Received March 25, 2018, accepted May 1, 2018, date of publication May 16, 2019, date of current version October 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2881020

Comparative Research of Swarm Intelligence Clustering Algorithms for Analyzing Medical Data

XUEYUAN GONG[1], LIANSHENG LIU[2], SIMON FONG[1], QIWEN XU[1], TINGXI WEN[3], AND ZHIHUA LIU[3]

[1]Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China
[2]First Affiliated Hospital, Guangzhou University of Traditional Chinese Medicine, Guangzhou 510405, China
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China

Corresponding author: Simon Fong (ccfong@umac.mo)

This work was supported in part by the Macao Science and Technology Development Fund through the EAE Project under Grants 072/2009/A3 and MYRG2015-00128-FST, in part by the University of Macau, in part by the Macau SAR Government, in part by Shenzhen Basic Research under Grant JCYJ20150401145529007, and in part by the Shenzhen Technology Research under Grants CXZZ20151015163619907, JSGG20160229123927512, and KQJSCX20170331161941176.

ABSTRACT As the Internet of medical Things emerge in the field of medicine, the volume of medical data is expanding rapidly and along with its variety. As such, clustering is an important procedure to mine the vast data. Many swarm intelligence clustering algorithms, such as the particle swarm optimization (PSO), firefly, cuckoo, and bat, have been designed, which can be parallelized to the benefit of mass data computation. However, few studies focus on the systematic analysis of the time complexities, the effect of instances (data size), attributes (dimensionality), number of clusters, and agents of these algorithms. In this paper, we performed a comparative research for the PSO, firefly, cuckoo, and bat algorithms based on both synthetic and real medical data sets. Finally, we conclude which algorithms are effective for the medical data mining. In addition, we recommend the more suitable algorithms that have been developed recently for the different medical data to achieve the optimal clustering.

INDEX TERMS Medical data analysis, data mining, swarm intelligence, clustering algorithms.

I. INTRODUCTION

Clustering is a well-known problem in computer science. In recent years, scholars have applied swarm intelligence algorithms to solve the clustering problem. Some examples are the PSO clustering, Firefly clustering, Bat clustering, etc. Swarm intelligence algorithms are popular in the optimization community. The core idea of swarm intelligence algorithms is imitating behaviors of creatures in nature, especially creatures that have a habit of swarming together, e.g. ants, fireflies, bees, etc. Researchers believed that there are some underlying reasons for their behavior, such as searching for food, being together with companions, evading obstacles, etc. It is found that swarm intelligence clustering approaches have more possibilities to deviate from the local optima, and therefore it is useful to apply swarm intelligence algorithms to solve clustering problems. Up-to-date, different kinds of swarm intelligence algorithms have been applied to clustering problems [1]–[5].

In literature, Tang *et al.* [4] have compared the performance of several swarm intelligence clustering approaches. However, there is no systematic experiment and analysis on how instances (data size), attributes (dimensionality), number of clusters, and number of agents can affect the performance of all those approaches. Therefore, this gives us the motivation to analyze the time complexities of four swarm intelligence clustering approaches (PSO, Firefly, Cuckoo and Bat) systematically in this paper. Then, by conducting experiments on synthetic and real data, we also confirmed that the assumption of their time complexity is correct. The experiments on synthetic data were conducted based on four aspects: data size, dimensionality, number of clusters and number of agents. In addition, we conducted experiments on real data to further confirm that our assumption is correct.

The remainder of this paper is organized as follows: Related work of swarm intelligence algorithms and swarm intelligence clustering approaches are reviewed in Section 2. Next, preliminaries (i.e. notations, problem definition and fitness function) are introduced in Section 3. After that, four swarm intelligence clustering approaches are introduced in Section 4 and their time complexity is analyzed.

The associate editor coordinating the review of this manuscript and approving it for publication was Kelvin Wong.

Then, Section 5 provides the experiment results for analysis of the algorithms, while Section 6 concludes the paper and outlines future work.

II. RELATED WORKS

This section briefly reviews several swarm intelligence algorithms, literature that involves application of swarm intelligence algorithms to solve clustering problem, as well as articles comparing swarm intelligence clustering algorithms.

For the current optimization problems, it is difficult to search the optima when the search space is very large. Therefore, Kennedy and Eberhart [6] proposed Particle Swarm Optimization (PSO) to obtain an approximate optimum with partially searching the search space. In this way, it is highly efficient as it does not require searching the whole search space and its strategy ensures its accuracy is quite good. This was the first time that the strategy of a group of individuals was presented to the swarm intelligence community. Later on, Yang [7] proposed the Firefly algorithm by imitating the behavior of this insect. The basic idea is that one Firefly will be attracted by another. The attractiveness is defined to be proportional to their brightness, which is mathematically represented by the fitness in clustering problems. Subsequently, Yang and Deb [8] also proposed another swarm intelligence algorithm called the Cuckoo algorithm, which imitates the behavior of cuckoos laying eggs. In particular, each Cuckoo (agent) will lay an egg in a random nest and that egg will randomly be dumped or kept by the host of that nest in one generation. Furthermore, Yang [9] proposed the third swarm intelligence algorithm in 2010, called Bat algorithm, whereby the basic idea is imitating bats to sense distance by echolocation.

As various swarm intelligence algorithms were proposed, Van der Merwe and Engelbrecht [5] became the first to suggest clustering by PSO. To the best of our knowledge, his was the first paper proposed to adopt a swarm intelligence algorithm to solve the clustering problem. After that, Senthilnath *et al.* [3] proposed the Firefly clustering approach. Recently, Ameryan *et al.* [1] and Saida *et al.* [2] also proposed new clustering algorithms based on the Cuckoo algorithm. Tang *et al.* [4] has compared the performance of several swarm intelligence clustering algorithms in 2012. However, none of the above papers have systematically compared the time complexities of all four swarm intelligence clustering algorithms (pertaining to PSO, Firefly, Cuckoo, and Bat). Furthermore, none of the above papers have systematically analyzed the effect of data size, dimensionality, number of clusters and number of agents to all four swarm intelligence clustering algorithms.

III. PRELIMINARIES

A. NOTATIONS AND PROBLEM DEFINITION

The key terms as well as the problem investigated by this paper are defined in this section. First of all, the definition of an agent (based on a particle, Firefly, Bat, or Cuckoo) is given below:

Definition of Agent: An agent is a set of points in n-dimensional space, denoted A $= \{a_1, a_2, a_3, \ldots, a_n\}$. Each point $\mathrm{a}_i = \{x_1, x_2, x_3, \ldots, x_n\}$ is a n-dimensional vector, namely a point in n-dimensional space.

Note that a_i also represents i-th cluster from the perspective of clustering. Based on the definition of agent, the distance between the agent and a point in n-dimensional space is defined as follows:

Definition of Distance: The distance between the agent A and a point p in n-dimensional space is defined as $Dist\,(A, p) = \min(||a_1 - p||\,, ||a_2 - p||\,, \ldots, ||a_m - p||)$.

Note that a_i and p are both points in n-dimensional space. p is assigned to cluster a_i if $||a_i - p||$ is minimal for all $a_i \in A$. After the agent and distance are defined, we are in the position to define the problem of clustering.

Problem Definition: Given a set of points $P = \{p_1, p_2, \ldots, p_l\}$, the objective of clustering is to find an agent A which minimizes the equation $\sum_{i=1}^{l} Dist(A, p_i)$.

Therefore, as the objective is to find the agent A, which can minimize the equation $\sum_{i=1}^{l} Dist(A, p_i)$, we adopt PSO, Firefly, Bat and Cuckoo respectively to find the best agent A. Table 1 summarizes the above notations as follows.

TABLE 1. Definitions and notations.

Notation	Definition
A	An agent consisting of m n-dimensional points
a_i	One n-dimensional point contained in A
P	A set of l n-dimensional points
p_i	One n-dimensional point contained in P
$Dist(A, p)$	Distance between A and p

B. FITNESS FUNCTION

Fitness/objective function is essential in optimization problems. In this paper, swarm intelligence algorithms, which are used for optimization problems, are adopted to perform clustering. However, optimization problem and clustering problem are different. Thus, we transform the clustering problem into an optimization problem so that swarm intelligence algorithms can be easily implemented. The fitness function for clustering problems is defined as follows:

$$\boldsymbol{F}\,(\boldsymbol{A}) = \sum\nolimits_{i=1}^{l} \boldsymbol{Dist}(\boldsymbol{A}, \boldsymbol{p_i}) \tag{1}$$

Interestingly, Equation (1) is the objective of our problem definition, which represents that the clustering problem can be transformed into an optimization problem in a straightforward manner. Furthermore, the time complexity of Equation (1) is O(ml) as there are l points in total and each p_i is compared with all a_i according to the definition of $Dist(A, p_i)$.

IV. SWARM INTELLIGENCE CLUSTERING

A. PSO CLUSTERING

Particle Swarm Optimization (PSO) was firstly attributed to [6] and [8]. For applying PSO to clustering, given P,

Algorithm 1 PSO Clustering Algorithm

Input: A set of points $P = \{p_1, p_2, \ldots, p_l\}$ and three parameters w, c_1 and c_2
Output: An agent A with best fitness calculated by Equation (1)
Initialize $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$;
Calculate $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$ according to Equation (1);

Initialize $\mathcal{PA} = \{PA_1, PA_2, \ldots, PA_k\}$;
Calculate $\mathcal{PF} = \{PF_1, PF_2, \ldots, PF_k\}$;
Initialize GA;
Calculate GF;
Initialize $\mathcal{V} = \{V_1, V_2, \ldots, V_k\}$;
For before stop criterion meets **do**
  For each A_i **do**
    Update A_i by Equation (2);
    Calculate F_i according to Equation (1);
    If $F_i < PF_i$ **then**
      $PA_i = A_i$;
      $PF_i = F_i$
    End
    If $PF_i < GF$ **then**
      $GA = PA$;
      $GF = PF_i$;
    End
  End
End

Algorithm 2 Firefly Clustering Algorithm

Input: A set of points $P = \{p_1, p_2, \ldots, p_l\}$ and three parameters α, δ and γ
Output: An agent A with best fitness calculated by Equation (1)
For before stop criterion meets **do**
  Calculate all F_i of $\mathcal{F}$ according to Equation (1);
  For each A_i **do**
    For each A_j except A_i **do**
      If $F_i > F_j$ **then**
        Update A_i according to Equation (3);
      End
    End
  End
  $\alpha = \alpha\delta$;
End

the first step is to initialize a set of agents $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$. After that, the fitness value of all agents are calculated by Equation (1), which is denoted $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$. In addition, there is another set $\mathcal{V} = \{V_1, V_2, \ldots, V_k\}$ to store the velocity of $\mathcal{A}$, where $V_i = \{v_1^i, v_2^i, \ldots, v_m^i\}$ and each $v_j^i = (x_1, x_2, \ldots, x_n)$. Then, the location of agents is updated based on the previous best agent of itself PA_i and the global best agent of all agent GA. Note that the agent in PSO is a n-dimensional point, but the agent outlined in this paper is a set of n-dimensional points. Therefore, when updating one agent, all the points in this agent will be updated accordingly. As an example, A_i is going to be updated based on PA_i and GA. The equation for this is given below:

$$\begin{aligned} a_j^i &= a_j^i + v_j^i \\ \boldsymbol{v_j^i} &= \boldsymbol{wv_j^i + c_1 r(pa_j^i - a_j^i) + c_2 r(ga^i - a_j^i)} \end{aligned} \tag{2}$$

where w is a weight parameter set by the user and r is a random number that is subject to a uniform distribution, denoted as $r \sim U(0, 1)$. Furthermore, the time complexity of Equation (2) is O(mn) as the size of each A_i is m and the dimensionality of each a_j^i is n, where m represents the number of clusters.

The pseudo code is shown using Algorithm 1. As the time complexity of Equation (1) is O(ml) and Equation (2) is O(mn), the time complexity of updating each A_i is O(ml+mn). After that, the size of $\mathcal{A}$ is k and thus the time complexity of PSO for one generation is O($k(ml$+$mn)$), where k represents the number of agents.

B. FIREFLY CLUSTERING

The Firefly algorithm was proposed by Yang, which simulates the behavior of the Firefly for searching the optima in a search space. Given are P and three parameters α, δ and γ, where α is the randomness of each agent, δ is the randomness reduction rate and γ is the absorption coefficient. Firstly, initialize a set of agents $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$. Then, the fitness of all agents $\mathcal{F}$ is calculated in advance. After that, the location of each agent A_i is affected by all other better agents and updated accordingly. For example, the location of A_i is waited to be updated. Suppose that by comparing fitness, A_x and A_y are found to be better than A_i. Afterwards, A_i will firstly move towards A_x based on Equation (3) and then move towards A_y based on Equation (3). Suppose A_i is moving towards A_x, the movement equation of each a_j^i of A_i is given below:

$$\begin{aligned} a_j^i &= a_j^i + de^{-\gamma d^2} + \alpha r \\ \boldsymbol{d} &= \boldsymbol{a_j^x - a_j^i} \end{aligned} \tag{3}$$

where γ and α are parameters given by the user, and r is a random number such that $r \sim U(-1, 1)$. The time complexity of Equation (3) is O(mn) as the size of each A_i is m and the dimensionality of each a_j^i in A_i is n.

The pseudo code of Firefly clustering is given in Algorithm 2. As each A_i of $\mathcal{A}$ will be compared with all other A_j of $\mathcal{A}$, the time complexity to compute this is O(k^2), where k denotes the number of agents. Besides, Equation (3) may be calculated after comparing A_i and A_j; therefore, the time complexity is O(mnk^2). Finally, the cost to calculate all F_i of $\mathcal{F}$ in advance is O(mlk). Therefore, the time complexity of Firefly clustering for one generation is O($mlk + mnk^2$).

Algorithm 3 Firefly Clustering Algorithm

Input: A set of points $P = \{p_1, p_2, \ldots, p_l\}$ and a parameter p_a
Output: An agent A with best fitness calculated by Equation (1)
Initialize $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$;
Calculate $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$ according to Equation (1);
Find the minimum fitness F_{min} of $\mathcal{F}$ and its corresponding agent A_{min} of $\mathcal{A}$;
Initialize TA as temporary A and TF as temporary F;
For before stop criterion meets **do**
  For each A_i **do**
    Update A_i according to Equation (4) and store into TA;
    Calculate TF according to TA by Equation (1);
    If $TF < F_i$ **then**
      Assign TA and TF to A_i and F_i;
    End
  End
  Find the minimum fitness F_{min} of $\mathcal{F}$ and its corresponding agent A_{min} of $\mathcal{A}$;
  For each A_i **do**
    Generate a random number r;
    If $r > p_a$ **then**
      Get a randomly chosen A_{rnd} in $\mathcal{A}$ and store into TA;
    End
    Calculate TA according to TA by Equation (1);
    If $TF < F_i$ **then**
      Assign TA and TF to A_i and F_i;
    End
  End
  Find the minimum fitness F_{min} of $\mathcal{F}$ and its corresponding agent A_{min} of $\mathcal{A}$;
End

C. FIREFLY CLUSTERING

Yang proposed the Cuckoo algorithm in 2009. The Cuckoo algorithm searches the optima by simulating the behavior of a Cuckoo laying eggs. Given are P and a parameter p_a, where p_a is the probability to abandon an old agent. First of all, a set of agents $\mathcal{A}$ is initialized and then the corresponding fitness $\mathcal{F}$ of $\mathcal{A}$ is calculated. Besides, the minimum fitness F_{min} of $\mathcal{F}$ and its corresponding agent A_{min} are recorded. Next, there are two steps in one generation to update the location of agents. The pseudo code of the Cuckoo clustering is given in Algorithm 3.

The first step is updating the location of agents via Levy flights using the Mantegna's algorithm. Suppose A_i is going to be updated, the equation to update every a_j^i of A_i is given in Equation (4).

$$\boldsymbol{a_j^i = a_j^i + 0.01rs(a_j^{min} - a_j^i)} \tag{4}$$

where $r \sim U(0, 1)$ is a random number and s is the step size calculated by Mantegna's algorithm. In Mantegna's algorithm, step size can be calculated as below:

$$s = \frac{u}{|v|^{1/\beta}}$$

$$\boldsymbol{u \sim U(0, \sigma^2) \quad v \sim N(0, 1)} \tag{5}$$

where σ is a parameter calculated via Levy flights and $\beta = 3/2$ by default for Cuckoo search. The equation of Levy flights to calculate σ is given below:

$$\sigma = \{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta 2^{(\beta-1)/2}}\}^{1/\beta}$$
$$\boldsymbol{\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt} \tag{6}$$

where β is set to $3/2$ by default for Cuckoo search. Besides, the time complexity to update one agent of Cuckoo clustering is O(mn), where m is the number of clusters and n is the dimensionality. After the location of A_i is updated, the fitness of A_i should also be calculated, which costs O(ml), where l represents the size of data. Moreover, the size of A_i is k and thus the total time complexity of step one for one generation is O($k(ml+mn)$).

The second step is randomly replacing some agents by other random agents based on the probability p_a. The time complexity of replacing is O(l) as the only calculation is determination and replacement. Additionally, the time complexity to calculate fitness of A_i is O(ml). Moreover, the size of A_i is k. Therefore, the time complexity of step two for one generation is O(mlk), where O(l) is ignored as it is dominated by O(mlk).

In general, the total time complexity of Cuckoo clustering for one generation is the sum of these two steps. Besides, there is a calculation of finding the minimum fitness F_{min} of $\mathcal{F}$ and its corresponding agent A_{min} of $\mathcal{A}$ followed by each step, whose time complexity is O(k). Thus, the time complexity is O($k(ml+mn)+mlk+2k$).

D. BAT CLUSTERING

The Bat algorithm was again proposed by Yang. It searches the optima in search space by simulating bats that sense distance via echolocation. Given P and four parameters ld, pr, fq_{min} and fq_{max}, where ld indicates loudness, pr indicates pulse rate, fq_{min} and fq_{max} represent the domain of frequency. Firstly, a set of agents $\mathcal{A}$ and the corresponding fitness of $\mathcal{A}$ ($\mathcal{F}$) are initialized. In addition, the minimum fitness F_{min} and its corresponding agent A_{min} are recorded. Then, the velocity of all agents $\mathcal{V}$ is also initialized. After that, the location of agents can be updated by the Bat clustering algorithm. Suppose that A_i is going to be updated, then every a_j^i can be updated according to Equation (7).

$$a_j^i = a_j^i + v_j^i$$
$$\boldsymbol{v_j^i = v_j^i + fq(a_j^{min} - a_j^i)} \tag{7}$$

where fq is a random number which is subject to $fq \sim U(fq_{min}, fq_{max})$. The time complexity of Equation (7) is (mn) as there are ma_j^i in total and dimensionality n. Additionally, the location of agents have probability to be set to a position around A_{min} directly. The equation is given below:

$$\boldsymbol{a_j^i = a_j^{min} + 0.001r} \tag{8}$$

Algorithm 4 Bat Clustering Algorithm

Input: A set of points $P = \{p_1, p_2, \ldots, p_l\}$ and four parameters ld, pr, fq_{min} and fq_{max}
Output: An agent A with best fitness calculated by Equation (1)
Initialize $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$;
Calculate $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$ according to Equation (1);
Find the minimum fitness F_{min} of $\mathcal{F}$ and its corresponding agent A_{min} of $\mathcal{A}$;
Initialize $\mathcal{V} = \{V_1, V_2, \ldots, V_k\}$;
Initialize TA as temporary A and TF as temporary F;
For before stop criterion meets **do**
  For each A_i **do**
    Generate a random number $fq \sim U(fq_{min}, fq_{max})$;
    Calculate the updated A_i according to Equation (7) and assign to TA;
    Generate a random number $r1 \sim N(0, 1)$;
    If $r1 > pr$ **then**
      Set TA_i by Equation (8);
    End
    Calculate TF according to TA by Equation (1);
    Generate a random number $r2 \sim N(0, 1)$;
    If $TF < F_i$ and $r2 < ld$ **then**
      Assign TA and TF to A_i and F_i;
    End
  End
  Find the minimum fitness F_{min} of $\mathcal{F}$ and its corresponding agent A_{min} of $\mathcal{A}$;
End

where $r \sim N(0, 1)$ is a random number. The time complexity of Equation (8) is also O(mn) as it depends on the number of a^i_j (number of clusters) and dimensionality.

The pseudo code of Bat clustering is shown in Algorithm 4. After location updating and replacement, calculating TF costs O(ml), where l is the size of data points. Thus, the time complexity to update one A_i is ($2mn+ml$). Therefore, the time complexity to update $\mathcal{A}$ is O($k(2mn+ml)$). Finally, the step to find the minimum fitness F_{min} and its corresponding agent A_{min} costs O(k). Therefore, the total time complexity of Bat clustering for one generation is O($k(2mn+ml)+k$).

E. TIME COMPLEXITY ANALYSIS

After all four approaches have been introduced, we conclude their time complexities using Table 2. By analyzing the time complexities of these four clustering approaches, we can conclude that the number of clusters m and the number of agents k affect the efficiency most as they are outside the parenthesis and will multiply all components in the parenthesis. Besides, Cuckoo clustering is the slowest as it contains more components compared to other approaches. Firefly clustering would be also slow if k is large because it is $(l+k)$ in the parenthesis rather than $(l+1)$ for others, where k represents the number of agents. Lastly, PSO and Bat are relatively faster than the other two approaches.

TABLE 2. Time complexity of four clustering approaches.

Approach	Time Complexity
PSO	O($mk(l+n)$)
Firefly	O($mk(l+nk)$)
Cuckoo	O($mk(2l+n)+2k$)
Bat	O($mk(l+2n)+k$)

V. EXPERIMENT RESULT

In this section, parameters for every clustering algorithms are introduced in the first place. Next, the experiments are conducted on synthetic data for comparing the efficiency and effectiveness of four approaches. The synthetic data are scaled from four aspects (data size l, dimensionality n, number of clusters m and number of agents k) so as to compare different approaches from different perspectives. Afterwards, we also conduct the experiments based on real data sets to show our experiments on synthetic data are reasonable. Finally, six medical data sets are tested as case studies. In this paper, all our experiments were conducted on a computer with an Intel Xeon E5-1650 CPU at 3.5GHz, with 64 GB memory. The operating system was Windows 7 and programming language is Matlab with development environment of Matlab 2014a.

A. PARAMETER SET

The parameters of all algorithms are set to the default values as shown in Table 3. If not specifically mentioned otherwise, all experiments are implied to be based on the parameter settings in this table.

TABLE 3. Parameter set.

PSO		Firefly		Cuckoo		Bat	
w	0.7	α	0.6	p_a	0.25	ld	0.5
c_1	1.5	γ	0.3	k	16	pr	0.5
c_2	1.5	δ	0.97			fq_{min}	0
k	16	k	16			fq_{max}	2
						k	16

B. CALCULATE ACCURACY OF CLUSTERING ALGORITHM

The accuracy of clustering algorithm is represented by their purity. It is briefly introduced in this section. Given the best agent chosen by clustering algorithm A $= \{a_1, a_2, \ldots, a_n\}$. Note it also represents the set of clusters corresponding to the set of data P. Suppose the set of classes corresponding to P is C $= \{c_1, c_2, \ldots, c_n\}$. We interpret a_i and c_j as the set containing all the points $p_k \in P$ which are assigned to a_i and c_j. Then we can calculate the purity of the clustering result by Equation (9).

$$\boldsymbol{Purity}(\boldsymbol{A}, \boldsymbol{C}) = \frac{\mathbf{1}}{\boldsymbol{l}} \sum_{\boldsymbol{i}} \boldsymbol{max}_{\boldsymbol{j}} |\boldsymbol{a_i} \cap \boldsymbol{c_j}| \tag{9}$$

Where $l = |P|$, $1 \leq i \leq m$ and $1 \leq j \leq x$. $|a_i \cap c_j|$ represents the number of $p_k \in P$ which belong to cluster a_i and class c_j at the same time. For example, Figure 1 shows two clusters which have two classes. Then, the summation part of Equation (9) is max $(6, 2, 2)$ + max $(7, 1, 2)$ + max$(8, 1, 1)$. Finally, the purity is $(6 + 7 + 8)/30 = 0.7$.

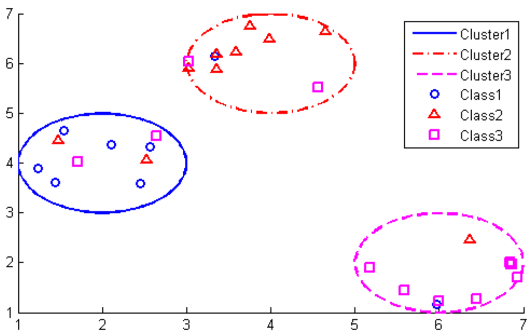


FIGURE 1. Example of calculating purity.

C. TEST ON SYNTHETIC DATA

The synthetic data are generated by uniformly putting the clusters of data into the search space. Figure 2 provides an example of synthetic data whose data size is 800, dimensionality is 3, and number of clusters is 8. In this experiment, the parameters of data and algorithm vary from data size and dimensionality to number of clusters and number of agents to test the scalability of the four algorithms. By default, the data size is 10, dimensionality is 2, number of clusters is 2, and number of agents is 16. Then, we increase the data size, dimensionality, number of clusters and number of agents respectively to compare the purity and execution time of the four clustering approaches (PSO, Firefly, Cuckoo and Bat). For correctness, each clustering approach is run ten times and we calculate the average and standard deviation of results to show its stability.

Firstly, the results of increasing the data size are shown in Table 4 and Figure 3. For purity, there is no significant change for the four approaches except a small drop when the data size is 10^4. That is, data size affects the purity of the clustering approaches but not very significantly. For execution time, all the clustering approaches are increasing gradually. This is because data size l is in the parenthesis for all four clustering approaches so that it affects the efficiency insignificantly. Besides, Cuckoo clustering is the slowest. This result is reasonable according to the analysis of time complexity in Section 4.5.

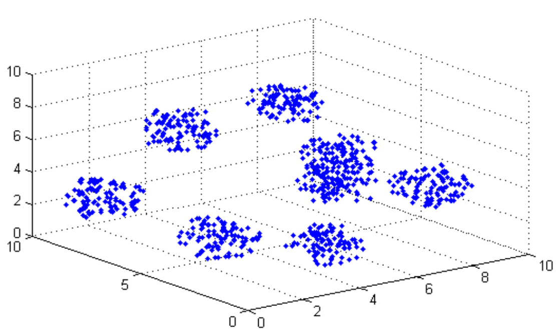


FIGURE 2. Example of the generated synthetic data.

TABLE 4. Results on the synthetic data when data size is different.

Approach	Data Size			
	10	10^2	10^3	10^4
Purity(%)				
PSO	98.5 $\pm$ 2.29	100 $\pm$ 0	100 $\pm$ 0	99 $\pm$ 0.13
Firefly	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	99 $\pm$ 0.01
Cuckoo	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	99 $\pm$ 0.02
Bat	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	99 $\pm$ 0.04
Time(sec)				
PSO	0.12 $\pm$ 0.01	0.74 $\pm$ 0.02	6.83 $\pm$ 0.11	67.19 $\pm$ 0.39
Firefly	0.1 $\pm$ 0.01	0.73 $\pm$ 0.02	6.83 $\pm$ 0.08	67.77 $\pm$ 0.37
Cuckoo	0.24 $\pm$ 0.02	1.99 $\pm$ 0.07	19.76 $\pm$ 0.09	195.3 $\pm$ 0.57
Bat	0.08 $\pm$ 0.01	0.71 $\pm$ 0.03	6.94 $\pm$ 0.11	68.48 $\pm$ 0.38

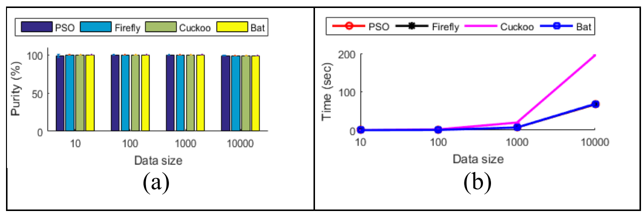


FIGURE 3. Results on the synthetic data when data size is different. (a) Purity. (b) Time.

Secondly, the results of increasing dimensionality of data are shown in Table 5 and Figure 4. For purity, there is no obvious decrease when dimensionality increases. However, the results of clustering approaches are not stable except Cuckoo, as the Cuckoo is much more stable in comparison to the other approaches when dimensionality increases, although it is the slowest. For execution time, the result is similar to increasing data size, where the execution time of all approaches increases gradually, as the dimensionality n is also inside the parenthesis.

Thirdly, the results of increasing number of clusters are shown in Table 6 and Figure 5. For purity, four clustering approaches tend to fluctuate, which represents that the number of clusters will affect the purity of clustering approaches but there is no obvious increasing or decreasing effect. For execution time, it increases dramatically along with the increasing of number of clusters. This result is expected as

TABLE 5. Results on the synthetic data when dimensionality of data is different.

Approach	Data Size			
	16	128	1024	16384
Purity(%)				
PSO	97 ± 6.4	99.5 ± 1.5	100 ± 0	100 ± 0
Firefly	100 ± 0	94.5 ± 7.57	98 ± 3.32	97 ± 5.1
Cuckoo	100 ± 0	100 ± 0	100 ± 0	100 ± 0
Bat	100 ± 0	100 ± 0	99 ± 2	96 ± 4.36
Time(sec)				
PSO	0.13 ± 0.01	0.18 ± 0.01	0.62 ± 0.02	8.92 ± 0.09
Firefly	0.1 ± 0.01	0.15 ± 0.02	0.44 ± 0.03	6.29 ± 0.07
Cuckoo	0.24 ± 0.02	0.32 ± 0.02	0.94 ± 0.04	12.07 ± 0.1
Bat	0.09 ± 0.01	0.12 ± 0.01	0.28 ± 0.02	3.87 ± 0.03

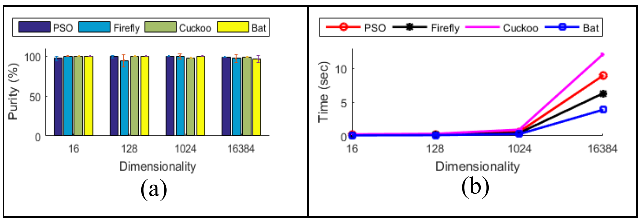


FIGURE 4. Results on the synthetic data when dimensionality of data is different. (a) Purity. (b) Time.

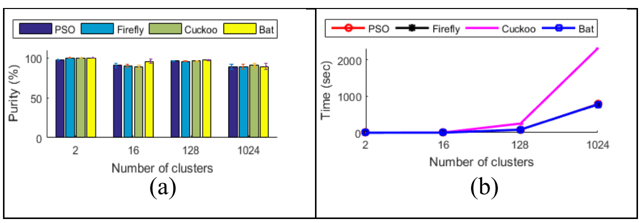


FIGURE 5. Results on synthetic data when number of clusters is different. (a) Purity. (b) Time.

TABLE 6. Results on synthetic data when number of clusters is different.

Approach	Data Size			
	2	16	128	1024
Purity(%)				
PSO	97.5 ± 0.03	90.8 ± 2.68	95.96 ± 0.84	89.2 ± 3.21
Firefly	100 ± 0	89.94 ± 2.6	95.78 ± 1	88.3 ± 3.32
Cuckoo	100 ± 0	88.3 ± 2.88	96 ± 0.8	90.6 ± 2.94
Bat	100 ± 0	95.56 ± 2.77	97.17 ± 0.6	88.9 ± 3.7
Time(sec)				
PSO	0.12 ± 0.01	1.93 ± 0.04	84.5 ± 0.56	783.4 ± 3.4
Firefly	0.12 ± 0.02	1.84 ± 0.06	84 ± 0.34	775.31 ± 4.2
Cuckoo	0.24 ± 0.02	5.03 ± 0.05	251.8 ± 0.6	2317.7 ± 5.7
Bat	0.09 ± 0.02	1.72 ± 0.05	84.2 ± 0.2	780.1 ± 3.8

TABLE 7. Results on the synthetic data when number of agents is different.

Approach	Data Size			
	2	16	128	1024
Purity(%)				
PSO	97.5 ± 0.1	93.8 ± 0.45	97.64 ± 0.41	96.19 ± 0.23
Firefly	96.5 ± 0.14	95.3 ± 0.6	93.72 ± 0.5	95.96 ± 0.46
Cuckoo	98.1 ± 0.13	96.9 ± 0.58	95.19 ± 0.64	97.32 ± 0.94
Bat	95.4 ± 0.17	94.7 ± 0.77	98.47 ± 0.43	96.23 ± 0.57
Time(sec)				
PSO	0.46 ± 0.03	3.53 ± 0.08	27.54 ± 0.35	221.14 ± 3.7
Firefly	0.42 ± 0.01	3.4 ± 0.07	44.52 ± 0.47	1531.2 ± 5.6
Cuckoo	1.24 ± 0.02	9.25 ± 0.08	71.94 ± 0.79	578.33 ± 4.7
Bat	0.43 ± 0.02	3.15 ± 0.06	24.4 ± 0.34	198.11 ± 3.2

the number of clusters m is outside the parenthesis according to Section 4.5. In other words, the number of clusters m affects the execution time more than dimensionality and data size.

Finally, Table 7 and Figure 6 demonstrate the results of increasing numbers of agents. For purity, there is no notable difference when increasing the number of agents as the data size is not big, so that a few agents are sufficient to achieve clustering. For execution time, Cuckoo is still very slow which is similar to other experiments on synthetic data. However, the execution time of Firefly increases dramatically this time when the number of agents increases. That is to be expected as the number of agents k is both outside and inside the parenthesis, which means the value of k affects the efficiency of Firefly significantly.

Based on these four experiments on synthetic data sets, we can conclude that there is no significant difference between these four approaches regarding purity. By analyzing the time complexities of the four approaches in our experiments, we can conclude that Cuckoo clustering is slowest among all four approaches. Firefly is very sensitive to the number of agents. PSO and Bat are relatively faster. In addition, the number of clusters and number of agents affect the efficiency of the four approaches the most. It is not acceptable

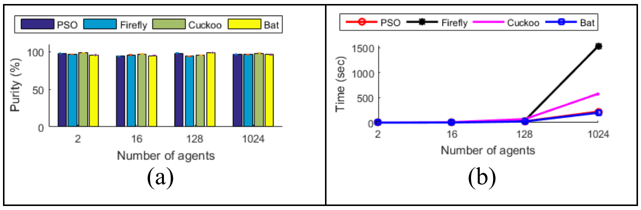


FIGURE 6. Results on the synthetic data when number of agents is different. (a) Purity. (b) Time.

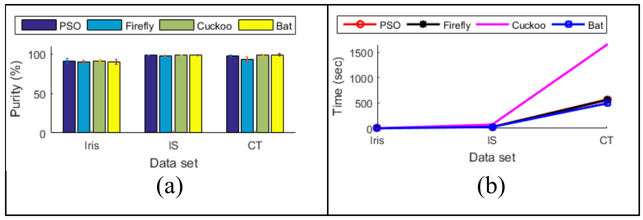


FIGURE 7. Results on real data sets. (a) Purity. (b) Time.

as it costs hours or maybe days to run when the number of clusters and number of agents reaches 10^5.

D. TEST ON REAL DATA

We also compared the four clustering approaches on three real data sets for further confirming of our conclusions. They are the Iris data set, Image Segmentation (IS) data set and Character Trajectories (CT) data set, respectively. Their descriptions are given below.

1) IRIS DATA SET

The Iris data set was first created by Fisher [11], and is widely used in the classification and clustering community as it is simple, clear, and proposed long ago. It contains 150 instances (data size) and 4 attributes (dimensionality) with 3 classes. The attributes represent sepal length, sepal width, petal length and petal width. This data set is adopted as a simple tester for four approaches. The Iris data set can be downloaded from [3].

2) IS DATA SET

The Image Segmentation (IS) data set was created by the vision group at the University of Massachusetts. This data set contains 2,310 instances, 19 attributes and 7 classes. The attributes are 19 features extracted from the image, e.g. the column of the center pixel of the region, the number of the center pixel of the region, etc. The IS data set can be downloaded from [2].

3) CT DATA SET

The Character Trajectories (CT) data set was created by Williams *et al.* [14]. It has 2,858 instances, 615 attributes and 20 clusters. The CT data set originally contained only one attribute, which is a 3 by 205 matrix. Each column of the matrix represents a feature (they are x-axis value, y-axis value and force of the pen). We vectorize this matrix to a vector with length 615 so that it can be conveniently transformed into an instance for clustering. The CT data set can be downloaded from [1].

The results of the four clustering approaches on the three real data sets are shown in Table 8 and Figure 7. The results in Table 8 are as expected, being similar to the results in Section 5.3. For purity, the four approaches are similar on all three data sets, except that Firefly appears to be slightly weaker (92.9%) compared to other three approaches (98.2% on average). For execution time, Cuckoo is still the slowest while the other three approaches are similar on all three data sets. As a systematical discussion of the performance (effectiveness and efficiency) of four algorithms was given in Section 5.3, the objective of our experiments on real data is validating the assumption and the detailed discussion is therefore omitted.

TABLE 8. Results on the real data sets.

Approach	Data Size		
	Iris	IS	CT
Purity(%)			
PSO	91.3 ± 3.15	98.32 ± 0.2	97.37 ± 1.42
Firefly	90.1 ± 2.34	97.63 ± 0.3	92.9 ± 3.5
Cuckoo	90.7 ± 2.74	98.15 ± 0.1	98.46 ± 1.72
Bat	89.7 ± 3.03	98.4 ± 0.3	98.78 ± 1.65
Time(sec)			
PSO	066 ± 0.04	27.33 ± 1.42	568.41 ± 2.57
Firefly	0.65 ± 0.03	27.07 ± 1.31	557.38 ± 2.39
Cuckoo	1.75 ± 0.02	73.92 ± 2.1	1648.9 ± 6.38
Bat	0.63 ± 0.02	27.38 ± 1.37	491.82 ± 2.72

E. CASE STUDY ON MEDICAL DATA SETS

In this section, we analyzed 6 medical databases as case studies using the sEMG for Basic Hand Movements (sEMG) data set [16], Arrhythmia data set [7], Mice Protein Expression (MPE) data set [20], Heart Disease (HD) data set [5], Arcene data set [10] and Dorothea data set [10]. The description of each data set is given below:

sEMG Data Set: The sEMG for Basic Hand Movements (sEMG) data set [16] contains 900 instances, 6000 attributes and 6 classes from 5 healthy subjects (based on three females and two males). The 6 classes refer to six kinds of hand grasps data, which are holding spherical tools, holding small tools, grasping with palm facing the object, holding thin, flat objects, holding cylindrical tools and supporting a heavy load respectively.

Arrhythmia Data Set: The Arrhythmia data set [7] has 452 instances, 279 attributes and 16 classes. Among the 16 classes, class 1 represents normal, classes 2 to 15 refer to different classes of arrhythmia and class 16 means unclassified ones.

MPE Data Set: MPE data set [20] contains 1080 instances, 77 attributes and 8 classes. The 8 classes are c-CS-s, c-CS-m, c-SC-s, c-SC-m, t-CS-s, t-CS-m, t-SC-s and t-SC-m, where

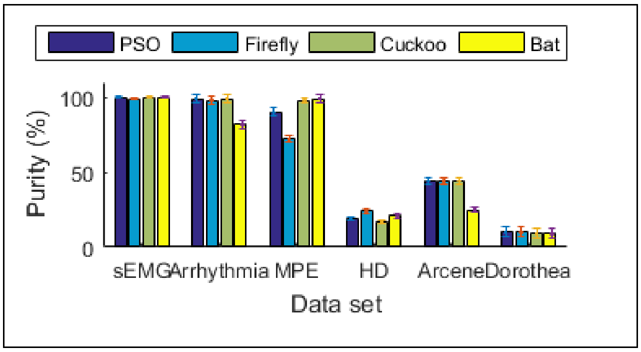


FIGURE 8. Purity on the six medical data sets.

TABLE 9. Purity on medical data sets.

Data Set	Approach			
	PSO	Firefly	Cuckoo	Bat
sEMG	100 $\pm$ 0.5	99 $\pm$ 0.3	100 $\pm$ 0.7	100 $\pm$ 0.3
Arrhythmia	99 $\pm$ 2.5	98 $\pm$ 2.3	99 $\pm$ 2.6	82 $\pm$ 2.5
MPE	90 $\pm$ 2.7	72 $\pm$ 2.2	98 $\pm$ 1.9	99 $\pm$ 2.4
HD	19 $\pm$ 1.1	24 $\pm$ 1.3	17 $\pm$ 1.1	21 $\pm$ 1.4
Arcene	44 $\pm$ 2.1	44 $\pm$ 2.2	44 $\pm$ 2.2	25 $\pm$ 2
Dorothea	10 $\pm$ 3.2	10 $\pm$ 3.3	9 $\pm$ 3.1	9 $\pm$ 3.1

c and t represent control mice and trisomy mice respectively, CS and SC mean stimulated to learn and not stimulated to learn respectively, s and m represent injected with saline and injected with memantine respectively.

HD Data Set: HD data set [5] contains 303 instances, 13 attributes, and 5 classes. Among the 5 classes, 0 represents absence of heart disease and 1, 2, 3, 4 represents presence of heart disease.

Arcene Data Set: Arcene data set [10] has 100 instances, 10000 attributes and 2 classes.

Dorothea Data Set: Dorothea data set [10] has 350 instances, 4857 attributes and 2 classes.

The purity on 6 medical data sets are given in Table 9 and Figure 8. As shown in Table 9, the purity on sEMG, Arrhythmia and MPE are quick good (around 99% averagely) while HD, Arcene and Dorothea are relatively low (around 20% averagely). This result illustrates that the swarm intelligence algorithms cannot be applied to all data sets. We can conclude that even though PSO, Firefly, Cuckoo and Bat can have good performance on some data sets (e.g. Iris, IS, CT, sEMG, Arrhythmia, MPE, etc), but they are not universal solution to all problems. Thus, it is required to consider whether the algorithm is suitable to solve a specific problem.

VI. CONCLUTION

In this paper, we introduced four main clustering approaches, which are based on swarm intelligence, and analyzed their time complexities. Our analysis showed that the Cuckoo clustering is the slowest one. Firefly clustering is slow when the number of agents is large. In comparison, the PSO and Bat are relatively faster than the other two approaches. After that, we conducted experiments on synthetic data by considering four aspects (data size, dimensionality, number of clusters and number of agents) to demonstrate our assumption, while we also conducted experiments on three real data sets to further confirm our assumption. Besides the conclusion on efficiency, we also conclude that there is no significant difference for these four clustering approaches on purity based on the experimental results using both synthetic data and real data.

In future, we aim to propose a new clustering algorithm based on swarm intelligence as the execution time of these four existing approaches is still not acceptable. Moreover, we are going to compare newly developed state-of-the-art approaches rather than just four classic swarm intelligence algorithms.

ACKNOWLEDGEMENT

(Xueyuan Gong and Liansheng Liu contributed equally to this work.)

REFERENCES

[1] M. Ameryan, M. R. A. Totonchi, and S. J. S. Mahdavi, "Clustering based on cuckoo optimization algorithm," in *Proc. Iranian Conf. Intell. Syst. (ICIS)*, Feb. 2014, pp. 1–6.
[2] I. B. Saida, K. Nadjet, and B. Omar, "A new algorithm for data clustering based on cuckoo search optimization," in *Genetic and Evolutionary Computing*. 2014, pp. 55–64.
[3] J. Senthilnath, S. N. Omkar, and V. Mani, "Clustering using firefly algorithm: Performance study," *Swarm Evol. Comput.*, vol. 1, no. 3, pp. 164–171, 2011.
[4] R. Tang, S. Fong, X.-S. Yang, and S. Deb, "Integrating nature-inspired optimization algorithms to k-means clustering," in *Proc. 7th Int. Conf. Digit. Inf. Manage. (ICDIM)*, Aug. 2012, pp. 116–123.
[5] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 1, Dec. 2003, pp. 215–220.
[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.
[7] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, 2008.
[8] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congr. Nature Biologically Inspired Comput. (NaBIC)*, Dec. 2009, pp. 210–214.
[9] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization—NICSO*, 2010, pp. 65–74.
[10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1998, pp. 69–73.
[11] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
[12] *Iris Data Set*. Accessed: Sep. 8, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Iris
[13] *Image Segmentation Data Set*. Accessed: Sep. 8, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Image+Segmentation
[14] B. H. Williams, M. Toussaint, and A. J. Storkey, "A primitive based generative model to infer timing information in unpartitioned handwriting data," in *Proc. IJCAI*, 2007, pp. 1119–1124.
[15] *Character Trajectories Data Set*. Accessed: Sep. 8, 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Character+Trajectories
[16] C. Sapsanis, G. Georgoulas, A. Tzes, and D. Lymberopoulos, "Improving EMG based classification of basic hand movements using EMD," in *Proc. 35th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2013, pp. 5754–5757.
[17] H. A. Güvenir, B. Acar, G. Demiröz, and A. Çekin, "A supervised machine learning algorithm for arrhythmia analysis," in *Proc. Comput. Cardiol.*, Sep. 1997, pp. 433–436.

[18] C. Higuera, K. J. Gardiner, and K. J. Cios, ''Self-organizing feature maps identify proteins critical to learning in a mouse model of down syndrome,'' *PLoS ONE*, vol. 10, no. 6, p. e0129126, 2015.

[19] R. Detrano *et al.*, ''International application of a new probability algorithm for the diagnosis of coronary artery disease,'' *Amer. J. Cardiol.*, vol. 64, no. 5, pp. 304–310, 1989.

[20] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, ''Result analysis of the NIPS 2003 feature selection challenge,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 545–552.

[21] A. A. A. Esmin, R. A. Coelho, and S. Matwin, ''A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data,'' *Artif. Intell. Rev.*, vol. 44, no. 1, pp. 23–45, 2015.

[22] T. Küçükdeniz and S. Esnaf, ''Data clustering by particle swarm optimization with the focal particles,'' in *Machine Learning, Optimization, and Big Data*. 2015, pp. 280–292.

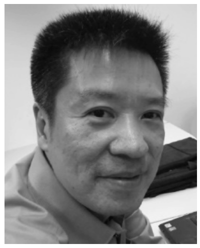
[23] A. M. Shanghooshabad and M. S. Abadeh, ''Robust medical data mining using a clustering and swarm-based framework,'' *Int. J. Data Mining Bioinf.*, vol. 14, no. 1, pp. 22–39, 2016.

[24] R. S. Kumar and G. T. Arasu, ''Modified particle swarm optimization based adaptive fuzzy k-modes clustering for heterogeneous medical databases,'' *J. Sci. Ind. Res.*, vol. 74, no. 1, pp. 19–28, 2015.

XUEYUAN GONG received the B.Sc. degree (Hons.) from the Macau University of Science and Technology in 2011 and the M.Sc. degree in computer science from the University of Macau in 2015, where he is currently pursuing the Ph.D. degree. He is currently a member of the Data Analytics and Collaborative Computing Research Group, University of Macau. He is a Research Assistant. He has published more than 10 EI-indexed conferences and SCI-indexed journal papers. His research interests include data mining, artificial intelligence, and optimization. He was a recipient of the University Fellowship in 2012.

LIANSHENG LIU received the Ph.D. degree in medicine from Southern Medical University, China. He is currently an Associate Professor with the First Affiliated Hospital, Guangzhou University of Traditional Chinese Medicine. He is engaged in clinical diagnosis of medical imaging teaching and scientific research for more than 20 years. His research interests are head and neck imaging and magnetic resonance diagnosis. Presided over six projects at all levels, participate in compiling the National Health and Family Planning Commission *13th Five-Year* planning textbook Imaging. He was also a member of the Head and Neck Imaging Committee of the Chinese Medical Association, and the Head and Neck Section of the Guangdong Society of Radiology.

SIMON FONG received the B.Eng. degree (Hons.) in computer systems and the Ph.D. degree in computer science from La Trobe University, Australia, in 1993 and 1998, respectively. He took up various managerial and technical posts, such as a Systems Engineer, an IT Consultant, and the E-Commerce Director in Melbourne, Hong Kong, and Singapore. He was with Hong Kong Telecom, Singapore Network Services, AES Pro-Data, and United Oversea Bank, Singapore. He was an Assistant Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. He is currently an Associate Professor with the Computer and Information Science Department, University of Macau. He is also one of the founding members of the Data Analytics and Collaborative Computing Research Group, Faculty of Science and Technology. He has published over 200 international conference and peer-reviewed journal papers, mostly in the area of e-commerce technology, business intelligence, and data mining.



QIWEN XU received the B.Sc. degree in computer science from East China Normal University, China, in 1985, and the Ph.D. degree in computing from the Computing Laboratory, Oxford University, U.K., in 1992. He is currently an Assistant Professor with the Department of Computer and Information Science, University of Macau. His research interests include program verification and refinement, formal specification, real time systems, and temporal logic. Over the years, he received many research grants such as Formal Methods for Java Like Programs, University of Macau (Ref No. RG039/02-038/XQW/FST); Natural Science Foundation of China 2006AA01Z165, from 2007 to 2010; Natural Science Foundation of China 20006201ïijN from 2008 to 2011; Process Expansion: Action Refinement in the Large, Macao Science and Technology Fund, from 2008 to 2011; Engineering Accountable Ensembles, Macao Science and Technology Development Fund, from 2010 to 2014; and the Open Project of Shanghai Key Laboratory of Trustworthy Computing (No. 07dz22304201202), in 2013.

TINGXI WEN was born in China. He received the M.S. degree in software engineering from Xiamen University. He has published widely in the field of robotic control based on EMG using various data classification methods. He has also developed a novel multi-objective optimization technique based on Spark platform applied for optimizing routing. His research interests include data mining, machine learning, and cloud computing. He is a reviewer for many biomedical engineering journals.

ZHIHUA LIU received the degree from Southeast University and the Ph.D. degree from Harvard University. He joined the Department of Biostatistics and Computational Biology, Dana–Farber Cancer Institute. He is currently a Professor with the Chinese Academy of Sciences. His research interests include bioinformatics and computational biology, big data in biomedicine, and artificial intelligence in medicine. As a first author, he has published more than 20 SCI-indexed papers in high impact journals such as bioinformatics, molecular biology, and evolution.