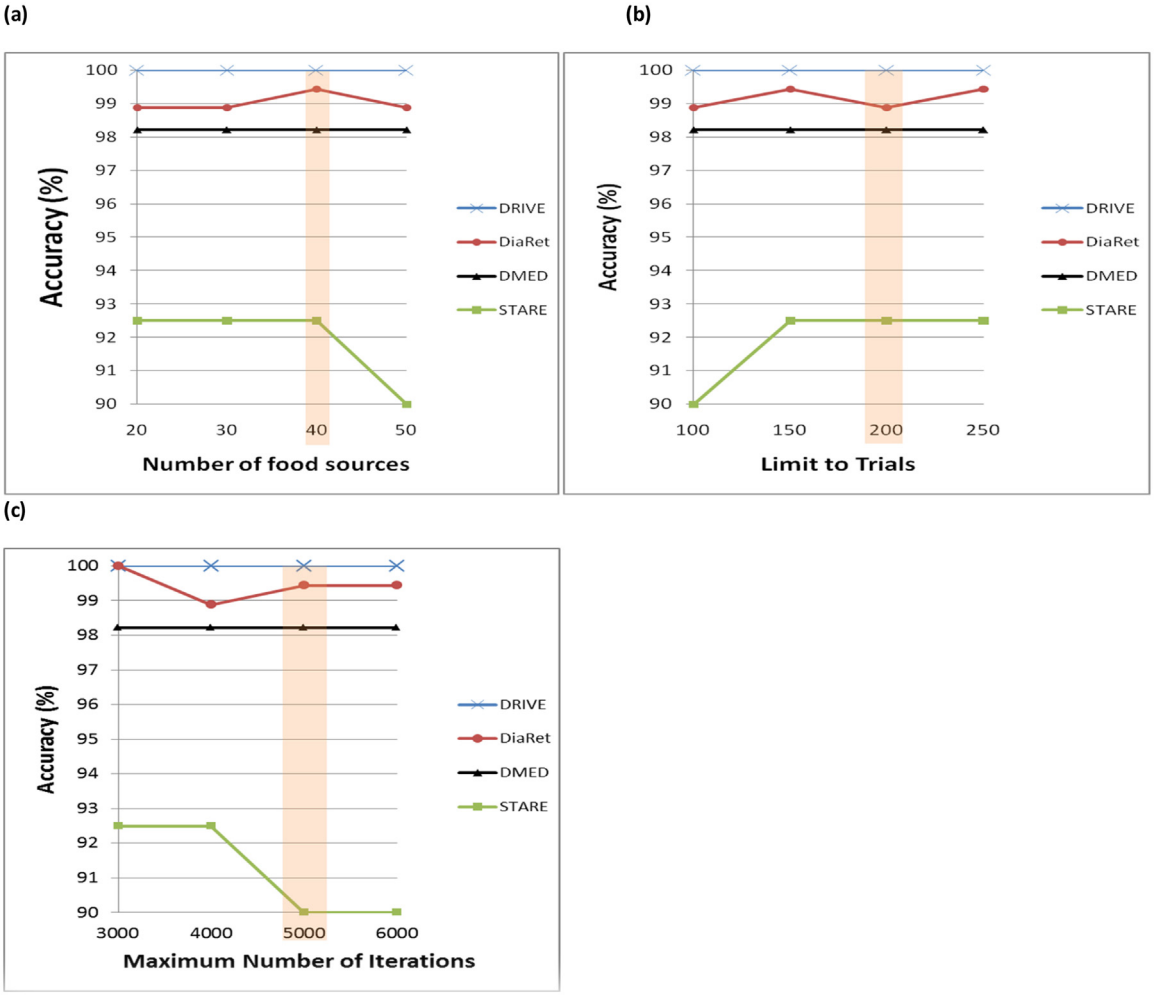

Fig. 10. Plots varying ABC parameters. Base parameters used for the plots are: Number of food sources = 40, Limit to trials = 200, and Maximum number of iterations = 5000. The highlighted semi-transparent red column represents the base parameters values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Although it is difficult to compare the running time of our work with other published work in the literature due to differences in the computational environment, the results discussed here provide a general idea of the obtained times. A running time of 3.52 s per image was reported for the firefly algorithm on the DiaRetDB1 dataset [11]. Ant colony optimization was used with a reported time of 75 s per image [12]. The running times of our work shown in Table 7 demonstrate some competitive numbers, particularly for the firefly and cuckoo search algorithms, which had a maximum of only 1.49 s across all datasets.

The firefly algorithm and ABC algorithm produced the same results in all runs and thus were the most consistent. The results of the cuckoo algorithm varied by one in two datasets but were consistent with those of the other algorithms for the other datasets. For the PSO and bat algorithms, the result varied by one in three databases. Overall, based on the four tested datasets, all five swarm intelligence algorithms were highly consistent. The above discussion is summarized in Table 8.

Overall, the firefly algorithm was the best in terms of detection rate, running time and consistency. Superior results were obtained using firefly in a previous study [11]. In the present work, the firefly algorithm had the highest accuracy and lowest running time among the five tested swarm techniques. In [11], the authors applied only single median filtering as a pre-processing technique; by contrast, the pre-processing in the present paper was performed with many levels of median filtering, background subtraction, mean filtering and masking. In addition, much more appropriate parameters were used for the firefly algorithm (Table 5). Hence, the results were improved compared to the work in [11].

Table 8
Ranking of the tested swarm intelligence algorithms in each performance category. Green indicates the best and red the worst.

Algorithm	Detection Rate Rank	Running Time Rank	Consistency Rank
Firefly Algorithm	1	1	1
Artificial Bee Colony	2	5	1
Cuckoo Search	2	2	2
Bat Algorithm	3	4	3
Particle Swarm Optimization	4	3	3

5.3. *Impact of swarm intelligence algorithm parameters on optic disc detection accuracy*

The parameters used for each swarm intelligence algorithm shown in Tables 1–5 represent what we found experimentally to give good results. In this subsection, we show the impact of varying these parameters on the optic disc detection accuracy.

The performance study is as follow; we conduct several experiments that vary one parameter while keeping others constant for

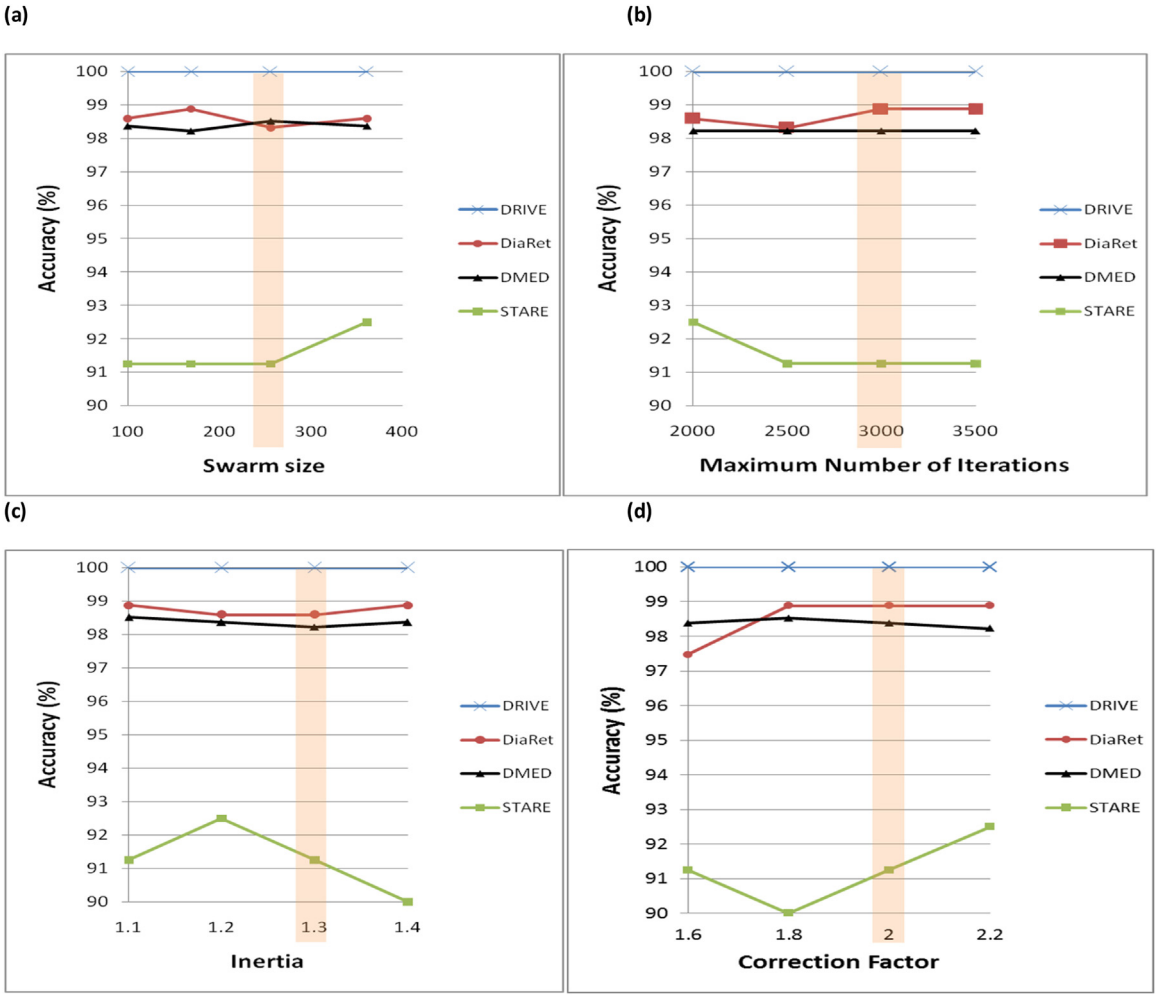

Fig. 11. Plots varying PSO parameters. The base parameters used are: swarm size = 256, maximum number of iterations = 3000, intertia = 1.3, and correction factor = 2.0. The highlighted semi-transparent red column represents the base parameters values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

all five swarm intelligence algorithm. The variation will be around the selected parameter value shown in Tables 1–5 (also called base parameter). The accuracy is calculated for all four datasets (DRIVE, DiaRetDB1, DMED and STARE) and a plot is produced for every parameter variation as shown in Figs. 10–14. The base parameter value in the plot is marked in semi-transparent red column.

An initial observation from these plots is that the variation parameters have different weight and sensitivity to the produced optic disc detection accuracy. For example, in Fig. 11(a), the variation in swarm size has a very negligible effect on the accuracy. On the other hand, a variation in β_0 parameter in firefly algorithm (Fig. 14(e)) has a noticeable effect in accuracy. In particular, the STARE accuracy in this experiment drops from 95% to 85% when the value of β_0 is increased beyond the base parameter value.

Let us consider each experiment separately. First, the ABC plot shown in Fig. 10 tests the variation of three parameters: number of food sources, limit to trials, and maximum number of iterations. These base parameters are given in Table 1. Only small variations are observed due to variation in the three parameters that can be assumed to be statistical variations. For PSO algorithm (Fig. 12), four parameters are varied: swarm size, *maxIter*, inertia and correction factor. Similar to ABC, here the variation in optic disc detection accuracy due to change in parameters are minimal and within statistical variation.

In the bat algorithm plot given in Fig. 12, six parameters were varied: population size, loudness, pulse rate, max number of iterations, *fmin*, and *fmax*. A significant improvement in performance is observed when the population size is increased from 100 to 300. Above 300, the improvement in performance is not observed. The algorithm appears to be very sensitive to the loudness parameter A, as significant deterioration in performance is observed when the parameter is increased and decreased.

Experiments conducted on cuckoo algorithm are shown in Fig. 13, which varies six parameters: number of nests, β, σ, *maxIter*, *pa*, and *a*. A significant deterioration in performance can be observed when the parameter *pa* is increased beyond 0.75. Parameter study of firefly algorithm is shown in Fig. 14. For this algorithm, the following parameters are studied: *MaxGeneration*, population size, Initial value of α, δ, β_0, and γ. There is a significant improvement in performance as the parameters population size and MaxGeneration are increased. The performance appears to be very sensitive to the parameters α, γ, and β_0 as performance is reduced when the parameters are varied from the chosen value. In addition, performance goes down when the value of δ is lower than the value used for taking results.

There is definitely potential in selecting better parameters for each algorithm beyond the base parameters selected in this research. The base parameters values for our research (see Tables 1–5) were selected on the bases of trial and error and

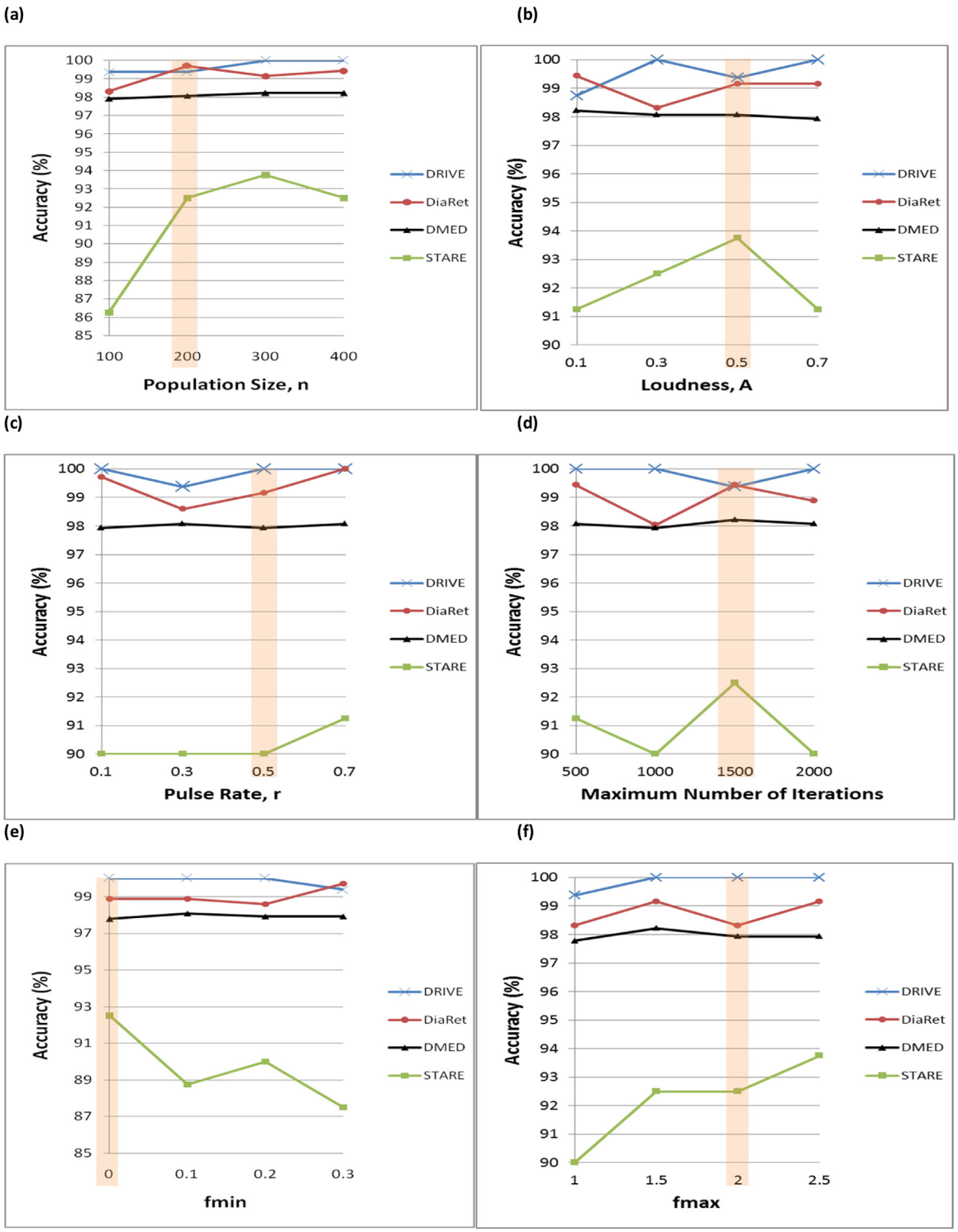

Fig. 12. Plots varying bat algorithm parameters. Base parameters used are: population size = 200, loudness = 0.5, pulse rate = 0.5, maximum number of iterations = 1500, fmin = 0, and fmax = 2. The highlighted semi-transparent red column represents the base parameters values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the accuracy produced was higher than what has been previously published in the literature (see Table 6). Finding optimal or semi optimal parameters is a convoluted problem that involves countless values with a possibility of varying results for each dataset which is beyond the scope of this research work and is to be tackled in future work.

6. Conclusion

This work further investigated the use of swarm intelligence techniques to detect the optic disc in fundus images. All five techniques were preceded by pre-processing involving many levels of median filtering, background subtraction, mean filtering and masking. The Background Subtraction based Optic Disc Detection

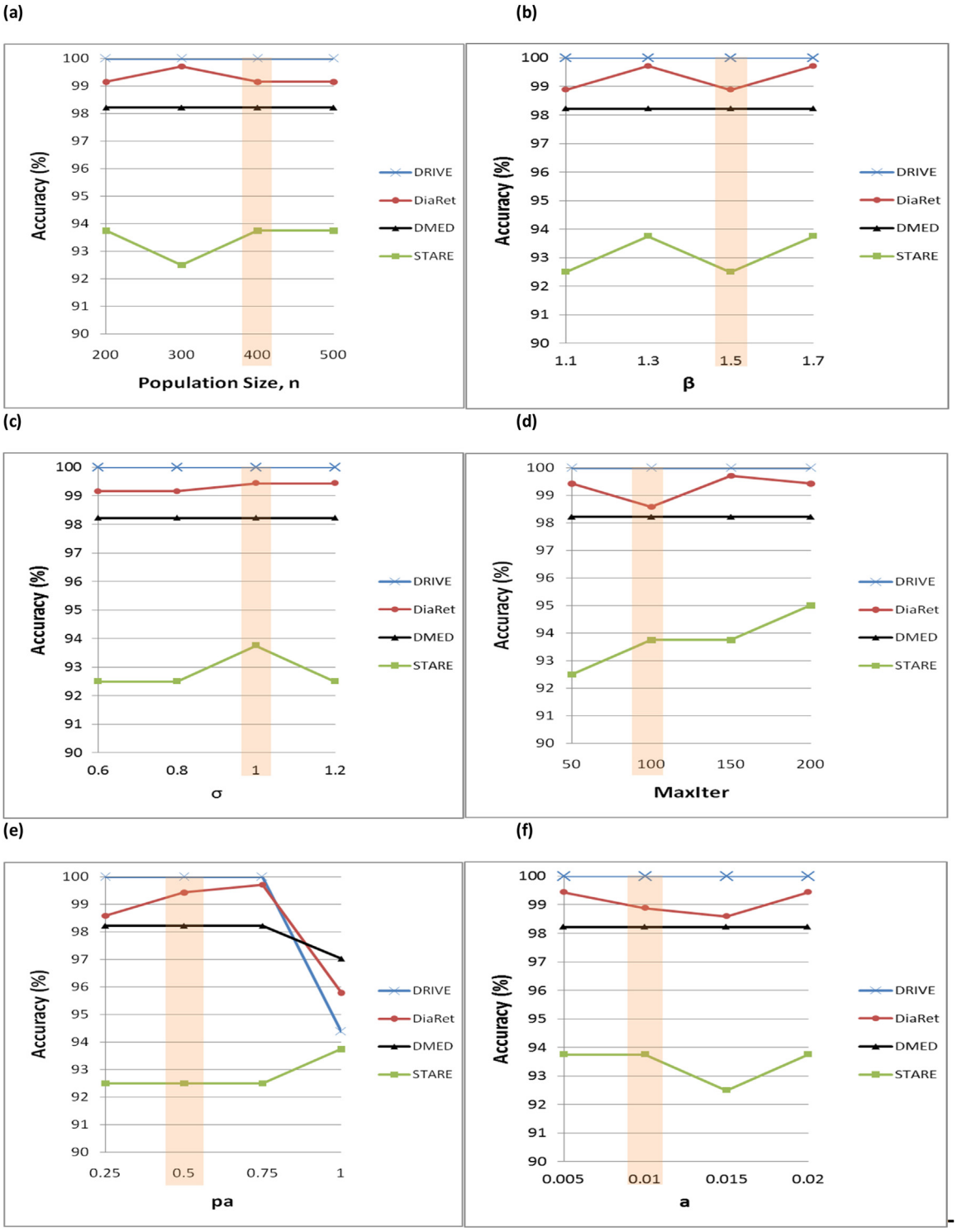

Fig. 13. Plots varying cuckoo algorithm parameters. Base parameters used are: population size = 400, $\beta = 3/2$, $\sigma = 1$, MaxIter = 100, pa = 0.5, and a = 0.01. The highlighted semi-transparent red column represents the base parameters values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

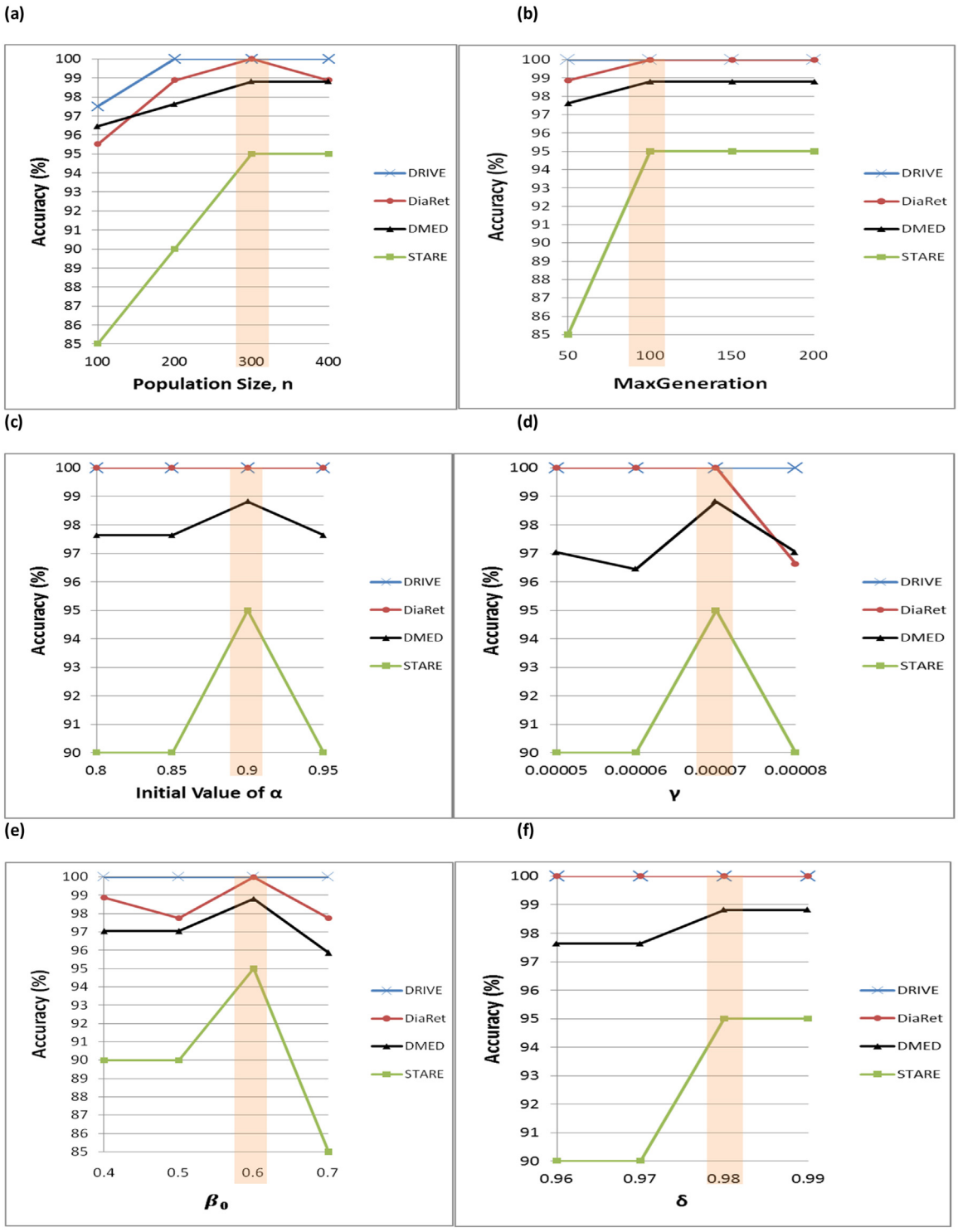

Fig. 14. Plots varying firefly algorithm parameters. Base parameters used are: population size = 300, MaxGeneration = 100, initial value of $\alpha = 0.9$, $\gamma = 0.00007$, $\beta_0 = 0.6$, and $\delta = 0.98$. The highlighted semi-transparent red column represents the base parameters values. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

algorithm was tested and evaluated. The experimental results confirmed the superiority of the swarm intelligence techniques to others in detecting intelligence. In particular, the firefly algorithm was experimentally demonstrated to have the highest accuracy and running time, followed closely by the cuckoo search algorithm. We conclude that with suitable pre-processing and parameters, swarm intelligence techniques are very effective in detecting the optic disc accurately and rapidly. A performance study on the effect of varying parameters of the nature inspired algorithms was also presented in this paper. Future work will include evaluating the performance of other swarm intelligence and other optic disc detection methods using the proposed pre-processing method.

Appendix A.

In this section, we have included a simplified pseudocode for each of the swarm intelligence algorithms used in this paper. This Appendix is added for illustration purposes.

A.1 Pseudocode of the ABC Algorithm

Initialize food sources $s_i. i = 1, \dots, n$ randomly and evaluate the objective function values at the points
Initialize $GlobalParams$ as the food source that gives lowest value of objective function and $BestInd$ as the corresponding objective function value
Initialize $trial$ as a vector of zeros of length $numFoodSources$
while $iter < maxIter$
  for each food source i
    Select the x or y coordinate at random for which the food source value is to be changed
    Select a neighbor food source n that is distinct from i
    Vary the selected dimension of i^{th} food source according to Eqn (A.1) and assign it to sol
    Calculate objective function value and fitness ft using Eqn (A.2)
    If $ft > ft_i$
      Assign value of sol to s_i and calculate objective function value and fitness ft using Eqn (A.2).
      Set value of $trial_i$ to zero
    else
    $trial_i = trial_i + 1$
    endif
  endfor
  Find the maximum of ft_i over all i and assign it to ft_{max}
  Find probability $p_i, i = 1, \dots, n$ using Eqn (A.3)
  foreach food source i
    Select *rand,* randomly selected number in the range $[0,1]$
    If $rand < p_i$
      Select at random, the dimension to be changed
      Select a neighbor food source n that is distinct from i
      Vary the selected dimension of i^{th} food source according to Equation A.1 and assign it to sol
      If $ft > ft_i$
        Assign value of sol to s_i and calculate objective function value and fitness ft_i using Eqn (A.2).
        Set value of $trial_i$ to zero
      else
    $trial_i = trial_i + 1$
      endif
    endif
    Find the food source s_{min} for which the objective function value is minimum
    If $f(s_{min}) < GlobalMin$
      Assign value of s_{min} to $GlobalParams$ and $f(s_{min})$ to $GlobalMin$
    endif
  endfor
  Find the value i for which $trial_i$ is maximum
  If $trial_i > limit$
  Randomly select the value of food source i and find its objective function and fitness function. Set value of $trial_i$ to zero
  endif
  Increment $iter$
endwhile
$GlobalParams$ holds the minimum found

Algorithm 1. Pseudocode of the ABC algorithm

The equations used in the ABC algorithm are given below:

$$sol_d = s_{id} + (s_{id} - s_{nd})\, rand \tag{A.1}$$

where sol_d is the d^{th} dimension of the vector sol, s_{id} is the d^{th} dimension of the i^{th} food source s_i, s_{nd} is the d^{th} dimension of the n^{th} food source, and *rand* is a random number in the interval [−1,1].

$$ft_i = \begin{cases} 1/(1+f(s_i)) & if \quad f(s_i) \geq 0 \\ 1+|f(s_i)| & if \quad f(s_i) < 0 \end{cases} \tag{A.2}$$

$$p_i = \left(0.9 \times ft_i / ft_{max}\right) + 0.1 \tag{A.3}$$

A.2 Pseudocode of the PSO Algorithm

The pseudocode of the PSO algorithm is shown in Algorithm 2.

foreach particle $i = 1, \ldots, S$
  Position of the particle x_i is initialized by sampling from a uniform random vector that is uniformly distributed in the interval between maximum and minimum values of the input, b_{up} and b_{lo} respectively
  The particle's best known positionp_i is initialized as the initial position x_i
  If$f(p_i) < f(g)$
    The best known position of swarm, g is updated asp_i
  endif
  Particle's velocityv_i is initialized by sampling from a uniform distribution in the interval $-|b_{up} - b_{lo}|$ and $|b_{up} - b_{lo}|$
endfor
for$N$ iterations
  foreach particle
    Pick d dimensional vectors of random numbers r_p and r_ω each element of which are uniformly distributed
    from 0 to 1. d is the number of dimensions if x which is 2 here
    Update velocity vector $v_i \leftarrow \omega r_{omega} v_i + \phi_p r_p \otimes (p_i - x_i) + \phi_p r_g \otimes (g_i - x_i)$
    Update position of the particle: $x_i \leftarrow x_i + v_i$
    If$f(x_i) < f(p_i))$
      Update the particle's best known position p_i to x_i
      If$f(p_i) < f(g)$
        Update the swarm's best known position$g$ to p_i
      endif
    endif
  endfor
endfor
The best positions is in g

Algorithm 2. Pseudocode of the PSO algorithm

In the pseudocode, the symbol $\otimes$ stands for element-wise multiplication between two vectors.

A.3 Pseudocode of the Bat Algorithm

Algorithm 3 gives the pseudocode of the bat algorithm.

Initialize Bat population x and velocity v_i
Initialize $best$ as the value of x_i that gives the minimum value of objective function
Initialize F_{best} as the minimum value found for the objective function
Define pulse frequency f_i as a sample uniformly distributed random variable between the maximum and minimum frequencies
Initialize the loudness A and pulse rates r
while iter < Maximum number of iterations
    for Each bat i
        Generate new solution by changing frequency f_i as a sample uniformly distributed random variable between the maximum and minimum frequencies, and updating locations and velocities using Eqns (A.4) and (A.5)
        If rand $> r$
            Generate a local candidate solution S_i around the best solution by adding to the best solution a Gaussian random variable with mean zero and standard deviation s
        endif
        Find the objective function value F_{new} for the new candidate S_i
        If F_{new} < Objective function value for the current i^{th} bat solution & rand $< A$
            Assign i^{th} solution candidate to the i^{th} solution
            Assign F_{new} to i^{th} fitness value
        endif
        if $F_{new} < F_{best}$
            Assign the i^{th} candidate solution to $best$
            Assign F_{new} to F_{best}
        endif
    endfor
    iter=iter+1
endwhile
$best$ holds the optimum value found

Algorithm 3. Pseudocode of the Bat algorithm

The update equations used in the algorithm are the following:

$$v_i^k = v_i^{k-1} + \left(x_i^k - x_*\right) f_i \tag{A.4}$$

$$x_i^k = x_i^{k-1} + v_i^k \tag{A.5}$$

In the above equation, β is a random number in the interval [0,1], and f_{max} and f_{min} are the maximum and the minimum frequencies, respectively. v_i^k and x_i^k are the velocity and location of the i^{th} bat in the k^{th} iteration, respectively.

A.4 Pseudocode of the cuckoo search

The pseudocode of the cuckoo search is given in Algorithm 4.

Randomly initialize the nest locations and evaluate the objective function values at those locations
Select the nest location that has the highest objective function value and assign it to $best$
for iter=1:MaxIter
  for each nest j
    Select $step1$ with Levy distribution with parameters β, σ_v
    Find $stepsize1$ using Eqn (A.6)
    Change the j^{th} nest location using Eqn (A.7) and assign it to $newnest$
    If objective function value for $newnest1$ is lower than that of j^{th} nest, assign $newnest1$ to j^{th} nest
  endfor
  for each nest j
    Select a random nest k that is distinct from j
    If $rand > pa$
      Find $stepsize2$ given by Eqn (A.8) and alter the location of the j^{th} nest using the step size and assign it to $newnest2$ as given by Eqn (A.9)
      If objective function value for $newnest2$ is lower than that of j^{th} nest, assign $newnest2$ to j^{th} nest
    endif
  endfor
  Find the nest that gives the minimum value of the objective function and assign it to $best$
endfor
$best$ holds the optimum that is obtained

Algorithm 4. Pseudocode of the Cuckoo search algorithm

The update equations used in the cuckoo search are given below:

$$stepsize = a \times step \times \left(s_j - best\right) \quad (A.6)$$

$$newnest = s_j + stepsize \times randn \quad (A.7)$$

$$stepsize2 = rand \times \left(s_j - s_k\right) \quad (A.8)$$

$$newnest2 = s_j + stepsize2 \quad (A.9)$$

where s_j is the j^{th} nest, *randn* is a number with a Gaussian distribution with zero mean and unit variance and *rand* is a random number in the interval [0,1].

A.5 Pseudocode of the Firefly algorithm

The pseudocode of the firefly algorithm is given in Algorithm 5.

Get the parameters of the algorithm: α, γ, β_0,δ
Randomly initialize the locations of fireflies and assign them to the vector x_s
for generation=1:MaxGeneration
  Evaluate the values of objective functions for all locations of fireflies
  Sort the firefly locations in the descending order of objective function values - sorted locations are given by x_o
  Find the firefly with the lowest value of objective function x_{best}
  for i=1:N
    for j=1:N
      Find r, the distance between x_{oi} and x_{sj}
      If $f(x_{sj}) < f(x_{oi})$
        Compute β using Eqn (A.10)
        Update firefly location x_{sj} using Eqn (A.11)
      endif
    endfor
  endfor
  Update α using Eqn (A.12)
endfor
Find the j for which $f(x_{sj})$ is minimum and assign that value to $best$
$best$ holds the minimum found

Algorithm 5. Pseudocode of the Firefly algorithm

The equations used in the firefly algorithm are given below:

$$\beta = \left(f\left(x_{oi}\right)/255\right)\beta_0 e^{-\gamma r^2} \quad \text{(A.10)}$$

$$x_{sj}^{t} = x_{sj}^{t-1}\left(1-\beta\right) + x_{best}\beta + a\alpha\left(p-0.5\right) \quad \text{(A.11)}$$

$$\alpha^{t} = \alpha^{t-1}\delta \quad \text{(A.12)}$$

where p is a random number in the interval [0,1] and α^t is the value of the parameter α in the t^{th} iteration.

References

[1] S. Sivaprasad, B. Gupta, R. Crosby-Nwaobi, J. Evans, Prevalence of diabetic retinopathy in various ethnic groups: a worldwide perspective, Surv. Ophthalmol. 57 (4) (2012) 347–370.

[2] X. Zhang, J.B. Saaddine, C. Chou, M.F. Cotch, Y.J. Cheng, L.S. Geiss, E.W. Gregg, A.L. Albright, B.E.K. Klein, R. Klein, Prevalence of diabetic retinopathy in the United States 2005–2008, JAMA 304 (6) (2010) 649–656.

[3] K. Ganesan, R.J. Martis, U.R. Acharya, C.K. Chua, L.C. Min, E. Ng, A. Laude, Computer-aided diabetic retinopathy detection using trace transforms on digital fundus images, Med. Biol. Eng. Comput. 52 (8) (2014) 663–672.

[4] M.R.K. Mookiah, U.R. Acharya, V. Chandran, R.J. Martis, J.H. Tan, J.E. Koh, C.K. Chua, L. Tong, A. Laude, Application of higher-order spectra for automated grading of diabetic maculopathy, Med. Biol. Eng. Comput. 53 (12) (2015) 1319–1331.

[5] S. Ibrahim, P. Chowriappa, S. Dua, U.R. Acharya, K. Noronha, S. Bhandary, H. Mugasa, Classification of diabetes maculopathy images using data-adaptive neuro-fuzzy inference classifier, Med. Biol. Eng. Comput. 53 (12) (2015) 1345–1360.

[6] A. Hoover, M. Goldbaum, Locating the optic nerve in a retinal image using the fuzzy convergence of the blood vessels, IEEE Trans. Med. Imaging 22 (8) (2003) 951–958.

[7] M. Foracchia, E. Grisan, A. Ruggeri, Detection of optic disc in retinal images by means of a geometrical model of vessel structure, IEEE Trans. Med. Imaging 23 (10) (2004) 1189–1195.

[8] A.A.-H.A.-R. Youssif, A.Z. Ghalwash, A.A.S.A.-R. Ghoneim, Optic disc detection from normalized digital fundus images by means of a vessels' direction matched filter, IEEE Trans. Med. Imaging 27 (1) (2008) 11–18.

[9] A.D. Fleming, K.A. Goatman, S. Philip, J.A. Olson, P.F. Sharp, Automatic detection of retinal anatomy to assist diabetic retinopathy screening, Phys. Med. Biol. 52 (2) (2007) 331.

[10] X. Zhu, R.M. Rangayyan, A.L. Ells, Detection of the optic nerve head in fundus images of the retina using the Hough transform for circles, J. Digit. Imaging 23 (3) (2010) 332–341.

[11] J. Rahebi, F. Hardalac, A new approach to optic disc detection in human retinal images using the firefly algorithm, Med. Biol. Eng. Comput. 45 (2) (2016) 453–461.

[12] C. Pereira, L. Goncalves, M. Ferreira, Optic disc detection in color fundus images using ant colony optimization, Med. Biol. Eng. Comput. 51 (3) (2013) 295–303.

[13] C. Sinthanayothin, J.F. Boyce, H.L. Cook, T.H. Williamson, Automated localization of the optic disc, fovea, and retinal blood vessels from digital colour fundus images, Br. J. Ophthalmol. 83 (8) (1999) 902–910.

[14] T. Walter, J.-C. Klein, P. Massin, A. Erginay, A contribution of image processing to the diagnosis of diabetic retinopathy-detection of exudates in color fundus images of the human retina, IEEE Trans. Med. Imaging 21 (10) (2002) 1236–1243.

[15] R.J. Qureshi, L. Kovacs, B. Harangi, B. Nagy, T. Peto, A. Hajdu, Combining algorithms for automatic detection of optic disc and macula in fundus images, Comput. Vision Image Understanding 116 (1) (2012) 138–145.

[16] H.-K. Hsiao, C.-C. Liu, C.-Y. Yu, S.-W. Kuo, S.-S. Yu, A novel optic disc detection scheme on retinal images, Expert Syst. Appl. 39 (12) (2012) 10600–10606.

[17] A. Suero, D. Marin, M.E. Gegíundez-Arias, J.M. Bravo, Locating the optic disc in retinal images using morphological techniques, IWBBIO (2013) 593–600.

[18] E. Cuevas, H. Sossa, et al., A comparison of nature inspired algorithms for multi-threshold image segmentation, Expert Syst. Appl. 40 (4) (2013) 1213–1219.

[19] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, J. Global Optim. 39 (3) (2007) 459–471.

[20] J. Kennedy, Particle swarm optimization, in: Encyclopediaof Machine Learning, Springer, 2010, pp. 760–766.

[21] X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, 2010, pp. 65–74.

[22] R. Rajabioun, Cuckoo optimization algorithm, Appl. Soft Comput. 11 (8) (2011) 5508–5518.

[23] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimization, International Journal of Bio-Inspire, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84.

[24] K. Zuiderveld, Contrast limited adaptive histogram equalization, in: Paul S. Heckbert (Ed.), Graphic Gems IV, Academic Press Professional, Inc., San Diego, CA, USA, 1994, pp. 474–485.

[25] J. Staal, M.D. Abr'amoff, M. Niemeijer, M. Viergever, B. Van Ginneken, Ridge-based vessel segmentation in color images of the retina, IEEE Trans. Med. Imaging 23 (4) (2004) 501–509.

[26] T. Kauppi, V. Kalesnykiene, J. k.Kamarainen, L. Lensu, I. Sorri, A. Raninen, R. Voutilainen, H. Uusitalo, H. Kälviäinen, J. Pietilä, Diaretdb1 diabetic retinopathy database and evaluation protocol, in: British Machine Vision Conference—BMVC, University of Warwick, 2007, pp. 252–261.

[27] L. Giancardo, F. Meriaudeau, T.P. Karnowski, Y. Li, S. Garg, K.W. Tobin, E. Chaum, Exudate-based diabetic macular edema detection in fundus images using publicly available datasets, Med. Image Anal. 16 (1) (2012) 216–226.

[28] A. Hoover, V. Kouznetsova, M. Goldbaum, Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response, IEEE Trans. Med. Imaging 19 (3) (2000) 203–210.