Information Sciences 222 (2013) 175–184

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Black hole: A new heuristic optimization approach for data clustering

Abdolreza Hatamlou *

Islamic Azad University, Khoy Branch, Iran
Data Mining and Optimization Research Group, Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 26 September 2011
Received in revised form 4 July 2012
Accepted 18 August 2012
Available online 4 September 2012

Keywords:
Nature-inspired algorithm
Black hole phenomenon
Data clustering

ABSTRACT

Nature has always been a source of inspiration. Over the last few decades, it has stimulated many successful algorithms and computational tools for dealing with complex and optimization problems. This paper proposes a new heuristic algorithm that is inspired by the black hole phenomenon. Similar to other population-based algorithms, the black hole algorithm (BH) starts with an initial population of candidate solutions to an optimization problem and an objective function that is calculated for them. At each iteration of the black hole algorithm, the best candidate is selected to be the black hole, which then starts pulling other candidates around it, called stars. If a star gets too close to the black hole, it will be swallowed by the black hole and is gone forever. In such a case, a new star (candidate solution) is randomly generated and placed in the search space and starts a new search. To evaluate the performance of the black hole algorithm, it is applied to solve the clustering problem, which is a NP-hard problem. The experimental results show that the proposed black hole algorithm outperforms other traditional heuristic algorithms for several benchmark datasets.

1. Introduction

Nature-inspired metaheuristic algorithms are becoming popular and powerful in solving optimization problems [9,49,53,86]. A wide range of nature-inspired algorithms have emerged over the last few decades. For instance genetic algorithms (GAs) are search and optimization techniques that evolve a population of candidate solutions to a given problem, using natural genetic variation and natural selection operators [41]. The simulated annealing (SA) algorithm was developed by modelling the steel annealing process [48]. The ant colony optimization (ACO) was inspired from the behavior of a real ant colony, which is able to find the shortest path between its nest and a food source [19]. The particle swarm optimization (PSO) algorithm was developed based on the swarm behavior, such as fish and bird schooling in nature [52,75]. The gravitational search algorithm (GSA) was constructed based on the law of gravity and the notion of mass interactions. In the GSA algorithm, the searcher agents are a collection of masses that interact with each other based on the Newtonian gravity and the laws of motion [77]. The intelligent water drops (IWDs) algorithm was inspired from observing natural water drops that flow in rivers and how natural rivers find almost optimal paths to their destination. In the IWD algorithm, several artificial water drops cooperate to change their environment in such a way that the optimal path is revealed as the one with the lowest soil on its links [80]. The firefly algorithm (FA) was inspired by the flashing behavior of fireflies in nature [4], while the honey bee mating optimization (HBMO) algorithm was inspired by the process of marriage in real honey bees [23,61]. The Bat Algorithm (BA) was inspired by the echolocation behavior of bats. The capability of the echolocation of bats is fascinating as they can find their prey and recognize different types of insects even in complete darkness [31]. The harmony search

* Address: Islamic Azad University, Khoy Branch, Iran.
E-mail addresses: hatamlou@iaukhoy.ac.ir, hatamlou@ftsm.ukm.my, rezahatamloo@gmail.com.

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.08.023

optimization algorithm was inspired by the improvising process of composing a piece of music. The action of finding the harmony in music is similar to finding the optimal solution in an optimization process [27]. The Big Bang–Big Crunch (BB–BC) optimization is based on one of the theories of the evolution of the universe. It is composed of the big bang and big crunch phases. In the big bang phase the candidate solutions are spread at random in the search space and in the big crunch phase a contraction procedure calculates a center of mass for the population [22].

Nature-inspired metaheuristic algorithms have now been used in many fields such as computer science [2,21,30], data mining [76,87], industry [24], agriculture [26], computer vision [13,14,70,82], forecasting [10], medicine and biology [12], scheduling [34], economy [90] and engineering [60].

This paper presents a new optimization method and its application to data clustering which is inspired by the black hole phenomenon. The basic idea of a black hole is simply a region of space that has so much mass concentrated in it that there is no way for a nearby object to escape its gravitational pull. Anything falling into a black hole, including light, is forever gone from our universe. The proposed black hole algorithm (BH) starts with an initial population of candidate solutions to an optimization problem and an objective function that is calculated for them. At each iteration of the BH, the best candidate is selected to be the black hole and the rest form the normal stars. After the initialization process, the black hole starts pulling stars around it. If a star gets too close to the black hole it will be swallowed by the black hole and is gone forever. In such a case, a new star (candidate solution) is randomly generated and placed in the search space and starts a new search.

The rest of the paper is organized as follows: In Section 2, the clustering problem is discussed. A brief explanation of the black hole phenomenon is given in Section 3. In Section 4, we introduce our proposed black hole algorithm and its application to data clustering. The performance of the proposed algorithm is tested with several benchmark datasets and compared with *K*-means, particle swarm optimization algorithm (PSO), Big Bang–Big Crunch algorithm (BB–BC) and gravitational search algorithm (GSA) in Section 5. Finally, Section 6 includes a summary and the conclusion of this work.

2. Cluster analysis

Data clustering is one of the most important and popular data analysis techniques, and refers to the process of grouping a set of data objects into clusters, in which the data of a cluster must have great similarity and the data of different clusters must have high dissimilarity [3,36,46].

Basically, to evaluate the similarity between data objects, the distance measurement is used. Particularly, the problem is specified as follows: given *N* objects, assign each object to one of *K* clusters and minimize the sum of squared Euclidean distances between each object and the center of the cluster that belongs to every allocated object:

$$F(O,Z) = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\|(O_i - Z_j)\|^2 \tag{1}$$

where $||O_i - Z_j||$ is the Euclidean distance between a data object O_i and the cluster center Z_j. *N* and *K* are the number of data objects and the number of clusters, respectively. w_{ij} is the association weight of data object O_i with cluster *j*, which will be either 1 or 0 (if object *i* is assigned to cluster *j*; w_{ij} is 1, otherwise 0). Fuzzy clustering allows w_{ij} to take values in the interval (0,1).

There are many clustering algorithms in the literature. The classical clustering algorithms are broadly classified as hierarchical and partitional algorithms [36,45,46]. Among the classical clustering algorithms, *K*-means is the most well known algorithm due to its simplicity and efficiency [36,46]. However, it suffers from two problems. It needs the number of clusters before starting (i.e., the number of clusters must be known a priori). In addition, its performance strongly depends on the initial centroids and may get stuck in local optima solutions [78]. In order to overcome the shortcomings of *K*-means, many heuristic approaches have been applied in the last two decades. For instance, simulated annealing [33], tabu search [58], genetic algorithms [57,62,63,66], ant colony optimization [28,67,89], neural gas algorithm [72–74], honey bee mating optimization [23], differential evolution algorithm [15,16], particle swarm optimization algorithm [1,44,54], artificial bee colony [51], gravitational search algorithm [38–39], a binary search algorithm [40], firefly algorithm [79], and big bang–big crunch algorithm [37] have been used for data clustering.

Clustering techniques have been used in many areas such as image processing [17,47,65,85], document clustering [6,8,59], geophysics [29,81], prediction [7,11], marketing and costumer analysis [55,83], agriculture [68], security and crime detection [32], medicine [35,42,56], anomaly detection [25,69] and biology [43,84].

3. Black hole phenomenon

In the eighteens-century John Michell and Pierre Laplace were the pioneers to identify the concept of black holes. Integrating Newton's law they formulated the theory of a star becoming invisible to the eye, however, during that period it was not known as a black hole and it was only in 1967 that John Wheeler the American physicist first named the phenomenon of mass collapsing as a black hole.

A black hole in space is what forms when a star of massive size collapses. The gravitational power of the black hole is too high that even the light cannot escape from it. The gravity is so strong because matter has been squeezed into a tiny space. Anything that crosses the boundary of the black hole will be swallowed by it and vanish and nothing can get away from its enormous power. The sphere-shaped boundary of a black hole in space is known as the event horizon. The radius of the event horizon is termed as the Schwarzschild radius. At this radius, the escape speed is equal to the speed of light, and once light passes through, even it cannot escape. Nothing can escape from within the event horizon because nothing can go faster than light. The Schwarzschild radius is calculated by the following equation:

$$R = \frac{2GM}{c^2} \tag{2}$$

where G is the gravitational constant, M is the mass of the black hole, and c is the speed of light.

If anything moves close to the event horizon or crosses the Schwarzschild radius it will be absorbed into the black hole and permanently disappear. The existence of black holes can be discerned by its effect over the objects surrounding it [50,71].

4. Black hole algorithm

The BH algorithm is a population-based method that has some common features with other population-based methods. As with other population-based algorithms, a population of candidate solutions to a given problem is generated and distributed randomly in the search space. The population-based algorithms evolve the created population towards the optimal solution via certain mechanisms. For example, in GAs, the evolving is done by mutation and crossover operations. In PSO, this is done by moving the candidate solutions around in the search space using the best found locations, which are updated as better locations are found by the candidates. In the proposed BH algorithm the evolving of the population is done by moving all the candidates towards the best candidate in each iteration, namely, the black hole and replacing those candidates that enter within the range of the black hole by newly generated candidates in the search space. The black hole terminology has been used for the first time in solving benchmark functions [88]. However, that method is different from the proposed BH algorithm in this paper. The proposed method in [88] introduces a new mechanism into PSO, which is named the black hole. In this method, at each iteration, a new particle is generated randomly near to the best particle, and then, based on two random generated numbers, the algorithm updates the locations of the particles either by the PSO or the new mechanism. In other words, that method is an extension of the PSO and a new generated particle called the black hole attracts other particles under certain conditions, which used to accelerate the convergence speed of the PSO and also to prevent the local optima problem. In this method there is nothing about the event horizon of the black hole and the destruction of the stars (candidates). The proposed BH algorithm in this paper is more similar to the natural black hole phenomenon and is completely different from the black hole PSO. In our BH algorithm the best candidate among all the candidates at each iteration is selected as a black hole and all the other candidates form the normal stars. The creation of the black hole is not random and it is one of the real candidates of the population. Then, all the candidates are moved towards the black hole based on their current location and a random number. The details of the BH algorithms are as follows:

Like other population-based algorithms, in the proposed black hole algorithm (BH) a randomly generated population of candidate solutions – the stars – are placed in the search space of some problem or function. After initialization, the fitness values of the population are evaluated and the best candidate in the population, which has the best fitness value, is selected to be the black hole and the rest form the normal stars. The black hole has the ability to absorb the stars that surround it.

After initializing the black hole and stars, the black hole starts absorbing the stars around it and all the stars start moving towards the black hole. The absorption of stars by the black hole is formulated as follows:

$$x_i(t+1) = x_i(t) + rand \times (x_{BH} -, x_i(t)) \quad i = 1, 2, \ldots, N \tag{3}$$

where $x_i(t)$ and $x_i(t+1)$ are the locations of the ith star at iterations t and $t+1$, respectively. x_{BH} is the location of the black hole in the search space. $rand$ is a random number in the interval $[0,1]$. N is the number of stars (candidate solutions).

While moving towards the black hole, a star may reach a location with lower cost than the black hole. In such a case, the black hole moves to the location of that star and vice versa. Then the BH algorithm will continue with the black hole in the new location and then stars start moving towards this new location.

In addition, there is the probability of crossing the event horizon during moving stars towards the black hole. Every star (candidate solution) that crosses the event horizon of the black hole will be sucked by the black hole. Every time a candidate (star) dies – it is sucked in by the black hole – another candidate solution (star) is born and distributed randomly in the search space and starts a new search. This is done to keep the number of candidate solutions constant. The next iteration takes place after all the stars have been moved.

The radius of the event horizon in the black hole algorithm is calculated using the following equation:

$$R = \frac{f_{BH}}{\sum_{i=1}^{N} f_i} \tag{4}$$

where f_{BH} is the fitness value of the black hole and f_i is the fitness value of the ith star. N is the number of stars (candidate solutions). When the distance between a candidate solution and the black hole (best candidate) is less than R, that candidate is collapsed and a new candidate is created and distributed randomly in the search space.

Based on the above description the main steps in the BH algorithm are summarized as follows:

```
Initialize a population of stars with random locations in the search space
Loop
  For each star, evaluate the objective function
  Select the best star that has the best fitness value as the black hole
  Change the location of each star according to Eq. (3)
  If a star reaches a location with lower cost than the black hole, exchange their locations
  If a star crosses the event horizon of the black hole, replace it with a new star in a random location in the search space
  If a termination criterion (a maximum number of iterations or a sufficiently good fitness) is met, exit the loop
End loop
```

To assess the performance of the BH algorithm we have applied it to solve the clustering problem. According to [20] while the quantity of clusters goes beyond three the clustering problem becomes NP-hard.

The candidate solution to the clustering problem corresponds to a 1-dimensional array while applying black hole algorithm for data clustering. Every candidate solution is considered as *k* initial cluster centers and the individual unit in the array as the cluster center dimension. Fig. 1 illustrates a candidate solution of a problem with three clusters and all the data objects have four features.

5. Experimental results

Six benchmark datasets with a variety of complexity are used to evaluate the performance of the proposed approach. The datasets are *Iris*, *Wine*, *Glass*, *Wisconsin Breast Cancer*, *Vowel* and *Contraceptive Method Choice* (*CMC*), which are available in the repository of the machine learning databases [5]. Table 1 summaries the main characteristics of the used datasets.

The performance of the BH algorithm is compared against well known and the most recent algorithms reported in the literature, including *K*-means [46], particle swarm optimization [52], gravitational search algorithm [38] and the big bang–big crunch algorithm [37]. The performance of the algorithms is evaluated and compared using two criteria:

- Sum of intra-cluster distances as an internal quality measure: The distance between each data object and the center of the corresponding cluster is computed and summed up, as defined in Eq. (1). Clearly, the smaller the sum of intra-cluster distances, the higher the quality of the clustering. The sum of intra-cluster distances is also the evaluation fitness in this work.
- Error Rate (ER) as an external quality measure: The percentage of misplaced data objects, as shown in the following equation:

$$\mathrm{ER} = \frac{\text{number of misplaced objects}}{\text{total umber of objects within dataset}} \times 100 \tag{5}$$

A summary of the intra-cluster distances obtained by the clustering algorithms is given in Table 2. The values reported are best, average, worst and the standard deviation of solutions over 50 independent simulations.

As seen from the results in Table 2, the BH algorithm achieved the best results among all the algorithms. For the *Iris* dataset, the best, worst, and average solutions obtained by BH are 96.65589, 96.65681, and 96.66306, respectively, which are better than the other algorithms. Foe the *Wine* dataset, the BH algorithm achieved the optimum value of 16293.41995, which is significantly better than the other test algorithms. As seen from the results for the *Glass* dataset, the BH algorithm is far superior to the other algorithms. The worst solution obtained by the BH algorithm on the *Glass* dataset is 213.95689, which is much better than the best solutions found by the other algorithms. For the *Cancer* dataset, the BH algorithm outperformed the *K*-means, PSO and GSA algorithms; however, the results of the BB–BC algorithm are better than the BH. For the *CMC* dataset, the proposed BH algorithm reached an average of 5533.63122, while other algorithms were unable to reach this solution even once within 50 runs. On the *Vowel* dataset, the BH algorithm provided the best solutions and small standard deviation compared to the other algorithms.

From the above results, we can say that in five of the test datasets the proposed BH algorithm is superior to the other test algorithms. It can find high quality solutions and provides small standard deviation. In other words, the BH algorithm converges to global optimum in all the runs while the other algorithms may get trapped in local optimum solutions. Only in the

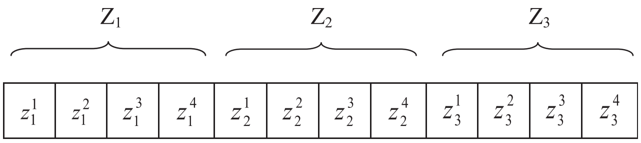
Z1
Z2
Z3
z_1^1 z_1^2 z_1^3 z_1^4 z_2^1 z_2^2 z_2^3 z_2^4 z_3^1 z_3^2 z_3^3 z_3^4

Fig. 1. Example of a candidate solution.

Table 1
Main characteristics of the test datasets.

Dataset	Number of clusters	Number of features	Number of data objects
Iris	3	4	150 (50,50,50)
Wine	3	13	178 (59,71,48)
Glass	6	9	214 (70,76,17,13,9,29)
Cancer	2	9	683 (444,239)
Vowel	6	3	871 (72,89,172,151,207,180)
CMC	3	9	1473 (629,334,510)

Table 2
The sum of intra-cluster distances obtained by algorithms on different datasets.

Dataset	Criteria	*K*-means	PSO	GSA	BB–BC	BH
Iris	Best	97.32592	96.87935	96.68794	96.67648	96.65589
	Average	105.72902	98.14236	96.73105	96.76537	96.65681
	Worst	128.40420	99.76952	96.82463	97.42865	96.66306
	Std	12.38759	0.84207	0.02761	0.20456	0.00173
Wine	Best	16,555.67942	16,304.48576	16,313.87620	16,298.67356	16,293.41995
	Average	16,963.04499	16,316.27450	16,374.30912	16,303.41207	16,294.31763
	Worst	23,755.04949	16,342.78109	16,428.86494	16,310.11354	16,300.22613
	Std	1180.69420	12.60275	34.67122	2.66198	1.65127
Glass	Best	215.67753	223.90546	224.98410	223.89410	210.51549
	Average	227.97785	230.49328	233.54329	231.23058	211.49860
	Worst	260.83849	246.08915	248.36721	243.20883	213.95689
	Std	14.13889	4.79320	6.13946	4.65013	1.18230
Cancer	Best	2986.96134	2974.48092	2965.76394	2964.38753	2964.38878
	Average	3032.24781	2981.78653	2972.66312	2964.38798	2964.39539
	Worst	5216.08949	3053.49132	2993.24458	2964.38902	2964.45074
	Std	315.14560	10.43651	8.91860	0.00048	0.00921
Vowel	Best	149,394.80398	152,461.56473	151,317.56392	149,038.51683	148,985.61373
	Average	153,660.80712	153,218.23418	152,931.81044	151,010.03392	149,848.18144
	Worst	168,474.26593	158,987.08231	155,346.69521	153,090.44077	153,058.98663
	Std	4123.04203	2945.23167	2486.70285	1859.32353	1306.95375
CMC	Best	5542.18214	5539.17452	5542.27631	5534.09483	5532.88323
	Average	5543.42344	5547.89320	5581.94502	5574.75174	5533.63122
	Worst	5545.33338	5561.65492	5658.76293	5644.70264	5534.77738
	Std	1.52384	7.35617	41.13648	39.43494	0.59940

Cancer dataset did one of the algorithms (BB–BC) reach a better solution than the BH. Even in this dataset, the BH algorithm reached high quality clusters compared to the other three test algorithms.

Table 3 shows the mean error rate obtained by the clustering algorithms from 50 simulation runs on the test datasets. As seen from the results in Table 3, the BH algorithm provided a minimum average error rate in all the test datasets.

In order to find significant differences among the results obtained by the clustering algorithms, statistical analysis is carried out. We employed the Friedman test as well as the Iman–Davenport test to determine whether there are significant differences in the results of the clustering algorithms. If there are statistically significant differences, then we proceed with the Holm as a post hoc test, which is used to compare the best performing algorithm (control algorithm) against the remaining ones. We used $\alpha = 0.05$ as the level of confidence in all cases. A wider description of these tests is presented in [18,64].

Table 4 reports the average ranking of clustering algorithms obtained by the Friedman's test based on the sum of intra-cluster distances. The proposed BH algorithm is ranked first, followed by BB–BC, GSA, PSO and *K*-means, successively.

Table 3
The error rate of clustering algorithms on the test datasets.

Dataset	*K*-means (%)	PSO (%)	GSA (%)	BB–BC (%)	BH (%)
Iris	13.42	10.06	10.04	10.05	10.02
Wine	31.14	28.79	29.15	28.52	28.47
Glass	38.44	41.20	41.39	41.37	36.51
Cancer	4.39	3.79	3.74	3.70	3.70
Vowel	43.57	42.39	42.26	41.89	41.65
CMC	54.48	54.50	55.67	54.52	54.39

Table 4
Average ranking of clustering algorithms based on the sum of intra-cluster distances.

Algorithm	*K*-means	PSO	GSA	BB–BC	BH
Ranking	4	3.49999	3.66666	2.66666	1.16666

Table 5
Results of Friedman's and Iman–Davenport's tests based on the sum of intra-cluster distances.

Method	Statistical value	*p*-Value	Hypothesis
Friedman	12.40000	0.01461	Rejected
Iman–Davenport	5.34482	0.00429	Rejected

Table 6
Results of the Holm's method based on the sum of intra-cluster distances (BH is the control algorithm).

i	Algorithm	*z*	*p*-Value	α/i	Hypothesis
4	*K*-means	3.10376	0.00191	0.0125	Rejected
3	GSA	2.73861	0.00616	0.01666	Rejected
2	PSO	2.55603	0.01058	0.025	Rejected
1	BB–BC	1.64316	0.10034	0.05	Not rejected

Table 7
Average ranking of clustering algorithms based on the error rate.

Algorithm	*K*-means	PSO	GSA	BB–BC	BH
Ranking	4	3.33333	3.66666	2.91666	1.08333

Table 8
Results of Friedman's and Iman–Davenport's tests based on the error rate.

Method	Statistical value	*p*-Value	Hypothesis
Friedman	12.56666	0.01359	Rejected
Iman–Davenport	5.49562	0.00375	Rejected

Table 9
Results of the Holm's method based on the error rate (BH is the control algorithm).

i	Algorithm	*z*	*p*-Value	α/i	Hypothesis
4	*K*-means	3.19504	0.00139	0.0125	Rejected
3	GSA	2.82989	0.00465	0.01666	Rejected
2	PSO	2.46475	0.01371	0.025	Rejected
1	BB–BC	2.00831	0.04460	0.05	Rejected

Table 10
The best centroids obtained by the BH algorithm on the *Iris* dataset.

Iris		
Center 1	Center 2	Center 3
6.73305	5.01186	5.93229
3.06805	3.40303	2.79775
5.62938	1.47143	4.41857
2.10908	0.23532	1.41608

The *p*-value computed by the Friedman test and the Iman–Davenport test are given in Table 5, which both reject the null hypothesis of equivalent performance and confirm the existence of significant differences among the performance of all the clustering algorithms. Therefore, the Holm's method is carried out as a post hoc test to detect effective statistical differences between the control approach, i.e., the one with the lowest Friedman's rank, and the remaining approaches, the results of

Table 11
The best centroids obtained by the BH algorithm on the *Wine* dataset.

Wine		
Center 1	Center 2	Center 3
12.87096	12.63469	13.31401
2.11606	2.44139	2.26752
2.39431	2.37083	2.56857
19.46178	21.26462	17.34232
98.84497	92.39332	105.03031
2.03580	2.12789	2.82361
1.44765	1.58430	3.24277
0.43320	0.40206	0.28947
1.49193	1.13521	2.67352
5.36444	4.83774	5.20622
0.88652	0.81497	1.03286
2.12046	2.71348	3.38781
686.93205	463.69590	1137.44167

Table 12
The best centroids obtained by the BH algorithm on the *Glass* dataset.

Glass					
Center 1	Center 2	Center 3	Center 4	Center 5	Center 6
1.51474	1.52117	1.51745	1.51326	1.51743	1.52095
14.59500	13.79589	13.31326	13.01074	12.85016	13.02689
0.06789	3.55131	3.59522	−0.00358	3.45851	0.26652
2.25305	0.95428	1.42358	3.02527	1.30894	1.51925
73.29150	71.84335	72.67659	70.66960	73.02754	72.75985
0.00937	0.19175	0.57686	6.22227	0.60704	0.35290
8.71261	9.54099	8.20015	6.94351	8.58511	11.95589
1.01385	0.08156	−0.00741	−0.00710	0.02745	−0.04668
−0.01161	0.00710	0.03106	−0.00041	0.05789	0.03072

Table 13
The best centroids obtained by the BH algorithm on the *Cancer* dataset.

Cancer	
Center 1	Center 2
2.88939	7.11206
1.12825	6.64387
1.20020	6.62667
1.16519	5.61122
1.99385	5.23857
1.12076	8.10586
2.00426	6.07815
1.10184	6.01691
1.03182	2.32526

which are shown in Table 6. The results of the Holm's method reveal that the control algorithm (BH) is statistically better than the *K*-means, GSA and PSO regarding the sum of intra-cluster distances. In the BB–BC case, there is no significant difference based on the Holm's method results. However, the results reported in Table 2 show that the proposed BH approach outperforms BB–BC in 5 out 6 datasets.

The same procedure is performed to check whether there are significant differences in the error rate of clustering algorithms. The results are shown in Tables 7–9. The results obtained by the Friedman's test indicate that the BH algorithm is ranked first and there are significant differences in the results of the algorithms. Moreover, from the results of the Holm's method in Table 9, it could be concluded that the control algorithm (BH) performs significantly better regarding the error rate than the remaining algorithms, with a significant level of 0.05.

Tables 10–15 show the best centroids obtained by the BH algorithm on the test datasets. The best centroids are presented to validate the sum of intra-cluster distances in Table 2. By assigning the data objects within each dataset to the corresponding centroids in Tables 10–15, the best values in Table 2 should be reached. For example, by assigning all of the 150 data objects within the *Iris* dataset to the nearest centroid among the three centroids that are presented in Table 10, the best value

Table 14
The best centroids obtained by the BH algorithm on the *Vowel* dataset.

Vowel					
Center 1	Center 2	Center 3	Center 4	Center 5	Center 6
506.77159	407.56882	623.56890	356.26075	376.46463	437.32102
1839.53155	1012.04806	1309.41100	2291.80365	2150.14725	992.97518
2556.19007	2311.15800	2333.37166	2977.13380	2677.81056	2658.27897

Table 15
The best centroids obtained by the BH algorithm on the *CMC* dataset.

CMC		
Center 1	Center 2	Center 3
24.42273	43.63258	33.49565
3.03421	2.99608	3.13181
3.51476	3.45429	3.56438
1.79348	4.57393	3.64850
0.92053	0.82686	0.79404
0.82924	0.83295	0.66550
2.29826	1.82888	2.09068
2.95830	3.47833	3.29362
0.02510	0.11822	0.06771

for the sum of the intra-cluster distances found by the BH algorithm in the *Iris* dataset, which is reported in Table 2, should be reached (96.65589). Otherwise either the best values in Table 2 or the best centroids in Table 10 or both of them are wrong. This procedure can also be done for other test datasets.

6. Conclusion

Modelling and simulating natural phenomena for solving complex problems has been an interesting research area for several decades. In this paper, we have introduced a new heuristic optimization algorithm based on the black hole phenomenon. There are two significant advantages for the proposed BH algorithm. First, it has a simple structure and it is easy to implement. Second, it is free from parameter tuning issues. The proposed algorithm was applied to solve the clustering problem. The results of the experiment using six benchmark datasets show that the black hole (BH) algorithm outperforms other test algorithms in most of the datasets. In future research, the proposed algorithm can also be utilized for many different areas of applications. In addition, the application of BH in combination with other algorithms may be effective.

References

[1] A. Ahmadi, F. Karray, M.S. Kamel, Model order selection for multiple cooperative swarms clustering using stability analysis, Information Sciences 182 (2012) 169–183.
[2] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences 192 (2012) 120–142.
[3] W. Barbakh, Y. Wu, C. Fyfe, Review of clustering algorithms, in: Non-Standard Parameter Adaptation for Exploratory Data Analysis, Springer, Berlin/Heidelberg, 2009, pp. 7–28.
[4] M. Bramer, R. Ellis, M. Petridis, X.-S. Yang, Firefly algorithm, Lévy flights and global optimization, Research and Development in Intelligent Systems, vol. XXVI, Springer, London, 2010, pp. 209–218.
[5] C.J. Merz, C.L. Blake, UCI Repository of Machine Learning Databases. <http://www.ics.uci.edu/-mlearn/MLRepository.html>.
[6] X. Cai, W. Li, A spectral analysis approach to document summarization: clustering and ranking sentences simultaneously, Information Sciences 181 (2011) 3816–3827.
[7] G. Cardoso, F. Gomide, Newspaper demand prediction and replacement model based on fuzzy clustering and rules, Information Sciences 177 (2007) 4799–4809.
[8] M. Carullo, E. Binaghi, I. Gallo, An online document clustering technique for short web contents, Pattern Recognition Letters 30 (2009) 870–876.
[9] O. Castillo, R. Martinez-Marroquin, P. Melin, F. Valdez, J. Soria, Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot, Information Sciences 192 (2012) 19–38.
[10] D. Chaturvedi, Applications of genetic algorithms to load forecasting problem, in: Soft Computing, Springer, Berlin/Heidelberg, 2008, pp. 383–402.
[11] S.-M. Chen, Y.-C. Chang, Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques, Information Sciences 180 (2010) 4772–4783.
[12] J. Christmas, E. Keedwell, T.M. Frayling, J.R.B. Perry, Ant colony optimisation to identify genetic variant association with type 2 diabetes, Information Sciences 181 (2011) 1609–1622.
[13] J.-F. Connolly, E. Granger, R. Sabourin, An adaptive classification system for video-based face recognition, Information Sciences 192 (2012) 50–70.
[14] E. Cuevas, D. Oliva, D. Zaldivar, M. Perez-Cisneros, H. Sossa, Circle detection using electro-magnetism optimization, Information Sciences 182 (2012) 40–55.
[15] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved differential evolution algorithm, IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans 38 (2008) 218–237.

[16] S. Das, A. Abraham, A. Konar, Automatic hard clustering using improved differential evolution algorithm, in: Studies in Computational Intelligence, 2009, pp. 137–174.
[17] S. Das, S. Sil, Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm, Information Sciences 180 (2009) 1237–1256.
[18] J. Derrac, S. Garcia, 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 and Evolutionary Computation 1 (2011) 3–18.
[19] M. Dorigo, C. Blum, Ant colony optimization theory: a survey, Theoretical Computer Science 344 (2005) 243–278.
[20] R. Drenick, F. Kozin, T. Gonzalez, On the computational complexity of clustering and related problems, in: System Modeling and Optimization, Springer, Berlin/Heidelberg, 1982, pp. 174–182.
[21] M. El-Abd, Performance assessment of foraging algorithms vs. evolutionary algorithms, Information Sciences 182 (2012) 243–263.
[22] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, Advances in Engineering Software 37 (2006) 106–111.
[23] M. Fathian, B. Amiri, A. Maroosi, Application of honey-bee mating optimization algorithm on clustering, Applied Mathematics and Computation 190 (2007) 1502–1513.
[24] B. Fox, W. Xiang, H. Lee, Industrial applications of the ant colony optimization algorithm, The International Journal of Advanced Manufacturing Technology 31 (2007) 805–814.
[25] M. Friedman, M. Last, Y. Makover, A. Kandel, Anomaly detection in web documents using crisp and fuzzy-based cosine clustering methodology, Information Sciences 177 (2007) 467–475.
[26] Z. Geem, M. Cisty, Application of the harmony search optimization in irrigation, in: Recent Advances in Harmony Search Algorithm, Springer, Berlin/Heidelberg, 2010, pp. 123–134.
[27] Z. Geem, X.-S. Yang, Harmony search as a metaheuristic algorithm, in: Music-Inspired Harmony Search Algorithm, Springer, Berlin/Heidelberg, 2009, pp. 1–14.
[28] A. Ghosh, A. Halder, M. Kothari, S. Ghosh, Aggregation pheromone density based data clustering, Information Sciences 178 (2008) 2816–2831.
[29] A. Ghosh, N.S. Mishra, S. Ghosh, Fuzzy clustering algorithms for unsupervised change detection in remote sensing images, Information Sciences 181 (2011) 699–715.
[30] S. Ghosh, S. Das, S. Roy, S.K. Minhazul Islam, P.N. Suganthan, A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization, Information Sciences 182 (2012) 199–219.
[31] J. Gonzalez, D. Pelta, C. Cruz, G. Terrazas, N. Krasnogor, X.-S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, Berlin/Heidelberg, 2011, pp. 65–74.
[32] T. Grubesic, On the application of fuzzy clustering for crime hot spot detection, Journal of Quantitative Criminology 22 (2006) 77–105.
[33] Z. Gungr, A. Unler, *K*-harmonic means data clustering with simulated annealing heuristic, Applied Mathematics and Computation 184 (2007) 199–209.
[34] Y.W. Guo, W.D. Li, A.R. Mileham, G.W. Owen, Applications of particle swarm optimisation in integrated process planning and scheduling, Robotics and Computer-Integrated Manufacturing 25 (2009) 280–288.
[35] W. Halberstadt, T.S. Douglas, Fuzzy clustering to detect tuberculous meningitis-associated hyperdensity in CT images, Computers in Biology and Medicine 38 (2008) 165–170.
[36] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Academic Press, 2006.
[37] A. Hatamlou, S. Abdullah, M. Hatamlou, Data clustering using big bang–big crunch algorithm, in: Communications in Computer and Information Science, 2011, pp. 383–388.
[38] A. Hatamlou, S. Abdullah, H. Nezamabadi-pour, Application of gravitational search algorithm on data clustering, in: Rough Sets and Knowledge Technology, Springer, Berlin/Heidelberg, 2011, pp. 337–346.
[39] A. Hatamlou, S. Abdullah, H. Nezamabadi-pour, A combined approach for clustering based on *K*-means and gravitational search algorithms, Swarm and Evolutionary Computation 6 (2012) 47–52.
[40] A. Hatamlou, In search of optimal centroids on data clustering using a binary search algorithm, Pattern Recognition Letters 33 (2012) 1756–1760.
[41] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, second ed., John Wiley & Sons, 2004.
[42] S. Hirano, X. Sun, S. Tsumoto, Comparison of clustering methods for clinical databases, Information Sciences 159 (2004) 155–165.
[43] E.R. Hruschka, R.J.G.B. Campello, L.N. de Castro, Evolving clusters in gene-expression data, Information Sciences 176 (2006) 1898–1927.
[44] H. Izakian, A. Abraham, Fuzzy *C*-means and fuzzy swarm for fuzzy clustering problem, Expert Systems with Applications 38 (2011) 1835–1838.
[45] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, in: Computing Surveys, ACM, 1999, pp. 264–323.
[46] A.K. Jain, Data clustering: 50 years beyond *K*-means, Pattern Recognition Letters 31 (2010) 651–666.
[47] R.I. John, P.R. Innocent, M.R. Barnes, Neuro-fuzzy clustering of radiographic tibia image data using type 2 fuzzy sets, Information Sciences 125 (2000) 65–82.
[48] D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon, Optimization by simulated annealing. An experimental evaluation. Part I. Graph partitioning, Operations Research 37 (1989) 865–892.
[49] F. Kang, J. Li, Z. Ma, Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions, Information Sciences 181 (2011) 3508–3531.
[50] L. Kaper, E. Heuvel, P. Woudt, R. Giacconi, Black hole research past and future, in: Black Holes in Binaries and Galactic Nuclei: Diagnostics, Demography and Formation, Springer, Berlin/Heidelberg, 2001, pp. 3–15.
[51] D. Karaboga, C. Ozturk, A novel clustering approach: artificial bee colony (ABC) algorithm, Applied Soft Computing 11 (2011) 652–657.
[52] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, vol. 1944, 1995, pp. 1942–1948.
[53] D. Kundu, K. Suresh, S. Ghosh, S. Das, B.K. Panigrahi, S. Das, Multi-objective optimization with artificial weed colonies, Information Sciences 181 (2011) 2441–2454.
[54] R.J. Kuo, Y.J. Syu, Z.-Y. Chen, F.C. Tien, Integration of particle swarm optimization and genetic algorithm for dynamic clustering, Information Sciences 195 (2012) 124–140.
[55] J. Li, K. Wang, L. Xu, Chameleon based on clustering feature tree and its application in customer segmentation, Annals of Operations Research 168 (2009) 225–245.
[56] L. Liao, T. Lin, B. Li, MRI brain image segmentation and bias field correction based on fast spatially constrained kernel clustering approach, Pattern Recognition Letters 29 (2008) 1580–1588.
[57] R. Liu, L. Jiao, X. Zhang, Y. Li, Gene transposon based clone selection algorithm for automatic clustering, Information Sciences 204 (2012) 1–22.
[58] Y. Liu, Z. Yi, H. Wu, M. Ye, K. Chen, A tabu search approach for the minimum sum-of-squares clustering problem, Information Sciences 178 (2008) 2680–2704.
[59] M. Mahdavi, M.H. Chehreghani, H. Abolhassani, R. Forsati, Novel meta-heuristic algorithms for clustering web documents, Applied Mathematics and Computation 201 (2008) 441–451.
[60] V.J. Manoj, E. Elias, Artificial bee colony algorithm for the design of multiplier-less nonuniform filter bank transmultiplexer, Information Sciences 192 (2012) 193–203.
[61] Y. Marinakis, M. Marinaki, G. Dounias, Honey bees mating optimization algorithm for the Euclidean traveling salesman problem, Information Sciences 181 (2011) 4684–4698.
[62] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognition 33 (2000) 1455–1465.
[63] U. Maulik, S. Bandyopadhyay, Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification, IEEE Transactions on Geoscience and Remote Sensing 41 (2003) 1075–1081.

[64] W. Mendenhall, R.J. Beaver, B.M. Beaver, Introduction to Probability and Statistics, 12th ed., 2010.
[65] S. Mitra, P.P. Kundu, Satellite image segmentation with shadowed *C*-means, Information Sciences 181 (2011) 3601–3613.
[66] C.A. Murthy, N. Chowdhury, In search of optimal clusters using genetic algorithms, Pattern Recognition Letters 17 (1996) 825–832.
[67] T. Niknam, B. Amiri, An efficient hybrid approach based on PSO, ACO and *K*-means for cluster analysis, Applied Soft Computing 10 (2010) 183–197.
[68] P. Papajorgji, R. Chinchuluun, W.S. Lee, J. Bhorania, P.M. Pardalos, Clustering and classification algorithms in food and agricultural applications: a survey, in: Advances in Modeling Agricultural Systems, Springer, US, 2009, pp. 433–454.
[69] N.H. Park, S.H. Oh, W.S. Lee, Anomaly intrusion detection by clustering transactional audit streams in a host computer, Information Sciences 180 (2010) 2375–2389.
[70] D. Picard, A. Revel, M. Cord, An application of swarm intelligence to distributed image retrieval, Information Sciences 192 (2012) 71–81.
[71] C. Pickover, Black Holes: A Traveler's Guide, John Wiley & Sons, 1998.
[72] A.K. Qin, P.N. Suganthan, Robust growing neural gas algorithm with application in cluster analysis, Neural Networks 17 (2004) 1135–1148.
[73] A.K. Qin, P.N. Suganthan, Kernel neural gas algorithms with application to cluster analysis, in: Proceedings – International Conference on Pattern Recognition, 2004, pp. 617–620.
[74] A.K. Qin, P.N. Suganthan, A robust neural gas algorithm for clustering analysis, in: Proceedings of International Conference on Intelligent Sensing and Information Processing, ICISIP 2004, 2004, pp. 342–347.
[75] B.Y. Qu, J.J. Liang, P.N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, Information Sciences 197 (2012) 131–143.
[76] S. Rana, S. Jasola, R. Kumar, A review on particle swarm optimization algorithms and their applications to data clustering, Artificial Intelligence Review 35 (2011) 211–222.
[77] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Information Sciences 179 (2009) 2232–2248.
[78] S.Z. Selim, M.A. Ismail, *K*-means-type algorithms: a generalized convergence theorem and characterization of local optimality, pattern analysis and machine intelligence, IEEE Transactions on PAMI 6 (1984) 81–87.
[79] J. Senthilnath, S.N. Omkar, V. Mani, Clustering using firefly algorithm: performance study, Swarm and Evolutionary Computation 1 (2011) 164–171.
[80] H. Shah_Hosseini, Problem solving by intelligent water drops, in: IEEE Congress on Evolutionary Computation, CEC 2007, 2007, pp. 3226–3231.
[81] Y.-C. Song, H.-D. Meng, M. O'Grady, G. O'Hare, The application of cluster analysis in geophysical data interpretation, Computational Geosciences 14 (2011) 263–271.
[82] J. Wang, H. Peng, P. Shi, An optimal image watermarking approach based on a multi-objective genetic algorithm, Information Sciences 181 (2011) 5501–5514.
[83] Y.-J. Wang, H.-S. Lee, A clustering method to identify representative financial ratios, Information Sciences 178 (2008) 1087–1097.
[84] A.K.C. Wong, D.K.Y. Chiu, W. Huang, A discrete-valued clustering algorithm with applications to biomolecular data, Information Sciences 139 (2001) 97–112.
[85] S. Yang, R. Wu, M. Wang, L. Jiao, Evolutionary clustering based vector quantization and SPIHT coding for image compression, Pattern Recognition Letters 31 (2010) 1773–1780.
[86] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.
[87] W.-C. Yeh, Novel swarm optimization for mining classification rules on thyroid gland data, Information Sciences 197 (2012) 65–76.
[88] J. Zhang, K. Liu, Y. Tan, X. He, Random black hole particle swarm optimization and its application, in: 2008 IEEE International Conference Neural Networks and Signal Processing, ICNNSP, 2008, pp. 359–365.
[89] L. Zhang, Q. Cao, A novel ant-based clustering algorithm using the kernel method, Information Sciences 181 (2010) 4658–4672.
[90] Y. Zhang, D.-W. Gong, Z. Ding, A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch, Information Sciences 192 (2012) 213–227.