Northumbria Research Link

This version was downloaded from Northumbria Research Link: http://nrl.northumbria.ac.uk/id/eprint/30210/

Northumbria
University
NEWCASTLE

University**Library**

Applied
Soft
Computing

Intelligent leukaemia diagnosis with bare-bones PSO based feature optimization

Worawut Srisukkham [a], Li Zhang [a,*], Siew Chin Neoh [b], Stephen Todryk [c], Chee Peng Lim [d]

[a] Computational Intelligence Research Group, Department of Computing Science and Digital Technologies, Faculty of Engineering and Environment, University of Northumbria, Newcastle, NE1 8ST, UK
[b] Faculty of Engineering, Technology and Built Environment, UCSI University, Malaysia
[c] Department of Applied Sciences, Faculty of Health and Life Sciences, University of Northumbria, Newcastle, NE1 8ST, UK
[d] Institute for Intelligent Systems Research and Innovation, Deakin University, Waurn Ponds, VIC 3216, Australia

ARTICLE INFO

ABSTRACT

In this research, we propose an intelligent decision support system for acute lymphoblastic leukaemia (ALL) diagnosis using microscopic images. Two Bare-bones Particle Swarm Optimization (BBPSO) algorithms are proposed to identify the most significant discriminative characteristics of healthy and blast cells to enable efficient ALL classification. The first BBPSO variant incorporates accelerated chaotic search mechanisms of food chasing and enemy avoidance to diversify the search and mitigate the premature convergence of the original BBPSO algorithm. The second BBPSO variant exhibits both of the abovementioned new search mechanisms in a subswarm-based search. Evaluated with the ALL-IDB2 database, both proposed algorithms achieve superior geometric mean performances of 94.94% and 96.25%, respectively, and outperform other metaheuristic search and related methods significantly for ALL classification.

1. Introduction

Leukaemia is a type of cancer pertaining to white blood cells (WBCs), in which abnormal and immature WBCs are produced by the bone marrow and enter the bloodstream. There are two types of acute leukaemia, i.e. acute lymphoblastic leukaemia (ALL) and acute myeloid leukaemia (AML). Since ALL diagnosis associates closely with morphological changes of WBCs and manual morphological analysis may suffer from several potential limitations (e.g. non-standard precision and relying heavily on medical professionals' knowledge and skill) [1–3], many automatic ALL diagnosis methods have been proposed in recent years [1–5]. In order to achieve robust and efficient computerized diagnosis, identifying the characteristics of healthy and blast cells is a crucial factor. Although many studies on the separation and retrieval of the nucleus and cytoplasm or purely nuclei of the cells using segmentation techniques are available, limited investigations have been conducted on the selection of significant discriminative character-

istics from the segmented regions to effectively benefit subsequent ALL diagnosis [2–7].

This research aims to deal with the aforementioned challenges by proposing an intelligent decision support system with evolutionary feature optimization for robust ALL classification. Specifically, we propose two Bare-bones Particle Swarm Optimization (BBPSO) algorithms to extract the most significant discriminative characteristics of normal and abnormal lymphocytic cells for ALL classification. The proposed BBPSO variants incorporate accelerated search mechanisms of attraction to the food source and avoidance of enemies to diversify the search and overcome premature convergence of the original BBPSO algorithm.

Fig. 1 shows the overall flow of the proposed system. It contains the following key steps: (a) WBC identification from blood smear images, (b) nucleus-cytoplasm separation, (c) feature extraction, (d) BBPSO-based feature optimization, and (e) lymphocyte and lymphoblast identification. After employing marker-controlled watershed segmentation to extract WBCs from microscopic images, a stimulating discriminant measure (SDM)-based clustering algorithm proposed in our previous research [2] is used for nucleus-cytoplasm separation. We initially extract 80 raw features from the segmented nucleus and cytoplasm sub-images. The proposed BBPSO algorithms are then used to identify the most significant discriminative characteristics of healthy and blast cells from the extracted raw features, respectively. Both 1-Nearest

* Corresponding author.
E-mail addresses: worawut@northumbria.ac.uk (W. Srisukkham), li.zhang@northumbria.ac.uk (L. Zhang), u_jane80@yahoo.co.uk (S.C. Neoh), stephen.todryk@northumbria.ac.uk (S. Todryk), chee.lim@deakin.edu.au (C.P. Lim).
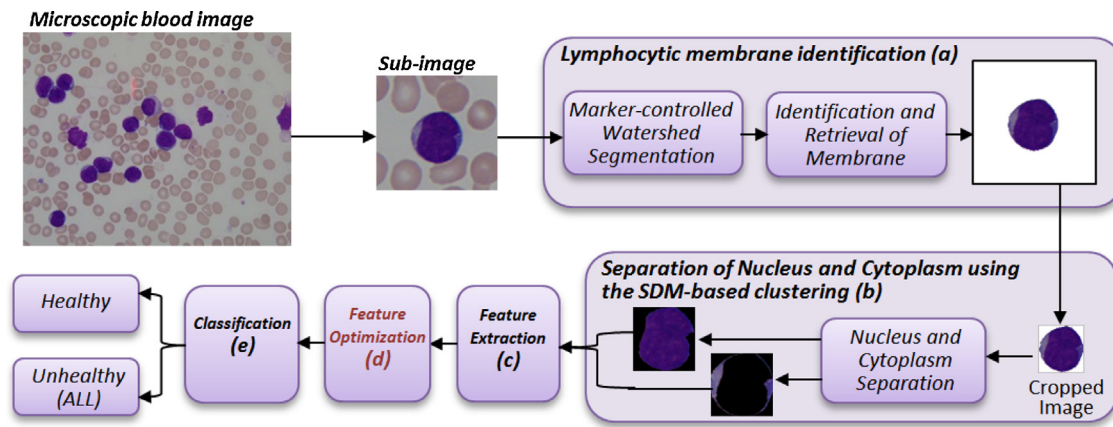
Fig. 1. The system architecture (this research focuses on (*d*) feature optimization).

Neighbour (1NN) and Support Vector Machine (SVM) with Gaussian Radial Basis Function (RBF) kernel are used to classify lymphocytes and lymphoblasts using the identified optimal feature subsets.

The contributions of this research are summarized, as follows.

1. We propose two BBPSO algorithms for feature optimization. Besides the original position updating operation of the BBPSO algorithm, the proposed variants incorporate mechanisms of attraction to the food source and avoidance of enemies to increase search diversity and overcome local optima of the original BBPSO algorithm. These two new behaviours are also accelerated by the Logistic chaotic map.

2. The food chasing behaviour is guided by the average personal best experience and the global best solution to enable the search to reach attractive optimal regions more effectively. The mechanism of fleeing from enemies enables the particles to move away from unpromising search regions, in order to accelerate convergence. The first proposed BBPSO variant incorporates these two new search behaviours to guide the search in the main swarm, while the second proposed variant exhibits these strategies in the subswarm-based search. These two search mechanisms and the original BBPSO operation work in a cooperative manner to lead the search to attain global optima.

3. In comparison with other metaheuristic search methods, the proposed BBPSO algorithms possess efficient discriminative capabilities in which significant discriminating features for lymphocytes and lymphoblasts are revealed. Evaluated with 180 microscopic images extracted from the ALL-IDB2 database [3], the proposed algorithms show great efficiency, and outperform other search methods across different experimental settings under different fitness evaluations. They also compare favourably with other related methods for ALL diagnosis reported in the literature.

The organisation of this paper is as follows. Section 2 introduces the related research on automatic ALL diagnosis and feature optimization techniques. Section 3 presents the proposed ALL diagnosis system including its pre-processing steps, the proposed BBPSO-based feature optimization methods and ALL identification using both 1NN and SVM. Evaluation of the proposed algorithms and comparison with other search methods using the ALL-IDB2 database are discussed in Section 4. Finally, we draw conclusions and identify future research directions in Section 5.

2. Related work

In this section, we discuss related research on automatic ALL diagnosis and state-of-the-art feature optimization techniques.

2.1. Automatic leukaemia diagnosis

Defined by the French-American-British classification systems [1,4], there are three subtypes of ALL, i.e. L1 to L3 and eight subtypes of AML, i.e. M0 to M7. Many research studies have been dedicated to the automatic diagnosis of ALL, AML and their subtypes in order to promote early diagnosis. Neoh et al. [2] proposed an automatic ALL diagnosis system using microscopic blood images. Their work proposed a clustering algorithm with the stimulating discriminant measure (i.e. SDM) that took both within and between cluster scatter variances into account for nucleus-cytoplasm separation. The SDM-based clustering algorithm was integrated with the Genetic Algorithm (GA) to perform robust segmentation of nucleus, cytoplasm and background regions. Single and ensemble classifiers were applied in their work for ALL recognition. Bootstrapping and 10-fold cross validation were used for system evaluation. Shadowed C-means (SCM) clustering was used by Mohapatra et al. [5] to perform lymphocyte image segmentation. It clustered each pixel into one of the three regions, i.e. cytoplasm, nucleus and background. An ensemble classifier, consisting of a neural network (NN), SVM, and k-Nearest Neighbour (kNN), was used to recognize lymphocytes and lymphoblasts. The ensemble classifier outperformed other single models including the NN, kNN, Naïve Bayes Classifier (NB), SVM, and Radial Basis Function Network (RBFN). Furthermore, Madhloom et al. [6] integrated colour features with morphological reconstruction to localize and isolate blast cells. They also applied the Fisher Discriminant Ratio (FDR) to rank and select features from each cell for subsequent ALL recognition using kNN. Their work employed 260 cell images with 180 and 80 images for training and test respectively, and achieved 92.5% accuracy for the distinction of healthy and blast cells.

Agaian et al. [4] proposed an AML detection system which employed K-means Clustering and morphological filtering to segment nuclei from leucocytes. Local Binary Patterns (LBP) and Hausdorff Dimension (HD) were used in their work to extract useful features in addition to other extracted shape, Gray Level Co-occurrence Matrix (GLCM) and colour features. The SVM classifier was used to recognize AML and healthy cells. Meera and Matthew [7] introduced Fuzzy Local Information C-means for AML image segmentation with multiple nuclei, and used both GLCM and HD features with the SVM for identification of healthy and blast cells.

2.2. Feature extraction techniques

Feature extraction is an important step in contributing to accurate recognition of normal and blast cells. Features that are commonly extracted from the microscopic blood cell images include shape, colour, texture and statistical based information. Generally, shape-based features cover the geometric information such as area, perimeter, elongation, and eccentricity, while colour-based features include the type of colour space information such as RGB, CIELAB (CIE L*a*b*), or Hue-Saturation-Intensity (HSI). For textural features, GLCM, which provides information such as homogeneity, contrast, and entropy, is usually employed, whereas for statistical-based features, information such as mean and standard deviation is often used. Ongun et al. [8] adopted affine invariants, CIE L*a*b* colour space, colour histogram, and shape-based features from heuristic reasoning of haematologist to make up a total of 57 features for the classification of 12 types of blood cells (e.g. monocyte, neutrophil, myelocyte, plasma, etc). Putzu et al. [1] focused on the detection of abnormality in lymphocytes. A total of 30 shape, 21 colour and 80 GLCM-based texture descriptors were extracted from the obtained sub-images. Besides the GLCM textural features, some researchers employed different methods to interpret textural information from the cell images. As an example, LBP textural extraction was proposed by Singhal and Singh [9] for the detection of lymphocytes and lymphoblasts while Rezatofighi and Soltanian-Zadeh [10] used the LBP features for WBC extraction. In addition, HD was adopted by Mohapatra et al. [5] to extract roughness of the nucleus boundary of lymphocytes and lymphoblasts. The influence of the LBP operator on HD has also been evaluated by Agaian et al. [4]. Their experiments indicated the positive impact of LBP on HD, which boosted the AML classification performance greatly. A colour feature called cell energy was also employed in their work, which played a very important role in distinguishing between normal and abnormal cells.

2.3. Feature selection algorithms

The retrieval of shape, colour, texture and statistical based information from the blood cells often entails a large set of input features for the classification system, which could be computationally costly. While inadequate features reduce classification accuracy, a large feature set that involves redundant and insignificant information can reduce classification performance as well [11–13]. Therefore, optimal feature selection is crucial for the improvement of classification results. In this section, we first introduce well-known evolutionary optimization algorithms for feature selection, including Particle Swarm Optimization (PSO), Cuckoo Search (CS) and Dragonfly Algorithm (DA), followed by other advanced modified optimization mechanisms. Techniques dedicated to feature selection and dimension reduction for leukaemia classification are also introduced.

2.3.1. PSO and bare-bones PSO

A number of classical, well-known optimization algorithms have been applied to diverse engineering optimization tasks. Introduced by Kennedy and Eberhart [14], PSO is an efficient technique for feature selection [15,16]. In PSO, each particle has a position in the search space. The particle is characterised by a position vector, $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, and a velocity vector, $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, where D denotes the dimension of the search space. All particles move in the search space to search for the optimal solutions. In PSO, the best position ever achieved by a particle, i.e. the personal best, *pbest,* and the best position of the overall swarm, i.e. the global best, *gbest,* are used to update the velocity and position of each particle.

BBPSO is a variant of PSO [17]. Compared with PSO, it does not consider the velocity, but only updates the particles' positions. The Gaussian distribution is employed for position updating in BBPSO, as in Eq. (1).

$$x_{id}^{t+1} = \phi\left(\frac{pbest_{id}^t + gbest_d^t}{2}, |pbest_{id}^t - gbest_d^t|\right) \tag{1}$$

where ϕ denotes the Gaussian distribution, $\frac{pbest_{id}^t + gbest_d^t}{2}$ represents the mean or expectation of the distribution with $|pbest_{id}^t - gbest_d^t|$ as the standard deviation. Using Eq. (1), the new position of a particle is distributed according to the Gaussian distribution, although other distribution functions can also be applied. Compared with conventional PSO, BBPSO does not require any operating parameters. Therefore, BBPSO is more efficient, which has been extensively applied to real-world single and multi-objective optimization problems [18,19].

2.3.2. Cuckoo search

Proposed by Yang and Deb [20], Cuckoo search (CS) possesses both local and global search mechanisms to attain global convergence. CS employs the following three main principles for searching the global optimal solutions. Firstly, each cuckoo lays one egg (solution) at a time, which is discarded in a randomly chosen nest. Secondly, the best nests with high-quality eggs are selected for the next generation. Thirdly, the host bird discovers the egg laid by a cuckoo with a probability, p_a, therefore, a fraction (p_a) of the worse nests is abandoned and replaced by the new nests. The algorithm employs the following strategy to generate new nests (solutions).

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \varepsilon) \otimes \left(x_k^t - x_l^t\right) \tag{2}$$

where x_k^t and x_l^t denote the solutions selected randomly by random permutation, while s denotes the step size and $H(\nu)$ represents a Heaviside function. Note that ε is a random number drawn from a uniform distribution, while $\otimes$ represents the entry-wise product of two vectors. The new solution, x_i^{t+1}, is accepted if it has a better fitness value than that of x_i^t.

In each iteration, the Levy flights operation defined in Eq. (3) is used to perform the global random walk.

$$x_i^{t+1} = x_i^t + \alpha \times \psi(\lambda) \tag{3}$$

where x_i^{t+1} and x_i^t denote the ith solution in $t+1$-th and t-th generations, respectively, while ψ represents the Levy flights operation with λ as the random step length ($1 < \lambda \leq 3$), and α is the step-size scaling factor. CS employs these local and global search operations to search for the global optima.

Although CS shows impressive search capabilities, its search strategy could be further enhanced. For instance, the new nest (solution) generation strategy shown in Eq. (2) relies purely on two randomly selected individuals, x_k^t and x_l^t, and it does not explicitly employ optimal solutions identified so far for promising offspring generation. Therefore, it could be further improved by considering more explicit optimal signals, i.e. local and global best experiences, to increase the likelihood of generation of promising offspring solutions. Motivated by this perspective, we incorporate both personal and global best experiences to guide the attraction search mechanism and enable fast convergence in this research.

2.3.3. Dragonfly algorithm

Proposed by Mirjalili [21], DA simulates and implements static and dynamic swarming behaviours of dragonflies to balance between global exploration and local exploitation. It employs the following five social interaction behaviours, i.e. separation, alignment, cohesion, attraction (towards food), and distraction (outwards enemies) to guide the search process. Its velocity and

position updating operations are defined in Eqs. (4) and (5), respectively.

$$\Delta x_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta x_t \qquad (4)$$

$$x_{t+1} = x_t + \Delta x_{t+1} \qquad (5)$$

In Eq. (4), S_i, A_i, C_i, F_i, and E_i represent the social behaviours of separation, alignment, cohesion, attraction, and distraction, respectively while s, a, c, f, and e are the corresponding weights for the five actions. In addition, Δx_{t+1} and Δx_t represent the step/velocity vector in the $t+1$-th and t-th iterations, respectively, with w as the inertia weight. In Eq. (5), x_{t+1} and x_t indicate the positions of an individual in the $t+1$-th and t-th iterations, respectively. Eqs. (4) and (5) model the social behaviours of an artificial dragonfly when it has at least one neighbouring individual.

Among the five social behaviours, the distraction (i.e. evading) operation distinguishes DA from other swarm intelligence algorithms, and is defined as follows.

$$E_i = x^\varepsilon + x \qquad (6)$$

where E_i denotes the distraction action and x^ε represents the position of an enemy with x denoting the position of the current individual. Since this evading operation moves the current individual away from each solution with a lower fitness value, the search process is very likely to be computationally inefficient. In this research, we propose an evading mechanism that is guided by more explicit enemy signals, i.e. average personal historical and global worst experiences, for enemy avoidance. The rationale is to ensure each particle flees away from local and global unpromising search regions effectively, therefore accelerating convergence.

2.3.4. Other feature optimization techniques

There are also other modified or hybrid metaheuristic search algorithms proposed in recent years to overcome the limitations of some existing methods. Since a constant setting of operation parameters in CS may have a negative impact on its performance, Valian et al. [22] and Li and Yin [23] developed modified CS with self-adaptive parameter settings to overcome the problem associated with constant CS parameter settings. Valian et al. [22] adjusted the search parameters according to the number of generations, and recommended a comparatively larger parameter setting at the beginning to increase solution diversity and a smaller parameter setting in the later iterations to fine-tune the identified solutions. The work achieved impressive performance in complex engineering optimisation problems. Li and Yin [23] proposed two new local search strategies for CS. Based on a decreasing probability rule, the strategies aimed to balance between exploitation and exploration, and an adaptive parameter setting was introduced to enhance population diversity. Their work compared favourably with other related research based on a study with 16 benchmark functions. Jordehi [24] proposed an enhanced leader PSO (ELPSO), which employed successive mutation strategies such as Gaussian, Cauchy, opposition-based and differential evolution (DE) based mutation, to further enhance the swarm leader. The results indicated its efficiency in terms of accuracy and scalability. Zhang et al. [12] proposed a binary BBPSO-based feature selection algorithm. Their work used a reinforced memory strategy for personal best updating of each particle to retain particle diversity. It also used a uniform combination to diversify the swarm when stagnation occurred. The effects of uniform combination were strengthened along with the increase of stagnant iterations. The binary BBPSO algorithm showed a competitive performance in terms of classification accuracy and convergence rate. Neoh et al. [25] proposed two evolutionary algorithms under a layered cascade evolutionary framework, i.e. direct similarity and Pareto-based feature selection, for facial expression recognition. The direct similarity feature selection algorithm integrated the concept of micro Genetic Algorithm and focused on identifying common features within each class. Meanwhile, the Pareto-based optimization took both between-class and within-class variations into account for multi-objective feature optimization. Both optimization strategies achieved impressive performances and outperformed other baseline methods (e.g. GA and AdaBoosting) significantly. A comprehensive review of PSO and its applications has also been conducted by Zhang et al. [15].

There was also other research dedicated to dimension reduction for automatic leukaemia diagnosis. In order to select the input features for effective classification of normal and abnormal lymphocytes, Mohapatra et al. [5] applied an independent-sample "t" test to select 32 statistically significant features out of 44 raw features, representing shape, colour and texture information of nucleus and cytoplasm. Madhloom et al. [6] selected 7 out of 30 raw features, representing shape, colour and texture information of nucleus, cytoplasm and the whole cell, by employing FDR that considered cross-correlation among features for the identification of lymphocytes and lymphoblasts. In addition, Rezatofighi and Soltanian-Zadeh [10] further proposed sequential forward selection along with FDR for the recognition of five types of WBCs. Huang and Hung [26] used the Principal Component Analysis (PCA) to reduce the feature dimensions from 85 to 7 in leucocyte recognition. Despite the popularity of the filter-based approach, Osowski et al. [27] proposed an embedded method to recognize 11 types of blood cells (e.g. basophilic erythroblast, neutrophilic myelocyte, lymphocyte, etc) by using the GA to fine-tune the features with respect to the SVM performance during the training stage. Escalante et al. [28] employed PSO to guide the search process and automatically select ensemble classification models for different types of leukaemia detection. Their work achieved high accuracy for classification model selection without user intervention. The system achieved 97.68% for ALL and AML leukaemia detection, and 94.21% for subtypes of ALL (L1 and L2) and AML (M2, M3 and M5) identification. Besides the detection of ALL and AML, the GA was employed by Chan et al. [29] to obtain optimal feature parameter values for recognition of anaemia abnormal red blood cells.

3. The proposed all recognition system

There are five key steps of our proposed system: (a) WBC identification from blood smear images, (b) nucleus and cytoplasm separation, (c) feature extraction, (d) modified BBPSO-based feature optimization, and (e) lymphocyte and lymphoblast classification. First of all, modified marker-controlled watershed segmentation and morphological operations proposed in our previous research [30] are used to extract WBCs from microscopic images automatically. Then, an SDM-based clustering algorithm utilising both within- and between-cluster scatter variances as proposed in our recent work [2] is used to perform nucleus-cytoplasm separation. A set of 80 raw features comprising 16 shape descriptors, 54 GLCM textural descriptors, and 10 CIELAB colour descriptors is extracted from the segmented nucleus and cytoplasm sub-images. These 16 extracted shape features include information with respect to the cell size, nucleus size, nucleus shape, and details of cytoplasm, which consists of cytoplasm and nucleus areas, nucleus to cytoplasm ratio, length to diameter ratio, major axis length, orientation, filled area, perimeter, solidity, eccentricity, minor axis length, convex area, form factor, compactness based on Mohapatra et al. [5], another compactness measure based on Mohapatra et al. [31], and roundness of the nucleus region. The 54 texture features consist of 13 descriptors from the GLCM matrix, including correlation, sum of variance, normalized inverse difference moment, sum of average, contrast, difference variance, entropy, cluster prominence, cluster shade, dissimilarity, energy,

homogeneity, and normalized inverse difference, computed in four different angles (i.e. 0, 45, 90, and 135) plus two additional descriptors, i.e. skewness and kurtosis. In addition, the 10 colour features consist of the mean and standard deviations of the a* and b* components of the CIELAB colour space for both nucleus and cytoplasm, along with two descriptors pertaining to the ratio of the mean of a* and b* components between cytoplasm and nucleus. Since identifying the most discriminative characteristics of normal and abnormal lymphocytic cells and removing the redundant features have a great impact in boosting classification accuracy, in this research, we propose two modified BBPSO algorithms to identify significant discriminative feature subsets of healthy and blast cells from the 80 raw features to benefit subsequent robust ALL classification. We introduce the proposed BBPSO algorithms in detail in the following section.

provides chaotic adaptive steps of the search behaviour. In Eq. (7), each particle is guided by the mean of $pbest'_{id}$ and $gbest^t_d$ to move towards the food source (i.e. optimal regions) to accelerate the search process. In addition, $pbest'_{id}$ is further defined in Eq. (8), where $pbest^k_{id}$ represents the personal best solution for the i-th particle obtained in the k-th iteration, $k = 1, 2, \ldots, t$. As indicated in Eq. (8), instead of using the personal best solution identified from the current iteration, the proposed attraction action is enhanced by employing the mean of the personal historical best experiences obtained from the past t number of iterations. Overall, this food chasing mechanism can be viewed as a special case of CS, where $c = \alpha s \otimes H(p_a - \varepsilon)$ as shown in Eq. (2), and also a special case of PSO, where the personal and global best solutions are combined to guide the search process.

Algorithm 1. Pseudo-Code of the Proposed BBPSO Algorithm 1.

1.	**Start**
2.	Initialize the position of each particle in the swarm;
3.	**Repeat {**
4.	**For** each particle in the population **do**
5.	{
6.	Evaluate each particle using the defined fitness function;
7.	Update the position of each particle using a randomly selected movement from the following three operations;
8.	(1) $x_{id}^{t+1} = \phi(\frac{pbest_{id}^t + gbest_d^t}{2}, \|pbest_{id}^t - gbest_d^t\|)$ //the original BBPSO movement
9.	(2) $x_i^{t+1} = x_i^t + c\,(\frac{pbest'_{id} + gbest_d^t}{2} - x_i^t)$ // attraction behaviour with c as the Logistic map
10.	(3) $x_i^{t+1} = x_i^t - c\left(\frac{pworst'_{id} + gworst_d^t}{2} - x_i^t\right) - \alpha\varepsilon$ //avoid the less optimal regions. $\alpha \in (0, 1)$ and ε as Levy distribution.
11.	Update the personal best $pbest$ and the global best $\boldsymbol{gbest}$;
12.	}**End for**
13.	} **Until (Termination criteria are met);**
14.	Return the most optimal solution, $\boldsymbol{gbest}$;
15.	**End**

3.1. The proposed BBPSO algorithms with attraction and flee operations

Motivated by the accelerated search strategies of PSO, CS and DA, we propose two modified BBPSO algorithms that incorporate two new operations, i.e., attraction to the food source and flee from the enemies. The aim is to mitigate premature convergence of the original BBPSO algorithm. The first variant explores both the attraction and flee operations in the primary swarm while the second embeds them in the subswarm-based search. These special food chasing and fleeing behaviours show great potential in increasing local and global search capabilities of the original BBPSO algorithm. The details of both proposed variants are as follows.

3.1.1. The modified BBPSO variant 1

As mentioned above, the first proposed BBPSO algorithm (denoted as Algorithm 1) incorporates not only the conventional movement of BBPSO defined in Eq. (1), but also the newly proposed attraction and flee operations to guide the search process. The new search behaviour with attraction to the food source is defined in Eqs. (7) and (8).

$$x_i^{t+1} = x_i^t + c\left(\frac{pbest'_{id} + gbest_d^t}{2} - x_i^t\right) \tag{7}$$

$$pbest'_{id} = \frac{\sum_{k=1}^{t} pbest_{id}^k}{t} \tag{8}$$

where $pbest'_{id}$ and $gbest_d^t$ represent the average personal best experience and the global best solution in the d-th dimension, respectively. Note that c denotes the Logistic chaotic map, which

Guided by the global and the average personal best solutions, the proposed attraction search mechanism enables the overall population to reach promising search regions more efficiently in fewer iterations. It also shows great efficiency in escaping from the local optimum trap owing to the consideration of both local and global promising solutions.

Motivated by the concept of enemy avoidance in DA, the proposed flee mechanism is defined in Eqs. (9) and (10).

$$x_i^{t+1} = x_i^t - c\left(\frac{pworst'_{id} + gworst_d^t}{2} - x_i^t\right) - \alpha\varepsilon \tag{9}$$

$$pworst'_{id} = \frac{\sum_{k=1}^{t} pworst_{id}^k}{t} \tag{10}$$

where $pworst'_{id}$ and $gworst_d^t$ represent the average personal worst experience and the global worst solution in the d-th dimension, respectively, while α and ε represent a randomization vector (with each dimension $\in (0, 1)$) and a random walk strategy such as the Levy flights, respectively. Note that c also denotes the Logistic chaotic map. This search action allows each particle to flee away from enemies and move away from less optimal search regions (e.g. avoidance of enemies) to achieve fast convergence. In addition, $pworst'_{id}$, is further defined in Eq. (10), where $pworst_{id}^k$ represents the personal worst solution for the i-th particle identified in the k-th iteration. Similar to the attraction behaviour, instead of using the personal worst solution identified from the current iteration, the proposed flee operation is enhanced by using the mean of the personal historical worst experiences from the past t number of iterations.

In comparison with the evading action in DA, the proposed mechanism is guided by both average personal historical and global worst experiences for enemy avoidance. It provides a way for each

particle to flee away from local and global unpromising search regions effectively, therefore accelerating convergence.

Algorithm 1 lists the first proposed BBPSO variant. After initialising the original swarm, in each iteration, any of the three actions (i.e.

ties. Algorithm 2 shows the pseudo-code of the proposed second BBPSO variant, while its flowchart is shown in Fig. 2.

Algorithm 2. Pseudo-Code of the Proposed BBPSO Algorithm 2.

1.	**Start**
2.	Initialize the position of each particle in the swarm;
3.	**Repeat** {
4.	**Repeat** {//perform the original BBPSO operation
5.	**For** each particle in the population **do**
6.	{
7.	Evaluate each particle using the defined fitness function;
8.	Update the position of each particle using Equation (1);
9.	Update the individual best *pbest* and the global best ***gbest_bbpso*** in the population;
10.	}**End for**
11.	}**Until (stopping condition);**
12.	Divide the population into two subswarms *s1* and s*2*;
13.	
14.	**//Attraction to food source in subswarm *s1***
15.	**Repeat**{
16.	Randomly select any movement among Equations (1) and (7) to update the position of each particle in subswarm *s1*;
17.	Update the best solution, ***gbest_s1***, in the subswarm 1*;*
18.	}**Until (stopping condition);**
19.	
20.	**//Fleeing from enemies in subswarm *s2***
21.	**Repeat**
22.	{
23.	Randomly select any search operation among Equations (1) and (9) to update the position of each particle in subswarm *s2*;
24.	Update the best solution, ***gbest_s2,*** in the subswarm 2;
25.	}**Until (stopping condition);**
26.	
27.	Compare the fitness of ***gbest_bbpso***, ***gbest_s1***, and ***gbest_s2*** and assign the best leader to ***gbest.***
28.	Discard the worst leader among ***gbest_bbpso***, ***gbest_s1***, and ***gbest_s2*** and combine *s1* and *s2* into one population;
29.	Re-insert ***gbest*** into the updated population as the swarm leader;
30.	} **Until (stopping condition).**
31.	Return the most optimal solution(s);
32.	**End**

the movements defined in Eqs. (7) and (9) and the original search behaviour of BBPSO defined in Eq. (1)) is randomly selected to guide the search of each particle. Moreover, the original search strategy of BBPSO together with both attraction to the food source and fleeing from enemies operations work in a collaborative manner to drive the search process out of the local optimum trap. For instance, when the search guided by the food chasing behaviour that follows the global and the average personal best experiences (e.g. the special cases of PSO and CS) stagnates, the fleeing from enemies operation is able to drive the particles out of the less optimal regions, in order to overcome premature convergence. On the other hand, when the enemy avoidance behaviour shows limited improvements in finding the best solution, the search mechanism guided by the global and the average personal best solutions leads the overall population to reach more promising search regions effectively, in order to escape from the local optimum trap. These two new search operations and the original search behaviour of BBPSO work alternatively to increase search diversity and attain global optima.

3.1.2. The proposed BBPSO variant 2 with subswarms

In this research, both the newly proposed search mechanisms, i.e. attraction to the food source and fleeing from enemies, are also embedded in the subswarms to evaluate their efficiency. The proposed second BBPSO variant (denoted as Algorithm 2) explores these new search mechanisms in subswarm-based search activi-

As illustrated in Algorithm 2, this second BBPSO variant with subswarms firstly performs the conventional BBPSO operation for *N* number of iterations to identify the initial global best solution, *gbest_bbpso*. Subsequently, the overall population is divided into two subswarms, *s1* and *s2*. We employ the newly proposed search mechanisms, i.e. attraction to the food source and fleeing from enemies, in the subswarms, in the search process. Specifically, we embed the original movement of BBPSO and the food chasing behaviour in subswarm *s1*, and randomly select any of the two operations in each iteration to update the position of each particle in *s1*. Similarly, the original operation of BBPSO and the fleeing from enemies mechanism are employed in subswarm *s2*. In each iteration, any of the two operations is randomly selected to update the position of each particle in *s2*. After *N* number of iterations, the subswarm leaders of both *s1* and *s2*, i.e. *gbest_s1* and *gbest_s2*, are identified. Then, the three optimal solutions, i.e. *gbest_bbpso*, *gbest_s1* and *gbest_s2,* are compared with each other. The one with the highest fitness value is identified as the global best solution, i.e. *gbest,* while, the worst leader among the three is discarded. Moreover, both food chasing and enemy fleeing operations embedded in the subswarms also work collaboratively to reduce the probability of being trapped in local optima.

The search process iterates until the termination criteria are met, i.e. (1) the maximum number of generations is reached, or (2) the most optimal solution is found. In this research, the initial population of 30 particles and the maximum number of generation of 200 are set for each experiment. The fitness function defined in Eq. (11), as formulated in other studies [32,33], is used to evaluate
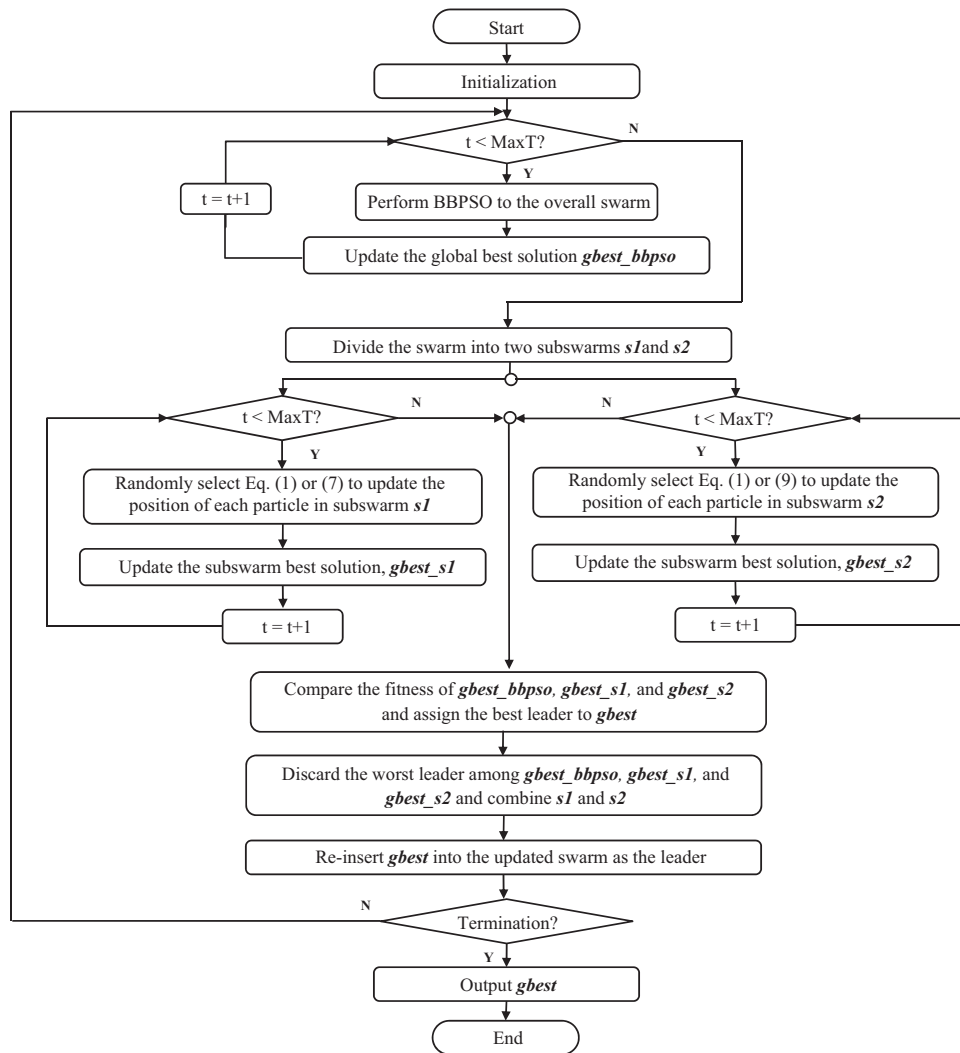
Fig. 2. The flowchart of the proposed BBPSO algorithm 2 with subswarms.

each particle.

$$\text{fitness}\,(C) = \mu * \text{performance}_C + (1 - \mu) * (\text{number}_{\text{features}C})^{-1}$$

$$(11)$$

where μ and $1 - \mu$ denote the weights for classification performance, performance$_C$, and the number of selected features, number$_{\text{features}C}$, respectively. Since classification performance is more important than the number of selected features, μ is assigned a higher value than that of $1 - \mu$. For fitness evaluation, we convert the continuous value in each dimension of each particle into a binary setting (i.e. 0 or 1) with '1' indicating the selection of a specific dimension and '0' representing the non-selection of that dimension. In order to have slow movements and avoid premature convergence, we use a continuous value in each dimension for each particle during the search process, and conversion to a binary setting takes place only for fitness evaluation.

Since the samples extracted from the ALL-IDB2 database are limited, the training and test data sets employed for the evaluation of the proposed algorithms exhibit imbalanced class instances. Instead of using the traditional accuracy measure, we employ the geometric mean (GM) as the performance indicator because it is frequently used for evaluation of imbalanced data problems [34,35]. Therefore, it is employed in this research for fitness evaluation

during the training phase and generation of the final classification results during the test phase.

We have also compared the two proposed BBPSO algorithms (with and without subswarms) with other conventional and state-of-the-art metaheuristic search methods to evaluate their efficiency. The detailed evaluation results are provided in Section 4.

3.2. ALL recognition

The proposed BBPSO based optimization algorithms are first used to identify the significant discriminative features pertaining to healthy and blast cells. Then, the selected features are normalized into the range of [−1,1]. Two classification techniques, i.e. 1NN and RBF-based SVM, are used for ALL detection in this research. 1NN is selected because it is a popular nonparametric classification technique with an efficient computational cost, and it has been employed frequently for fitness evaluation and/or ALL classification [12,36,37]. The RBF-based SVM model is selected because the RBF kernel supports nonlinear mapping of data samples and possesses fewer hyper-parameters. In order to provide a concrete evaluation methodology, we employ a consistent classification technique for both fitness evaluation (during training) and test in each experiment. Specifically, during the training stage, 1NN and RBF-based SVM with 10-fold cross validation are employed for fitness evalu-

Table 1
Parameters and levels used in the CCD method.

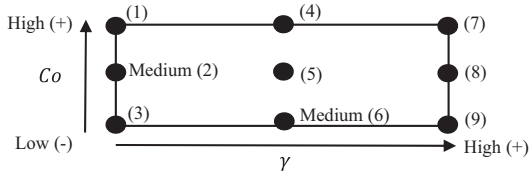| Parameters\levels	Low (−1)	Medium (0)	High (+1)
γ	2	4	8
Co	2	4	8

Fig. 3. The 9 experimental configurations in CCD employed in this research.

ation. During the test stage, the trained 1NN and RBF-based SVM models with hold-out validation are utilized for ALL classification for each test set.

In order to maximize the SVM performance, during both fitness evaluation and test, it is necessary to identify the optimal parameter settings for the scaling factor, γ, and the soft margin constant, Co. Although the grid search method is often used for parameter selection, it is a computational-expensive process [38]. This problem is intensified as we conduct the fitness evaluation for each particle in each iteration during the training stage. Therefore, parameter tuning using the grid search method during the training stage is computationally prohibitive, and sometimes infeasible. To mitigate this problem, a widely accepted design of experiment method, i.e. the centre composite design (CCD) [35,39], is employed to identify the optimal parameter setting of SVM during the training stage. This CCD method divides each parameter into different levels, and evaluates all the possible combinations from different levels pertaining to the parameters for optimal setting selection. It offers an affordable computational cost in comparison with that of grid search. This design of experiment method is especially useful when the optimal setting is subject to multiple (e.g. more than two) parameters, in order to overcome the computational cost of the grid search method.

In this research, the CCD method with 10-fold cross validation is used to identify the optimal parameter setting for Co and γ of SVM at the training stage. Following the CCD method, three different levels, i.e. low, medium and high, have been identified for each parameter. Therefore, a full factorial design of 9 (i.e. 3^2) key combinations of Co and γ has been used. Since CCD employs fewer parameter combinations as compared with those of grid search, we employ the ranges of $2^1 - 2^3$ and $2^1 - 2^3$ for Co and γ, respectively, for the search of the optimal SVM parameter settings. These ranges are identified based on trial and error for diverse training sets employed in this research. Table 1 and Fig. 3 indicate the detailed parameter settings and experimental configurations.

During the training stage, the CCD method uses all 9 key combinations of Co and γ (as indicated in Fig. 3) to evaluate each particle. The parameter setting that achieves the best GM score for a specific particle under 10-fold cross validation is used to generate the fitness value of that particle using Eq. (11). The optimal setting identified for the final global best solution is used to train the SVM using the entire training set. The trained SVM model is subsequently used with hold-out validation for evaluation of the test set.
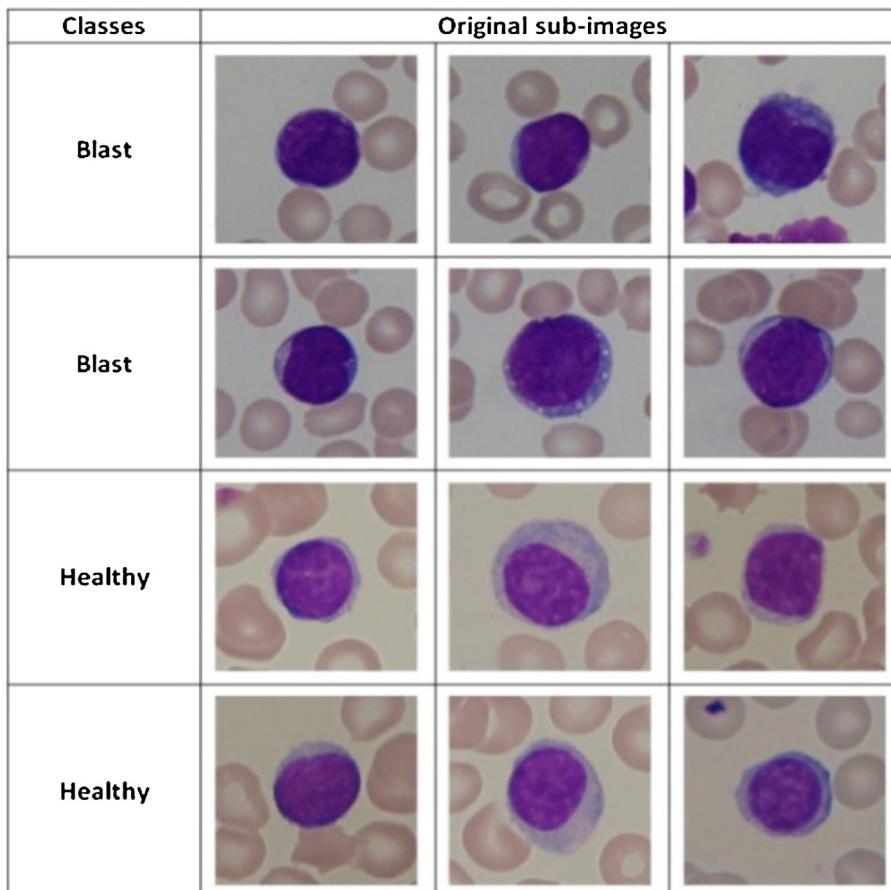

Fig. 4. Sample sub-images from the ALL-IDB2 data set.

Table 2
Average GM results over 30 runs for 90 test images and the results with the entire set of 80 raw features.

Methods	Avg. no of selected features	1NN (hold-out)	Avg. no of selected features	SVM (hold-out)
GA	34.23	0.7935	33.30	0.7458
PSO	34.47	0.8977	34.07	0.8597
BBPSO	35.37	0.8989	33.57	0.8482
Binary BBPSO [12]	44.87	0.9081	41.77	0.8659
ELPSO [24]	33.8	0.8759	34.93	0.8618
The prop. BBPSO 1 (without subswarms)	**10.33**	**0.9315**	**10.50**	**0.9171**
The prop. BBPSO 2 (with subswarms)	**26.83**	**0.9567**	**26.33**	**0.9287**
80 (entire set)	–	0.8683	–	0.8322

4. Evaluation

To compare the proposed BBPSO algorithms with other state-of-the-art PSO variants and classical methods, we implement the following search methods, i.e. BBPSO, binary BBPSO [12], ELPSO [24], GA, and PSO. We employ 180 microscopic images from the ALL-IDB2 database for experimentation. Some sample images are illustrated in Fig. 4. Two experimental settings with different proportions of training and test samples have been used to evaluate the efficiency of both of the proposed BBPSO algorithms.

The first experimental setting employs two distinctive sets of 90 images for training and test, respectively. For both data sets, two thirds of the images (i.e. 60) are blast cells while the remaining images (i.e. 30) represent healthy cells. The second experimental setting uses 100 images with 50 normal and 50 abnormal instances for training, and the remaining 80 (70 abnormal and 10 normal) unseen images for test. Since our algorithms and other optimization methods are all stochastic methods, we have conducted 30 trials for each algorithm. As discussed earlier, during the training stage, both 1NN and RBF-based SVM with 10-fold cross validation have been employed for fitness evaluation of each algorithm. During the test stage, the trained 1NN and RBF-based SVM models with hold-out validation have been employed for evaluation of all methods.

For the first experimental setting, we have compared the proposed algorithms with other state-of-the-art PSO variants and classical search methods over 30 trials. Table 2 shows the average GM performance of each algorithm integrated with both 1NN and SVM classifiers for evaluation of the 90 unseen test images over 30 runs. To compare the efficiency of the feature optimization process, the classification results using the entire set of 80 raw features without any feature selection are provided in the last row of Table 2. Overall, both proposed BBPSO algorithms have comparatively higher convergence rates with smaller numbers of iterations to achieve a reasonable classification performance

of normal and abnormal lymphocytic cells. Comparatively, other methods exhibit lower convergence rates with more iterations to produce the results shown in Table 2.

As illustrated in Table 2, with the 1NN classifier, the first BBPSO variant without subswarms (i.e. Algorithm 1) achieves an average GM performance of 93.15%, while the second BBPSO variant with subswarms (i.e. Algorithm 2) achieves the highest GM measure of 95.67%, over 30 runs. Both algorithms outperform all other methods, and exhibit the fastest convergence speed in comparison with other classical methods and PSO variants. Both algorithms are able to converge within 80 to100 iterations, on average, over 30 runs. The average numbers of selected features for Algorithms 1 and 2 are 10.33 and 26.83, respectively. Comparatively, other methods achieve convergence within 150–200 iterations with a larger set of features, e.g. an average of 35.37 selected features for BBPSO, 44.87 for binary BBPSO, 33.8 for ELPSO, 34.23 for GA and 34.47 for PSO. Algorithm 1 outperforms BBPSO, binary BBPSO, ELPSO, GA and PSO by 3.26%, 2.34%, 5.56%, 13.8% and 3.38%, respectively. Algorithm 2 outperforms BBPSO, binary BBPSO, ELPSO, GA and PSO by 5.78%, 4.86%, 8.08%, 16.32%, and 5.9%, respectively. When SVM is used for fitness evaluation and test, Algorithm 1 achieves an average GM score of 91.71% over 30 runs. It outperforms BBPSO, binary BBPSO, ELPSO, GA and PSO by 6.89%, 5.12%, 5.53%, 17.13% and 5.74%, respectively. Similarly, Algorithm 2 obtains an average GM performance of 92.87% over 30 runs. It outperforms BBPSO, binary BBPSO, GA and PSO by 8.05%, 6.28%, 6.69%, 18.29% and 6.9%, respectively.

Furthermore, as indicated in Table 2, the experimental results of both proposed algorithms outperform those using the original set of 80 raw features without any feature selection greatly. Under the same experimental setting, the performances of some baseline methods are comparable with, or sometimes lower than, the result obtained using the 80 raw features. From the clinical perspective, important features for ALL diagnosis include the cytoplasm and nucleus areas, ratio between the nucleus area and the cyto-
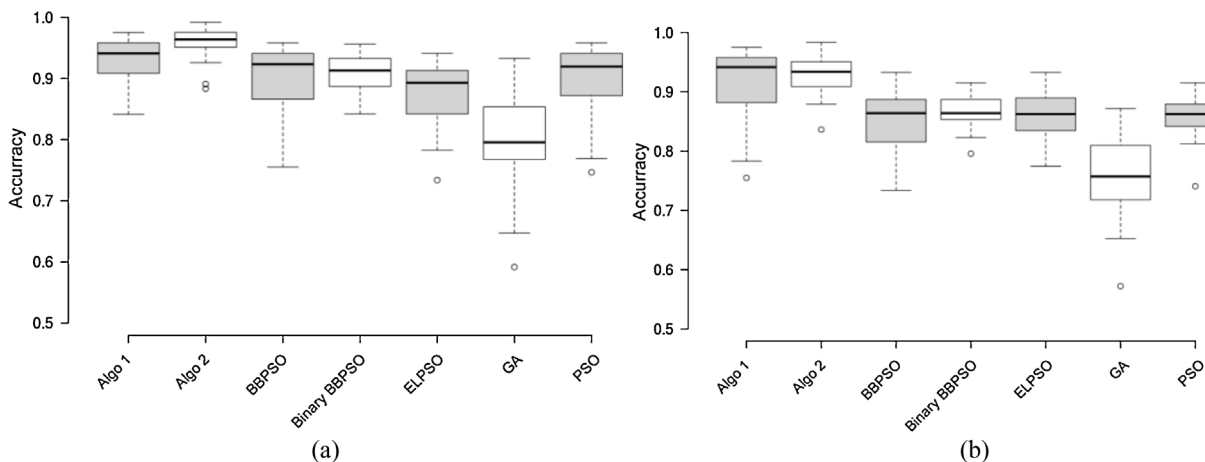

Fig. 5. Boxplots for different optimization methods integrated with 1NN (a) and SVM (b), respectively.

Table 3
Average GM results over 30 runs for 80 test images and the results with the entire set of 80 raw features.

Methods	Avg. no. of selected features	1NN (hold-out)	Avg. no of selected features	SVM (hold-out)
GA	30.93	0.7900	25.90	0.8261
PSO	33.50	0.9016	36.53	0.9279
BBPSO	33.53	0.9063	36.33	0.9331
Binary BBPSO [12]	39.83	0.8802	29.87	0.8987
ELPSO [24]	32.20	0.9027	31.87	0.9024
The prop. BBPSO 1 (without subswarms)	**20.33**	**0.9463**	**17.00**	**0.9480**
The prop. BBPSO 2 (with subswarms)	**24.1**	**0.9561**	**26.43**	**0.9614**
80 (entire set)	-	0.8185	-	0.9281

plasm area, form factor and compactness (supporting diagnosis in terms of irregularity of cell shape in nucleus), perimeter, texture changes related to open or close chromatin, and eccentricity [1–5]. The experimental results indicate that these important features have been identified and included in the selected feature subsets of both proposed algorithms. Note that a few of the abovementioned clinically important features, such as the nucleus area and ratio of nucleus to cytoplasm, are often overlooked, or they do not co-exist in the selected feature subsets by other optimization methods, despite the fact that these methods sometimes select more features, which may lead to performance degradation owing to the use of insignificant features.

We have also compared the proposed algorithms and other methods using boxplots. Fig. 5 illustrates the classification performance variations for all algorithms over 30 trials for 90 test images with 1NN and SVM, respectively. In both boxplot diagrams, the first two boxplots depict the results of Algorithm 1 (without subswarms) and Algorithm 2 (with subswarms), respectively. Integrated with 1NN, Algorithm 2 achieves the highest average GM performance of 95.67% over 30 runs. As shown in Fig. 5(a), 50% of the GM results of Algorithm 2 (with the median of 96%) are higher than the maximum results of BBPSO (96%), binary BBPSO (96%), and PSO (96%). Besides that, 25% of the results of Algorithm 1 (with the 3rd quartile of 96%) outperform the maximum scores of BBPSO, binary BBPSO and PSO. In addition, 75% of the GM results of Algorithm 2 (with the 1st quartile of 95%) are higher than the maximum results of ELPSO (94%) and GA (93%), while 50% of the results of Algorithm 1 (with the median of 94%) are higher than the maximum results of ELPSO and GA.

As indicated in Fig. 5(b), in combination with SVM, 50% of the GM results of both proposed algorithms (with the medians of 94% for Algorithm 1 and 93% for Algorithm 2) are higher than the maximum GM scores of BBPSO (93%), binary BBPSO (92%), ELPSO (93%) and PSO (92%). At least 75% of the classification results of both algo-

rithms (with the 1st quartiles of 88% for Algorithm 1 and 91% for Algorithm 2) are higher than the maximum result of GA (87%). The median (94%) of Algorithm 1 outperforms those of BBPSO (86%), binary BBPSO (86%), ELPSO (86%), GA (76%) and PSO (86%) by 8%, 8%, 8%, 18%, and 8%, respectively.

In the second experimental setting, we have evaluated all algorithms using 100 images for training with a 50:50 split for healthy and blast instances and 80 (70 abnormal and 10 normal) unseen images for test. A total of 30 runs for each algorithm have been conducted. Table 3 shows the average GM results integrated with both 1NN and RBF-based SVM for evaluation of the 80 unseen test images over 30 runs, as well as the result from the original set of 80 raw features for this experimental setting. Fig. 6 shows the classification performance variations in boxplots of both proposed algorithms and all other methods.

As shown in Table 3, with the SVM classifier, Algorithm 2 with subswarm-based attraction and fleeing operations achieves the highest average GM score of 96.14%, while Algorithm 1 without subswarms achieves an average GM performance of 94.80%. Both algorithms outperform all other methods with fewer selected features (17 for Algorithm 1 and 26.43 for Algorithm 2). Algorithm 2 outperforms GA, PSO, BBPSO, binary BBPSO and ELPSO by 13.53%, 3.35%, 2.83%, 6.27%, and 5.9%, respectively. With the 1NN classifier, Algorithm 2 achieves the best GM performance of 95.61% and outperforms GA, PSO, BBPSO, binary BBPSO and ELPSO by 16.61%, 5.45%, 4.98%, 7.59%, and 5.34%, respectively.

As indicated in Fig. 6(a), in combination with the 1NN classifier, 25% of the results of Algorithm 2 (with the 3rd quartile of 97%) outperform the maximum results of all other methods. All the results of both proposed algorithms are higher than the maximum GM score (91%) of the GA. The medians of both algorithms (i.e. 96%) are also higher than those of BBPSO (92%), binary BBPSO (88%), ELPSO (91%), GA (79%) and PSO (91%) by 4%, 8%, 5%, 17%, and 5%, respectively.
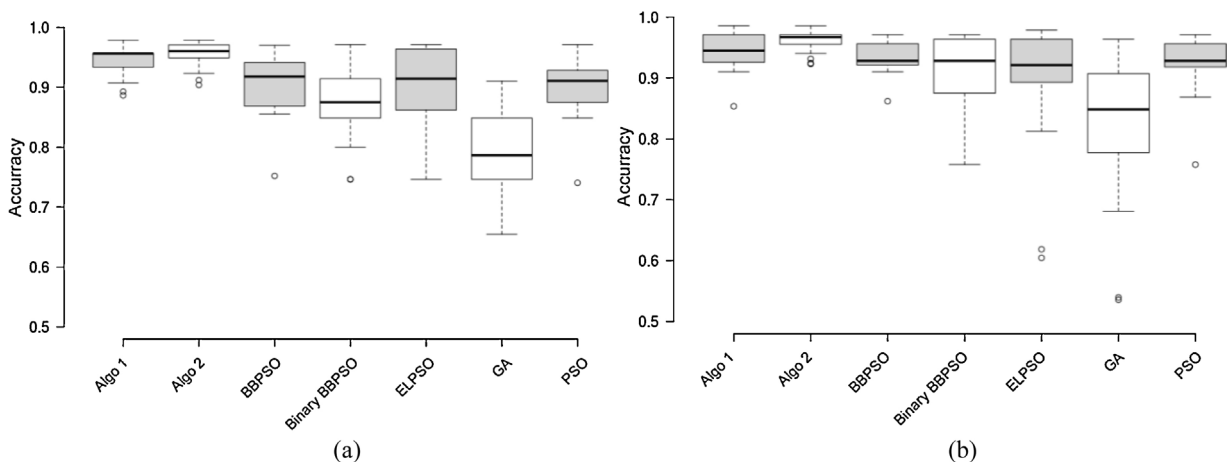

(a)   (b)

Fig. 6. Boxplots for different optimization methods integrated with 1NN (a) and SVM (b), respectively.

Table 4
Average GM results for 90 test images using the fitness function defined in Eq. (12).

Methods	Avg. no of selected features	1NN (hold-out)	Avg. no of selected features	SVM (hold-out)
GA	33.83	0.7909	37.23	0.7584
PSO	32.57	0.8970	31.63	0.8676
BBPSO	32.93	0.8951	32.37	0.8575
Binary BBPSO [12]	37.77	0.8970	37.13	0.8595
ELPSO [24]	32.37	0.8717	32.87	0.8553
The prop. BBPSO 1 (without subswarms)	**6.6**	**0.9409**	**10.77**	**0.9130**
The prop. BBPSO 2 (with subswarms)	**5.97**	**0.9478**	**11.17**	**0.9270**
80 (entire set)	–	0.8683	-	0.8322

Fig. 6(b) indicates that in combination with the SVM classifier, 25% of the results of Algorithm 1 (with the 3rd quartile of 97%) are higher than the maximum performances of BBPSO (97%), binary BBPSO (97%), GA (96%) and PSO (97%). Similarly, 50% of the GM results of Algorithm 2 (with the median of 97%) outperform the maximum scores of BBPSO, binary BBPSO, GA and PSO.

Since the current fitness function defined by Eq. (11) focuses more on classification performance as compared with the number of selected features, we have applied another fitness function defined in Eq. (12), which indicates a comparatively more balanced trade-off between the GM result and the number of selected features.

$$\text{fitness}_2\,(C) = \mu * \text{Performance}_C + (1-\mu) * \left(1 - \frac{\text{number}_{\text{features}\,C}}{\text{number}_{\text{all}}}\right)$$
(12)

where $\text{number}_{\text{all}}$ and $\text{number}_{\text{features}\,C}$ indicate the overall number of raw features (i.e. 80) and the number of selected features, respectively. The second part of Eq. (12) indicates that the number of selected features has more influences on the overall fitness function than the corresponding part of the original fitness function defined in Eq. (11). In addition, the same weight settings of μ and $1-\mu$ as in Eq. (11) are used in Eq. (12). Note that Performance_C represents the performance evaluation using GM.

A series of tests has been conducted using the new fitness function in Eq. (12) for the above two experimental settings, in order to further evaluate the efficiency of the proposed algorithms. Evaluated with a benchmark of 30 runs using the first experimental setting (with unbalanced 90 images for training and 90 unseen samples for test), Table 4 and Fig. 7 illustrate the average GM scores of all methods and the detailed performance variations of all algorithms integrated with both 1NN and SVM classifiers over 30 runs, respectively.

When integrated with 1NN classifier, the results in Table 4 indicate that Algorithm 2 achieves a higher average GM score of 94.78%, whereas Algorithm 1 obtains an average GM result of 94.09%. Both proposed algorithms outperform all other search methods. Specifically, Algorithm 1 outperforms BBPSO, binary BBPSO, ELPSO, GA, and PSO by 4.58%, 4.39%, 6.92%, 15% and 4.39%, respectively. Algorithm 2 outperforms BBPSO, binary BBPSO, ELPSO, GA, and PSO by 5.27%, 5.08%, 7.61%, 15.69% and 5.08%, respectively. Furthermore, in combination with the SVM classifier, Algorithm 2 achieves the best GM performance of 92.70%. It outperforms BBPSO, binary BBPSO, ELPSO, GA, and PSO by 6.95%, 6.75%, 7.17%, 16.86% and 5.94%, respectively.

As shown in Fig. 7(a), integrated with the 1NN classifier, 25% of the results of both proposed algorithms (with the 3rd quartile of 96%) are higher than the maximum GM performances of all other methods. The median (95%) of both proposed algorithms also outperforms those of BBPSO (89%), binary BBPSO (91%), ELPSO (89%), GA (79%), and PSO (91%) by 6%, 4%, 6%, 16% and 4%, respectively. When SVM-based classification is utilized, as illustrated in Fig. 7(b), 50% of the results of both proposed algorithms (with the same median of 93%) are higher than the maximum GM scores of nearly all other methods. The median of both proposed algorithms also outperforms those of BBPSO (86%), binary BBPSO (87%), ELPSO (86%), GA (77%), and PSO (87%) by 7%, 6%, 7%, 16% and 6%, respectively.

We have used the new fitness function defined in Eq. (12) to further evaluate the efficiency of all algorithms under the second experimental setting with balanced 100 images for training and 80 unseen images for test. A total of 30 runs have been conducted for each algorithm. Table 5 and Fig. 8 show the average GM results of all methods and the detailed performance variations of all algorithms integrated with both classifiers over 30 trials, respectively.

As shown in Table 5, when 1NN is used, Algorithm 2 achieves the highest average GM score of 96.25%, whereas Algorithm 1
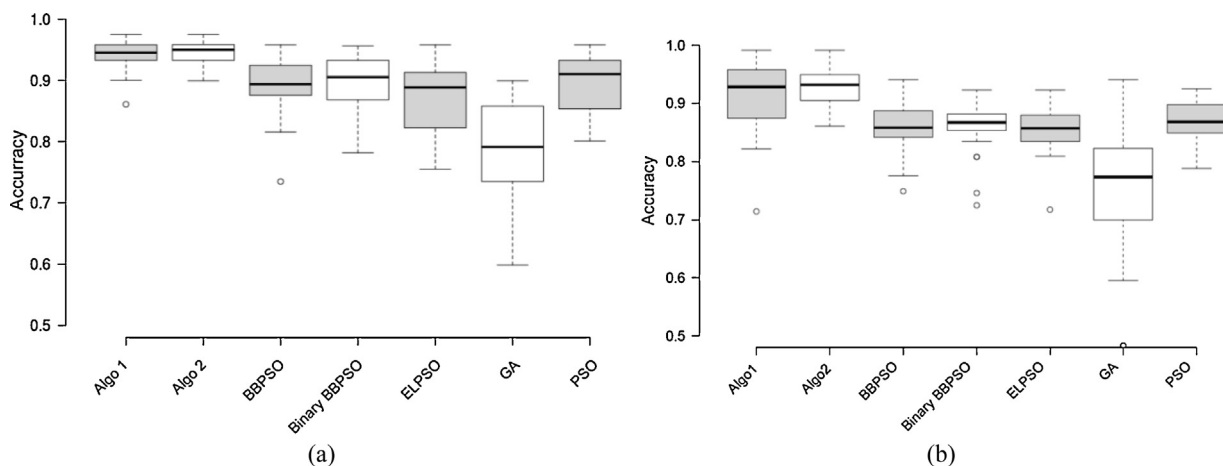

Fig. 7. Boxplots for different optimization methods integrated with 1NN (a) and SVM (b), respectively.

Table 5
Average GM results for 80 test images using the fitness function defined in Eq. (12).

Methods	Avg. no. of selected features	1NN (hold-out)	Avg. no of selected features	SVM (hold-out)
GA	35.53	0.7872	29.53	0.8310
PSO	32.40	0.9047	32.57	0.8970
BBPSO	31.73	0.8964	31.20	0.9138
Binary BBPSO [12]	36.53	0.8940	36.03	0.9227
ELPSO [24]	30.47	0.8864	33.37	0.9166
The prop. BBPSO 1 (without subswarms)	**18.00**	**0.9494**	**20.87**	**0.9453**
The prop. BBPSO 2 (with subswarms)	**20.97**	**0.9625**	**17.27**	**0.9619**
80 (entire set)	-	0.8185	-	0.9281

achieves an average GM result of 94.94%. Both proposed algorithms outperform all the other methods significantly. Algorithm 2 outperforms BBPSO, binary BBPSO, ELPSO, GA, and PSO by 6.61%, 6.85%, 7.61%, 17.53% and 5.78%, respectively. When integrated with SVM, Algorithm 2 achieves the best GM performance of 96.19%. It outperforms BBPSO, binary BBPSO, ELPSO, GA, and PSO by 4.81%, 3.92%, 4.53%, 13.09% and 6.49%, respectively.

As indicated in Fig. 8(b), when integrated with the SVM classifier, 25% of the results of Algorithm 1 and 50% of the GM performances of Algorithm 2 (with the 3rd quartile of 97% for Algorithm 1 and the median of 97% for Algorithm 2) are higher than the maximum results of all other methods. The median (97%) of Algorithm 2 also outperforms those of BBPSO (92%), binary BBPSO (92%), ELPSO (92%), GA (86%) and PSO (93%) by 5%, 5%, 5%, 11% and 4%, respectively. The minimum performance of Algorithm 2 (with the lower whisker of 96%) is higher than 75% of the results of BBPSO (with the 3rd quartile of 96%), binary BBPSO (with the 3rd quartile of 94%), ELPSO (with the 3rd quartile of 93%), GA (with the 3rd quartile of 92%), and PSO (with the 3rd quartile of 95%). In combination with 1NN, as indicated in Fig. 8(a), the medians of both proposed algorithms outperform those of all other methods significantly. The median (96%) of both algorithms is higher than those of BBPSO (91%), binary BBPSO (89%), ELPSO (90%), GA (80%) and PSO (92%) by 5%, 7%, 6%, 16% and 4% respectively. Besides that, 25% of the results of Algorithm 2 (with the 3rd quartile of 98%) are higher than the maximum GM results of nearly all other methods.

Evaluated with the new fitness function under the two experimental settings, the empirical results further ascertain the effectiveness of the proposed algorithms. Overall, the proposed algorithms outperform all other PSO variants and classical search methods across two different experimental settings with two fitness functions.

We have also compared our results with other related ALL detection studies reported in the literature. To the best of our knowledge, Putzu et al. [1] and Madhukar et al. [40] achieved high recognition performances using the ALL-IDB database. Evaluated using 10-fold cross validation, Putzu et al. [1] obtained 93.2% accuracy using RBF-based SVM with 131 features. Using SVM and leave-one-out cross validation, Madhukar et al. [40] achieved 93.5% accuracy with a high dimensional feature vector consisting of shape, texture, and HD features of the nuclei extracted to distinguish the normal and blast cells. However, both studies did not employ feature selection processes. Evaluated using SVM with hold-out validation, when trained with balanced 100 instances and tested with 80 unseen images, our study yields comparatively smaller discriminative feature subsets for healthy and blast cells classification, and achieves average GM results of 94.80% and 96.19% over 30 runs using Algorithm 1 with the fitness function defined in Eq. (11) and Algorithm 2 with the fitness function defined in Eq. (12), respectively. The performances of our algorithms are obtained by averaging the results of 30 runs in each experimental setting. In short, our algorithms compare favourably with other related methods for ALL detection reported in the literature, indicating the efficiency of the proposed search mechanisms.

Overall, in comparison with all other methods, the proposed algorithms incorporate two new accelerated chaotic search behaviours. Algorithm 1 employs the food chasing movement motivated by the special cases of PSO and CS to move towards promising search regions and the flee operation to avoid unattractive areas. Both these behaviours help overcome the local optimum trap and achieve fast convergence. On the other hand, Algorithm 2 enables the subswarm-based attraction and flee operations to work in a collaborative manner to avoid premature convergence. Both proposed search behaviours have also been further enhanced by using the Logistic chaotic map and the average personal best and worst
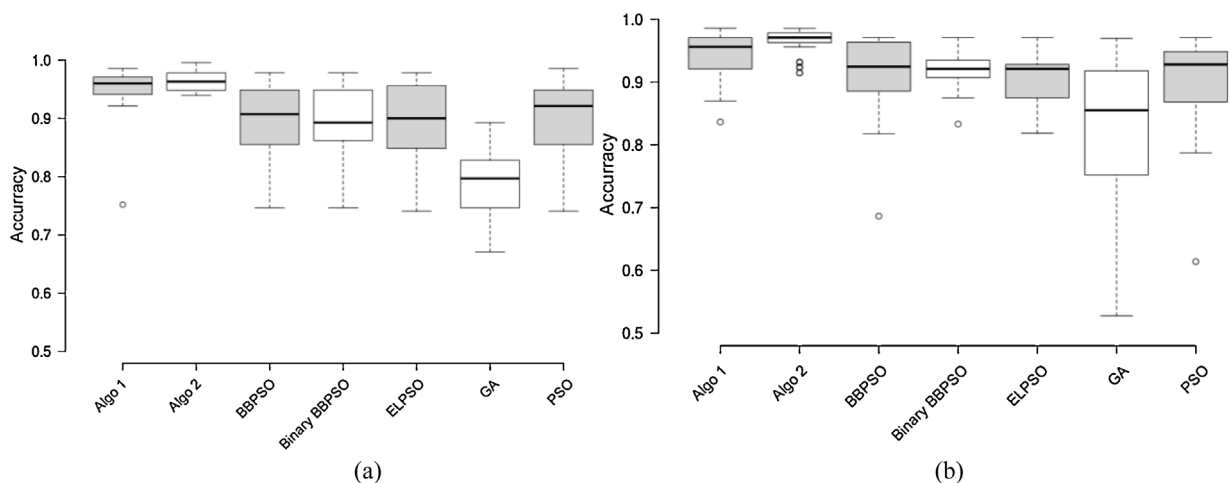

Fig. 8. Boxplots for different optimization methods integrated with 1NN (a) and SVM (b), respectively.

Table 6
GM results for the sonar data set using the fitness function defined in Eq. (11).

Methods	Avg. no. of selected features	1NN (hold-out)	Avg. no of selected features	SVM (hold-out)
GA	24.47	0.8018	20.53	0.7940
PSO	29.80	0.8526	32.17	0.8541
BBPSO	29.43	0.8490	29.50	0.8544
Binary BBPSO [12]	30.67	0.8532	32.13	0.8443
ELPSO [24]	28.37	0.8564	31.70	0.8505
The prop. BBPSO 1 (without subswarms)	**30.33**	**0.8810**	**28.00**	**0.8885**
The prop. BBPSO 2 (with subswarms)	**28.30**	**0.8984**	**33.43**	**0.8839**
60 (entire set)	-	0.8405	-	0.8280

Table 7
GM results for the sonar data set using the fitness function defined in Eq. (12).

Methods	Avg. no. of selected features	1NN (hold-out)	Avg. no of selected features	SVM (hold-out)
GA	29.17	0.8213	23.57	0.7938
PSO	26.33	0.8382	28.63	0.8429
BBPSO	26.67	0.8482	29.10	0.8453
Binary BBPSO [12]	28.07	0.8513	34.60	0.8579
ELPSO [24]	27	0.8467	28.47	0.8508
The prop. BBPSO 1 (without subswarms)	**24.6**	**0.8763**	**27.10**	**0.8851**
The prop. BBPSO 2 (with subswarms)	**24.27**	**0.8968**	**22.60**	**0.8789**
60 (entire set)	-	0.8405	-	0.8280

experiences from the past iterations. These mechanisms lead to the impressive performance of the proposed algorithms, which show significant efficiency in escaping from local optima.

To further evaluate the efficiency of the proposed BBPSO algorithms, a new, cross-domain data set (sonar) from the UCI Machine Learning Repository [41] has also been used. This sonar data set has 60 attributes, 2 classes, and 208 instances. We employ 140 and 68 instances for training and test, respectively. The training data set has a balanced number of samples for each class, with the remaining samples for test. Furthermore, each optimization algorithm is used for feature selection with both classifiers using hold-out validation for test. The detailed GM results using both fitness evaluations are shown in Tables 6–7, respectively.

As indicated in Tables 6–7, the two proposed BBPSO algorithms achieve the best performances for this sonar data set, and they outperform all other methods consistently. Using the fitness function defined in Eq. (11), Algorithm 1 with the SVM classifier achieves the highest GM performance of 88.85%. It outperforms BBPSO, binary BBPSO, ELPSO, GA and PSO by 3.41%, 4.42%, 3.8%, 9.45% and 3.44%, respectively. Integrated with the 1NN classifier, Algorithm 2 obtains the best GM performance of 89.84%. It outperforms BBPSO, binary BBPSO, ELPSO, GA and PSO by 4.94%, 4.52%, 4.2%, 9.66% and 4.58%, respectively. The results further ascertain the efficiency of the proposed attraction and flee search mechanisms in both algorithms.

To further indicate the efficiency of the proposed algorithms, the two-sided Wilcoxon rank sum test [42,43] has been conducted. This statistical test is a non-parametric method to determine whether two solutions are significantly different statistically. It tests the null hypothesis (whether both solutions have an equal median) with a p-value, which is set at the 95% significance level (i.e., $\alpha = 0.05$) in this study. As such, the null hypothesis (i.e., both solutions have an equal median) is rejected if the p-value is lower than 0.05. We employ this Wilcoxon rank sum test to further indicate the significance level of the proposed algorithms. Tables 8 and 9 illustrate the detailed results of the rank sum test.

As indicated in Tables 8 and 9, the p-values for all the experiments conducted for ALL classification, are lower than 0.05. This indicates the GM results of our algorithms for both experimental settings with two fitness functions are significantly better than those from other baseline methods, statistically. The results for the sonar data set also indicate that the proposed algorithms outperform other methods statistically, with all the p-values lower than 0.05.

Besides the Wilcoxon rank sum test, another popular statistical test, i.e. the two-tailed sign test [42,44], has also been conducted. In the sign test, a binomial distribution is used to define the critical number of wins needed to achieve the 95% significance level (i.e., $\alpha = 0.05$) under different experimental settings. Pairwise comparisons are conducted and the number of cases that an algorithm is

Table 8
p-Values of the Wilcoxon rank sum test for Algorithm 1, with ObjFun1 and ObjFun2 indicating the fitness functions in Eq. (11) and Eq. (12), respectively.

Datasets		BBPSO	Binary BBPSO	ELPSO	GA	PSO
	90 test images (ObjFun1, 1NN)	0.003366	0.002999	7.12E-06	6.93E-10	0.006017
	90 test images (ObjFun1, SVM)	1.37E-05	4.02E-05	4.05E-05	1.16E-09	1.91E-05
	90 test images (ObjFun2, 1NN)	2.1E-06	1.54E-05	3.39E-08	5.3E-11	1.48E-05
ALL	90 test images (ObjFun2, SVM)	6.52E-05	0.000148	2.19E-05	7.09E-09	0.000396
	80 test images (ObjFun1, 1NN)	0.000168	1.12E-05	0.019694	4.46E-11	6.69E-05
	80 test images (ObjFun1, SVM)	0.010148	0.002763	0.001142	1.56E-08	0.010493
	80 test images (ObjFun2, 1NN)	6.2E-05	0.000124	0.000156	2.43E-10	0.000886
	80 test images (ObjFun2, SVM)	0.009297	0.003619	0.001671	1.12E-06	0.001774
	ObjFun1, 1NN	5.29E-05	0.000462	0.000979	2.24E-06	0.001334
sonar	ObjFun1, SVM	0.000192	6.89E-05	9.31E-05	1.13E-08	0.000699
	ObjFun2, 1NN	0.003615	0.00492	0.00101	3.78E-06	0.000647
	ObjFun2, SVM	0.001012	0.005608	0.000988	3.42E-08	0.000798

Table 9
p-Values of the Wilcoxon rank sum test for Algorithm 2, with ObjFun1 and ObjFun2 indicating the fitness functions in Eqs. (11) and (12), respectively.

Datasets		BBPSO	Binary BBPSO	ELPSO	GA	PSO
ALL	90 test images (ObjFun1, 1NN)	6.36E-08	4.56E-08	1.65E-09	5.89E-11	1.33E-07
	90 test images (ObjFun1, SVM)	7.05E-09	7.61E-09	5.97E-08	4.04E-11	6.45E-09
	90 test images (ObjFun2, 1NN)	1.25E-07	9.79E-07	7.39E-09	3.13E-11	1.01E-06
	90 test images (ObjFun2, SVM)	2.05E-08	1.45E-08	1.38E-08	5.21E-10	1.26E-07
	80 test images (ObjFun1, 1NN)	4.65E-06	1.26E-06	0.001165	3.26E-11	3.46E-06
	80 test images (ObjFun1, SVM)	9.6E-06	5.16E-05	1.82E-05	1.24E-09	4.53E-05
	80 test images (ObjFun2, 1NN)	7.02E-06	2.08E-05	1.7E-05	2.87E-11	4.24E-05
	80 test images (ObjFun2, SVM)	7E-06	9.03E-07	1.22E-06	6.83E-09	2.07E-06
sonar	ObjFun1, 1NN	4.24E-09	5.65E-08	9.99E-08	1.74E-08	6.16E-07
	ObjFun1, SVM	0.000157	4.9E-05	8.56E-05	6.43E-09	0.000892
	ObjFun2, 1NN	9.3E-06	3.53E-05	2.31E-07	3.39E-09	6.89E-07
	ObjFun2, SVM	0.001111	0.018293	0.006644	6.5E-08	0.001591

the overall winner is identified. According to [42], for a total of 12 experiments (as in Tables 2–7), the critical number of wins for the sign test at the 95% significance level is 10. In other words, an algorithm needs to become an overall winner for at least 10 times, in order to achieve the significance level of $\alpha = 0.05$. Since the two proposed algorithms win each baseline method 12 times, respectively, both proposed algorithms show a statistically significant improvement (at the 95% significance level) over GA, PSO, ELPSO, BBPSO and binary BBPSO. These statistical tests indicate the effectiveness and superiority of the proposed algorithms.

5. Conclusions

In this research, we have proposed two modified BBPSO algorithms for feature optimization to enhance ALL classification. Both proposed algorithms employ accelerated chaotic search mechanisms of attraction to the food source and fleeing from enemies to diversify the search and escape from the local optimum trap. These two new search behaviours have also been further enhanced by the Logistic chaotic map. The first proposed algorithm explores the two new search operations in the main swarm, while the second proposed algorithm embeds these actions in the subswarm based search. Moreover, instead of using the personal best and worst experiences from the current iteration for the attraction and flee actions, we take the mean of the personal historical best and worst positions from the past iterations, respectively, to further enhance the proposed search mechanisms. Evaluated with the ALL-IDB2 database with 100 images for training and 80 for test, using 1NN with hold-out validation, our studies achieve superior average GM scores of 94.94% and 96.25% using both proposed Algorithms 1 and 2 with the fitness function defined in Eq. (12), respectively. In comparison with other advanced and classical search methods, both proposed algorithms are able to identify smaller numbers of features and to achieve faster convergence rates, which outperform other methods and related studies for ALL diagnosis reported in the literature. We have also evaluated the proposed algorithms using a cross-domain sonar data set from the UCI Machine Learning Repository. Both proposed algorithms outperform other related methods consistently for the sonar data set, which further indicates the efficiency and robustness of the proposed BBPSO-based feature optimization algorithms. The empirical results from the two statistical tests, i.e. the Wilcoxon rank sum and the sign tests, for both ALL and sonar data sets have also indicated statistically significant performances of both proposed algorithms. For further work, we will use other medical image data sets [45–47] to further evaluate the efficiency of the proposed algorithms in general medical diagnosis problems.

References

[1] L. Putzu, G. Caocci, C. Di Ruberto, Leucocyte classification for leukaemia detection using image processing techniques, Artif. Intell. Med. 62 (3) (2014) 179–191.

[2] S.C. Neoh, W. Srisukkham, L. Zhang, S. Todryk, B. Greystoke, C.P. Lim, M.A. Hossain, N. Aslam, An intelligent decision support system for leukaemia diagnosis using microscopic blood images Scientific Reports, Nat. Pub. Group 5 (14938) (2015) 1–14.

[3] R.D. Labati, V. Piuri, F. Scotti, ALL-IDB: The acute lymphoblastic leukemia image database for image processing, in: Proceeding of IEEE International Conference on Image Processing, Brussels Belgium. IEEE, 2011, pp. 2045–2048 ((Database URL:) http://homes.di.unimi.it/scotti/all/.

[4] S. Agaian, M. Madhukar, T.A. Chronopoulos, Automated screening system for acute myelogenous leukemia detection in blood microscopic images, IEEE Syst. 8 (3) (2014) 995–1004.

[5] S. Mohapatra, D. Patra, S. Satpathy, An ensemble classifier system for early diagnosis of acute lymphoblastic leukemia in blood microscopic images, Neural Comput. Appl. 24 (2014) 1887–1904.

[6] H.T. Madhloom, S.A. Kareem, H. Ariffin, A robust feature extraction and selection method for the recognition of lymphocytes versus acute lymphoblastic leukemia, Proceedings of IEEE International Conference on Advanced Computer Science Applications and Technologies Malayia (2012) 330–335.

[7] V. Meera, S.A. Mathew, Fuzzy local information C means clustering for acute myelogenous leukemia image segmentation, Proceeding of the International Conference on Innovation and Advances in Science Engineering and Technology (2014) 61–68.

[8] G. Ongun, U. Halici, K. Leblebicioglu, V. Atalay, M. Beksac, S. Beksac, Feature extraction and classification of blood cells for an automated differential blood count system, Proceedings of IEEE International Joint Conference on Neural Networks, No. 4 (2001) 2461–2466.

[9] V. Singhal, P. Singh, Local binary pattern for automatic detection of acute lymphoblastic leukemia, Proceedings of IEEE the Twentieth National Conference on Communications (2014) 1–5.

[10] S.H. Rezatofighi, H. Soltanian-Zadeh, Automatic recognition of five types of white blood cells in peripheral blood, Comput. Med. Imaging Graph. 35 (4) (2011) 333–343.

[11] R. Sheikhpour, M.A. Sarram, R. Sheikhpour, Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer, Appl. Soft Comput. 40 (2016) 113–131.

[12] Y. Zhang, D. Gong, Y. Hu, W. Zhang, Feature selection algorithm based on bare bones particle swarm optimization, Neurocomputing 148 (2015) 150–157.

[13] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, A review of feature selection methods on synthetic data, Knowl. Inf. Syst. 34 (3) (2013) 483–519.

[14] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceeding of IEEE International Conference on Neural Network 4 (1995) 1942–1948.

[15] Y. Zhang, S. Wang, G. Ji, A comprehensive survey on particle swarm optimization algorithm and its applications, Math. Prob. Eng. 2015 (2015) 1–38 (Article ID 931256).

[16] K. Mistry, L. Zhang, S.C. Neoh, C.P. Lim, B. Fielding, A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition, IEEE Trans. Cybern. ISSN 2168–2267 (2016) 1–14.

[17] J. Kennedy, Bare bones particle swarms, Proceedings of the 2003 IEEE Swarm Intelligence Symposium (2003) 80–87.

[18] Y. Zhang, D.W. Gong, Z.H. Ding, A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch, Inf. Sci. 192 (2012) 213–227.

[19] Y. Zhang, D. Gong, N. Geng, X. Sun, Hybrid bare-bones PSO for dynamic economic dispatch with valve-point effects, Appl. Soft Comput. 18 (2014) 248–260.

[20] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceedings of World Congress on Nature & Biologically Inspired Computing, Coimbatore, 2009, pp. 210–214.

[21] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. (2015) 1–21.

[22] E. Valian, S. Tavakoli, S. Mohanna, A. Haghi, Improved cuckoo search for reliability optimization problems, Comput. Ind. Eng. 64 (1) (2013) 459–468.

[23] X. Li, M. Yin, Modified cuckoo search algorithm with self adaptive parameter method, Inf. Sci. 298 (2015) 80–97.

[24] A.R. Jordehi, Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimisation problems, Appl. Soft Comput. 26 (2015) 401–417.

[25] S.C. Neoh, L. Zhang, K. Mistry, M.A. Hossain, C.P. Lim, N. Aslam, P. Kinghorn, Intelligent facial emotion recognition using a layered encoding cascade optimization model, Appl. Soft Comput. 34 (2015) 72–93.

[26] D.C. Huang, K.D. Hung, Leukocyte nucleus segmentation and recognition in color blood-smear images, in: Proceedings of IEEE International Conference of Instrumentation and Measurement Technology, Graz., 2012, pp. 171–176.

[27] S. Osowski, R. Siroic, T. Markiewicz, K. Siwek, Application of support vector machine and genetic algorithm for improved blood cell recognition, IEEE Trans. Instrum. Measure. 58 (7) (2009) 2159–2168.

[28] H.J. Escalante, M. Montes-y-Gómez, J.A. González, P. Gómez-Gil, L. Altamirano, C.A. Reyes, A. Rosales, Acute leukemia classification by ensemble particle swarm model selection, Artif. Intell. Med. 55 (3) (2012) 163–175.

[29] Y. Chan, C. Wang, Y. Mao, W. Chang, K. Lin, An aneima abnormal red blood cells recognition system, 2nd Int Conf. Chem. Biol. Med. Sci., Phnom Penh, Cambodia (2013) 1–8.

[30] W. Srisukkham, P. Lepcha, A. Hossain, L. Zhang, R. Jiang, H.N. Lim, A mobile enabled intelligent scheme to identify blood cancer for remote areas − cell membrane segmentation using marker controlled watershed segmentation phase, Proceedings of 7th International Conference on Software Knowledge, Intelligent Management and Applications (2013) 104–114.

[31] S. Mohapatra, D. Patra, S. Satpathi, Image analysis of blood microscopic images for acute leukemia detection, Proceedings of IEEE International Conference on Industrial Electronics, Control & Robotics (2010) 215–219.

[32] Y. Zhang, L. Zhang, S.C. Neoh, K. Mistry, M.A. Hossain, Intelligent affect regression for bodily expressions using hybrid particle swarm optimization and adaptive ensembles, Expert Syst. Appl. 42 (22) (2015) 8678–8697.

[33] L. Zhang, K. Mistry, S.C. Neoh, C.P. Lim, Intelligent facial emotion recognition using moth-firefly optimization, Knowl. Based Syst. 111 (1) (2016) 248–267.

[34] C. Beyan, R. Fisher, Classifying imbalanced data sets using similarity based hierarchical decomposition, Pattern Recogn. 48 (2015) 1653–1672.

[35] H. Samma, C.P. Lim, J.M. Saleh, S.A. Suandi, A memetic-based fuzzy support vector machine model and its application to license plate recognition, Memetic Comput. 8 (3) (2016) 235–251.

[36] A. Khashman, H.H. Abbas, Acute lymphoblastic leukemia identification using blood smear images and a neural classifier. Advances in computational intelligence, 12th International Conference on Artificial Neural Networks (IWANN 2013), Part II, LNCS 7903 (Springer-Verlag) (2013) 80–87.

[37] L.Y. Chuang, C.S. Yang, K.C. Wu, C.H. Yang, Gene selection and classification using Taguchi chaotic binary particle swarm optimization, Expert Syst. Appl. 38 (2011) 13367–13377.

[38] C. Hsu, C. Chang, C. Lin, Practical Guide to Support Vector Classification, Department of Computer Science National, Taiwan University, 2010.

[39] Q.K. Pan, M.F. Tasgetiren, Y.C. Liang, A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem, Comput. Operat. Res. 35 (9) (2008) 2807–2839.

[40] M. Madhukar, S. Agaian, A.T. Chronopoulos, New decision support tool for acute lymphoblastic leukemia classification, in: International Conference on Image Processing: Algorithms and System X; and Parallel Processing for Imaging Application II, Burlingame, California, USA. SPIE, 2012.

[41] K. Bache, M. Lichman, UCI Machine Learning Repository, School Information Comput. Sci., Univ. California, Irvine, CA, USA, 2013 ([Online]. Available:) http://archive.ics.uci.edu/ml.

[42] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.

[43] S. Mirjalili, The ant lion optimizer, Adv. Eng. Software 83 (2015) 80–98.

[44] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, 4th ed., Chapman & Hall/CRC, 2006.

[45] D. Pandit, L. Zhang, C. Liu, S. Chattopadhyay, N. Aslam, C.P. Lim, A lightweight QRS detector for single lead ECG signals using a max–min difference algorithm, Comput. Methods Programs Biomed. (2017) (in press).

[46] A. Bourouis, M. Feham, M.A. Hossain, L. Zhang, An intelligent mobile based decision support system for retinal disease diagnosis, Decis. Support Syst. 59 (2014) 341–350.

[47] Y. Zhang, S. Wang, P. Phillips, Z. Dong, G. Ji, J. Yang, Detection of Alzheimer's disease and mild cognitive impairment based on structural volumetric MR images using 3D-DWT and WTA-KSVM trained by PSOTVAC, Biomed. Signal Process. Control 21 (2015) 58–73.