Int. J. Mach. Learn. & Cyber. (2016) 7:773–782
DOI 10.1007/s13042-014-0294-5

ORIGINAL ARTICLE

Data clustering using multivariant optimization algorithm

Qin-Hu Zhang · Bao-Lei Li · Ya-Jie Liu · Lian Gao · Lan-Juan Liu · Xin-Ling Shi

Received: 1 January 2014 / Accepted: 18 August 2014 / Published online: 26 August 2014

Abstract Data clustering is one of the most popular techniques in data mining to group data with great similarity and high dissimilarity into each cluster. This paper presents a new clustering method based on a novel heuristic optimization algorithm proposed recently and named as multivariant optimization algorithm (MOA) to locate the optimal solution automatically through global and local alternating search implemented by a global exploration group and several local exploitation groups. In order to demonstrate the performance of MOA-clustering method, it is applied to group six real-life datasets to obtain their clustering results, which may be compared with those received by employing K-means algorithm, genetic algorithm and particle swarm optimization. The results show that the proposed clustering algorithm is an effective and feasible method to reach a high accurate rate and stability in clustering problems.

Keywords Data clustering · Cluster centers · Multivariant optimization algorithm · Global and local optimization

1 Introduction

Cluster analysis has become an important technique in exploratory data analysis, pattern recognition, machine learning, image segmentation, neural computing, and other engineering [1]. In the field of clustering, K-means algorithm as a popular clustering method has been successfully applied to many practical clustering problems [2, 3]. However, the results obtained by using K-means algorithm may contain several local minima as the objective function of K-means which is not convex [4, 5]. Evolutionary algorithms such as genetic algorithm (GA) [6] and particle swarm optimization (PSO) [7] have been therefore introduced to solve such problems and widely applied to various clustering problems [8–17].

In this paper, the recently proposed discrete heuristic optimization algorithm named multivariant optimization algorithm (MOA) is adopted to solve clustering problems. In MOA, a search individual is named as an atom. The main idea of MOA is to search the solution space through alternating global–local search iterations where global exploration atoms explore the whole solution space to locate potential areas and then multiple local exploitation groups with different population are allotted to these potential areas for different levels of local exploitations. The better atoms generated in the optimization process are recorded in a data structure which is made up of a queue and some stacks, whereas the worse ones are extruded in competition. As to clustering problem, we can regard an array (atom) recording all cluster centers as a solution, and only need to search for the optimal solution in the whole solution space, so the clustering problem is converted to an optimization problem. Then the MOA can be applied to search for the optimal solution which contains all cluster centers of different classes.

For the purpose of simplifying the description, MOA, K-means, GA, and PSO based data clustering are named as MOA-clustering, K-clustering, GA-clustering, and PSO-clustering respectively in this paper.

In order to evaluate the performance of MOA-clustering, some comparative experiments on MOA-clustering,

Q.-H. Zhang · B.-L. Li · Y.-J. Liu · L. Gao · L.-J. Liu · X.-L. Shi (✉)
Department of Electronic Engineering, School of Information,
Yunnan University, Kunming 650091, Yunnan Province, China
e-mail: xlshi@ynu.edu.cn; 15887152994@163.com

K-clustering, GA-clustering and PSO-clustering based on six datasets are conducted. The experimental results demonstrate that the MOA-clustering not only has the ability to locate the optimal solution but also has competitive performance to GA-clustering and PSO-clustering, and outperforms K-clustering in terms of the accurate rate and stability, and that it is an effective and feasible method to reach a high accurate rate and stability in clustering problems.

2 Multivariant optimization algorithm

With the development of computer technology, the memory and speed of computer had improved a lot in a rapid speed, as a result, a novel discrete evolutionary optimization algorithm making full use of the memory of computer is proposed in this paper. The idea of the proposed algorithm is inspired by the characters of computer data structure especially the ordered doubly-linked list. In the MOA search process, global search atoms explore the global space to locate potential areas and then local search atoms exploit each potential local area in detail to improve the results and then the better atoms are recorded in the structure table. After sufficient global–local search iterations under the instruction of a structure table, multiple optimal solutions are recorded in the queue of the structure table.

The search process of MOA is implemented by search atoms under the instruction of a structure. For a minimum optimization problem, the structure table illustrated in Fig. 1 is designed according to the following rules:

1. Global search atoms are recorded in the queue.
2. The fitness values of atoms recorded in the queue are increasing from the front to the rear.
3. Each node in the queue has a stack pointer which points to a stack, the depth of the stacks is descending from the left to the right.
4. Local search atoms generated in the neighborhood of the i-th global atom are recorded in the i-th stack. The fitness values of atoms recorded in each stack are increasing from the bottom to the top.

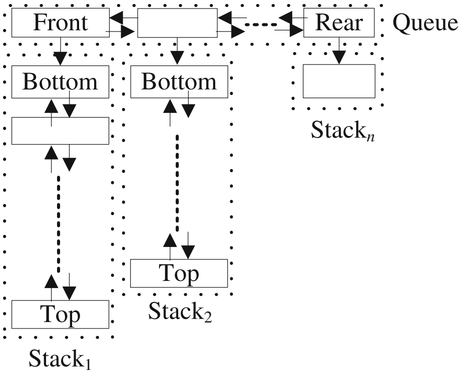


Fig. 1 Structure table of multivariant optimization algorithm

In MOA algorithm, an atom stands for a candidate solution of the optimization problem. The atoms are in two types: global search and local search atoms. Global search atoms are generated uniformly at random in the solution space. Local search atoms are generated in the neighborhood of the global atoms recorded in the queue node, which is considered as the center of a potential area. In a D-dimensional solution space, the global search atoms denoted as $atom_g$ are generated according to the Eq. (1):

$$atom_g = \{unifrnd(min_1, max_1), \ldots, unifrnd(min_D, max_D)\} \tag{1}$$

where min_i and max_i are the lower and the upper bounds of the i-th dimension of solution space, which are determined by calculating the minimum and maximum of the i-th dimension in datasets. The function $unifrnd(min_i, max_i)$ returns a random number which is uniformly distributed on the interval from min_i to max_i, so $atom_g$ is a vector.

The local search atoms denoted as $atom_l$ are generated in each corresponding global atom's neighborhood with radius R according to the Eq. (2):

$$atom_l = \begin{cases} atom_g + r * R * \dfrac{[h1, \ldots, h_D]}{\sqrt{\sum\limits_{i=1}^{D} h_i^2}} \left(\sum\limits_{i=1}^{D} h_i^2 \neq 0\right) \\ atom_g \left(\sum\limits_{i=1}^{D} h_i^2 = 0\right) \end{cases} \tag{2}$$

where $atom_g$ and R are the center vector and radius of the neighborhood respectively, $[h_1,\ldots,h_D]$ is a vector with random numbers uniformly distributed on $[-1,1]$, and $[h_1, \ldots, h_D] \Big/ \sqrt{\sum_{i=1}^{D} h_i^2}$ is a unit vector, so $R^*[h_1, \ldots, h_D] \Big/ \sqrt{\sum_{i=1}^{D} h_i^2}$ stands for a circle with radius R, r is a random number between 0 and 1. The definition of radius R is as follows: let $f(x)$ be defined on D and there exists a global optimal point $atom_g \in D$. We say that $f(x)$ has a limit as x tends to $atom_g$ provided that there exists an optimum $A \in R$ such that for every $\varepsilon > 0$, no matter how small, there always exists $R > 0$ such that for every $x \in D$,

$$\begin{cases} 0 < |x - atom_g| < R\ .implies.|f(x) - A| < \varepsilon \\ \lim\limits_{iterations- > \infty} P(|f(x) - A| < \varepsilon) = 1 \end{cases}$$

Further, in the case that such an optimum A exists, we shall say that A is the limit of $f(x)$ as x tends to $atom_g$. In

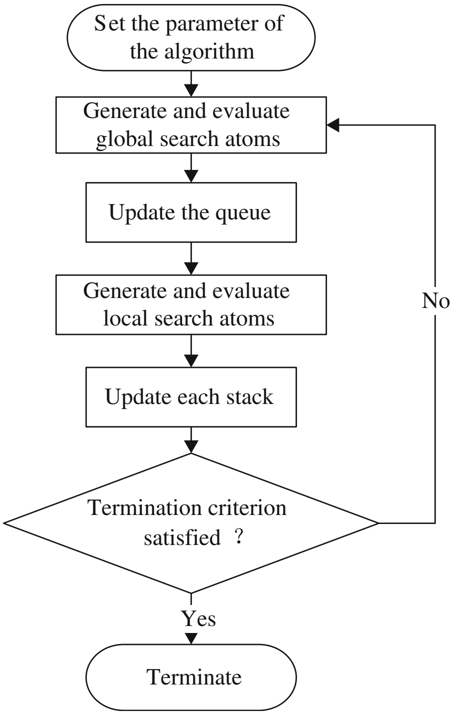

Set the parameter of the algorithm
Generate and evaluate global search atoms
Update the queue
Generate and evaluate local search atoms
Update each stack
Termination criterion satisfied ?
No
Yes
Terminate

Fig. 2 Flowchart of the MOA algorithm

addition, $f(x)$ will gradually converge to the optimum A with probability 1 when the number of iterations is enough.

The MOA algorithm searches the solution space by the following steps which are illustrated by Fig. 2:

*Step*1: set the initial parameters of the MOA algorithm: the length of queue, the depth of each steak, the scope of neighborhood and the maximum number of iterations

*Step*2: generate and evaluate global search atoms. At the beginning of iteration, a number of new global search atoms are generated and then their fitness values are evaluated

*Step*3: update the queue. Compared the fitness value of each new global atom with the atoms in the queue, if a new atom is better enough to be recorded in the queue, a new queue node which records this atom should be inserted into the queue following the same logic patterns used in a double linked general ordered list and the rear node should be deleted to keep the length of the queue fixed

*Step*4: generate and evaluate local search atoms. For each stack, the local search atoms whose number equals the depth of the stack are generated in the neighborhood of their corresponding global atoms in the queue and then their fitness values are evaluated by fitness function

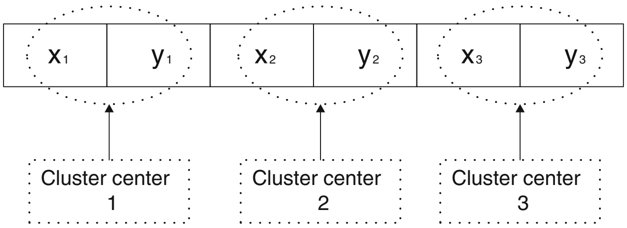


Fig. 3 The encoding of a single atom

*Step*5: update each stack. Compared the fitness value of each new atom with the atoms in the corresponding stack, if a new atom is better enough to be recorded in the stack, a new stack node which records this atom should be inserted into the stack. If the number of nodes in the stack is bigger than the depth of the stack, the redundant nodes should be deleted. If the best atom in the i-th stack is better than the i-th global atom, they should be replaced each other

*Step*6: check the termination criterion. If the termination criterion is satisfied, the algorithm will stop. Otherwise, return to *Step*2

3 Application of multivariant optimization algorithm

3.1 Encoding the search atoms

From the above description of MOA, we can know the main encoded objects are the search atoms, including global and local search atoms. In the context of clustering, a single atom represents all cluster centers. That is, in a n-dimensions space, the i-th atom is encoded as follows:

$$xi = (x^i_{11}, x^i_{12}, \ldots, x^i_{1n},\ x^i_{21}, x^i_{22}, \ldots, x^i_{2n}, \ldots,\ x^i_{K1}, x^i_{K2}, \ldots x^i_{Kn})$$

where x^i_{jm} refers to the m-th value of the j-th cluster center vector in the i-th atom, K is the number of clusters. So the length of each atom is $K \times n$.

Figure 3 is an example of the encoding of a single atom at the time of producing search atoms in the MOA. Let $n = 2$, $K = 3$, i.e., the search space is two-dimension and the number of clusters is three. The vector of this atom represents three cluster centers [(x_1, y_1) (x_2, y_2) (x_3, y_3)].

According to the encoding of a single atom, every atom produced during the process of searching stands for a candidate solution to optimization problem, so that we can make the best of MOA to search the optimal solution in the n-dimensions solution space as an optimization problem.

3.2 Designing the evaluation function

The design of the evaluation function of MOA-clustering is a key issue of applying MOA to clustering, the main role of it is to evaluate whether the atom gotten during the process of searching is a better cluster center or not, if a new atom is better than the old one during the process of updating, then the old one will be replaced by the new one.

In this paper, we design the evaluation fitness following two criteria as follows:

1. The inner-cluster distance as defined in Eq. (3), i.e. the distance between data vectors and their corresponding cluster center within a cluster, where the objective is to minimize the inner-cluster distance.

$$J1 = \sum_{j=1}^{K} \sum_{\forall xi \in Zj} \|xi - zj\|^2 \tag{3}$$

where z_j is the j-th cluster center, K is the number of clusters, x_i denotes data points belonging to z_j.

2. The inter-cluster distance as defined in Eq. (4), i.e. the distance between all cluster centers, where the objective is to maximize the distance between clusters.

$$J2 = \sum_{i=1}^{K} \sum_{j=i+1}^{K} \|zi - zj\|^2 \tag{4}$$

where z_i, z_j are the i-th and j-th cluster center respectively, K is the number of clusters.

According to the two criteria, the evaluation function is designed as defined in Eq. (5), where generally the objective is to minimize the value of evaluation function.

$$\text{fitness} = w_1 \times J_1 - w_2 \times J_2 \tag{5}$$

where w_1, w_2 are the weight coefficient of J_1 and J_2 respectively, which decide the influence of J_1 and J_2 in evaluation. If w_1 is bigger than w_2, it means that J_1 decides the result of evaluation in a large part. Through a series of experiments, the clustering results are relative stabile and better when $w_1 = 0.8$, $w_2 = 0.2$. So in this paper, we set $w_1 = 0.8$, $w_2 = 0.2$.

3.3 MOA-clustering

After encoding the search atoms and designing the evaluation function, the execution of MOA-clustering is as follow:

*Step*1: set the initial parameters of the MOA algorithm: the length of queue, the depth of each steak, the scope of neighborhood and the maximum number of iterations

*Step*2: generate and evaluate global search atoms. At the beginning of iteration, a number of new global search atoms are generated randomly as the above description of encoding the search atoms in Fig. 3 and all data points x_i are assigned to their corresponding cluster centers z_j with the shortest distance between x_i and z_j according to Euclidean distance as defined in Eq. (6), then their fitness values are evaluated as the definition of the Eq. (5)

$$D = \|xi - zj\|, \; i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, K \tag{6}$$

where N is the number of dataset, K is the number of clusters

*Step*3: update the queue. According to the principle that the minimum is the best, compared the fitness value of each new global atom with the atoms in the queue, if a new atom is better than the worst one in the queue, a new queue node which records this atom should be inserted into the queue following the same logic patterns used in a double linked general ordered list and the node where the worst atom is recorded should be deleted to keep the length of the queue fixed

*Step*4: generate and evaluate local search atoms as same as *Step*2. For each stack, the local search atoms whose number equals the depth of the stack are generated in the neighborhood of their corresponding global atoms in the queue and then their fitness values are evaluated by fitness function

*Step*5: update each stack. According to the principle that the minimum is the best, compared the fitness value of each new atom with the atoms in the corresponding stack, if a new atom is better enough to be recorded in the stack, a new stack node which records this atom should be inserted into the stack. If the number of nodes in the stack is bigger than the depth of the stack, the redundant nodes should be deleted. If the best atom in the i-th stack is better than the i-th global atom, they should be replaced each other

*Step*6: check the termination criterion. If the termination criterion is satisfied, the algorithm will stop. Otherwise, return to *Step*2

*Step*7: The best atom is obtained from the structure table of MOA, which contains the optimal cluster center vector. Then all data points again are assigned to their corresponding cluster centers according to the Eq. (6). Finally, the accurate rate of clustering results is calculated

Table 1 The brief description of six real-life datasets

Datasets	Total no. of dataset	No. of attributes	No. of classes	No. of instances in each class
Haberman's survival	306	3	2	306 (225, 81)
Iris	150	4	3	150 (50, 50, 50)
Vertebral column	310	6	2	310 (210, 100)
Cancer	683	9	2	683 (444, 239)
CMC	1,473	9	3	1,473 (629, 333, 511)
Wine	178	13	3	178 (59, 71, 48)

4 Experiments

This section compares the clustering results of K-clustering, GA-clustering, PSO-clustering and MOA-clustering on six real-life datasets to verify the performance of MOA-clustering.

4.1 Datasets

Six experimental datasets including Haberman's Survival, Iris, Vertebral Column, Wisconsin Breast Cancer, Contraceptive Method Choice and Wine are used to assess the performance of the respective clustering methods. All datasets are available at http://archive.ics.uci.edu/ml/index.html/ and listed in Table 1 and described briefly as follows:

1. Haberman's Survival Dataset ($n = 306$, $d = 3$, $k = 2$) consists of 306 objects characterized by three features: Age of patient at time of operation, Patient's year of operation, Number of positive axillary nodes detected. The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. There are two categories in the data: the patient survived 5 years or longer (225 objects) and the patient died within 5 year (81 objects).
2. Fisher's Iris Dataset ($n = 150$, $d = 4$, $k = 3$) consists of three different species of iris flowers: iris setosa, iris virginica and iris versicolour. For each species, 50 samples were collected from four features, namely sepal length and width as well as petal length and width.
3. Vertebral Column Dataset ($n = 310$, $d = 6$, $k = 2$) consists of 310 objects characterized by six features: pelvic incidence, pelvic tilt, lumbar lordosis angle, sacral slope, pelvic radius and grade of spondylolisthesis. Dataset containing values for six biomechanical features used to classify orthopaedic patients into two classes: normal (210) and abnormal (100).
4. Wisconsin Breast Cancer Dataset ($n = 683$, $d = 9$, $k = 2$) consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. There are two categories in the data: malignant tumors (444 objects) and benign tumors (239 objects).
5. Contraceptive Method Choice Dataset (CMC) ($n = 1{,}473$, $d = 9$, $k = 3$) consists of 1,473 objects characterized by nine features: Wife's age, Wife's education, Husband's education, Number of children ever born, Wife's religion, Wife's now working, Husband's occupation, Standard-of-living index, Media exposure. Dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. There are three contraceptive methods used in the data: No-use (629), Long-term (333), Short-term (511).
6. Wine Dataset ($n = 178$, $d = 13$, $k = 3$) consists of 178 objects characterized by 13 features: alcohol content, malic acid content, ash content, alkalinity of ash, concentration of magnesium, total phenols, flavanoids, non-flavanoid phenols, and proanthocyanins, and color intensity, hue and OD280/OD315 of diluted wines and pralines. These features were obtained by chemical analysis of wines that are produced in the same region in Italy but derived from three different cultivars. The quantities of objects in the three categories of the dataset are: class 1 (59 objects), class 2 (71 objects), and class 3 (48 objects).

4.2 Settings for clustering algorithms

In order to demonstrate the selection process of radius R and iteration number during data clustering, the relevant experiments with fixed iterations or radius R are carried out 10 times on three datasets termed Iris, Cancer as well as Wine and the relevant results are reported in Fig. 4 and Table 2, respectively. The relevant parameters used in the both two experiments are opted and enumerated as follows: the queue length of the upper triangular structure is 10, the depth of the i-th stack is determined by 10-i, thus the number of searching atoms is 60. Meanwhile, the number of iterations is 200 while radius R is changed from 0.1 to 10 in the fixed-iteration experiments and the radius R are set as 0.5, 3, 5, 7 as well as 9 with corresponding iterations of 200, 600, 800, 1,000 as well as 1,000 respectively in the determined-radius experiments. Figure 4 reveals that the radius R can be chosen randomly with 200 iterations and the lower average clustering accuracy for larger radius

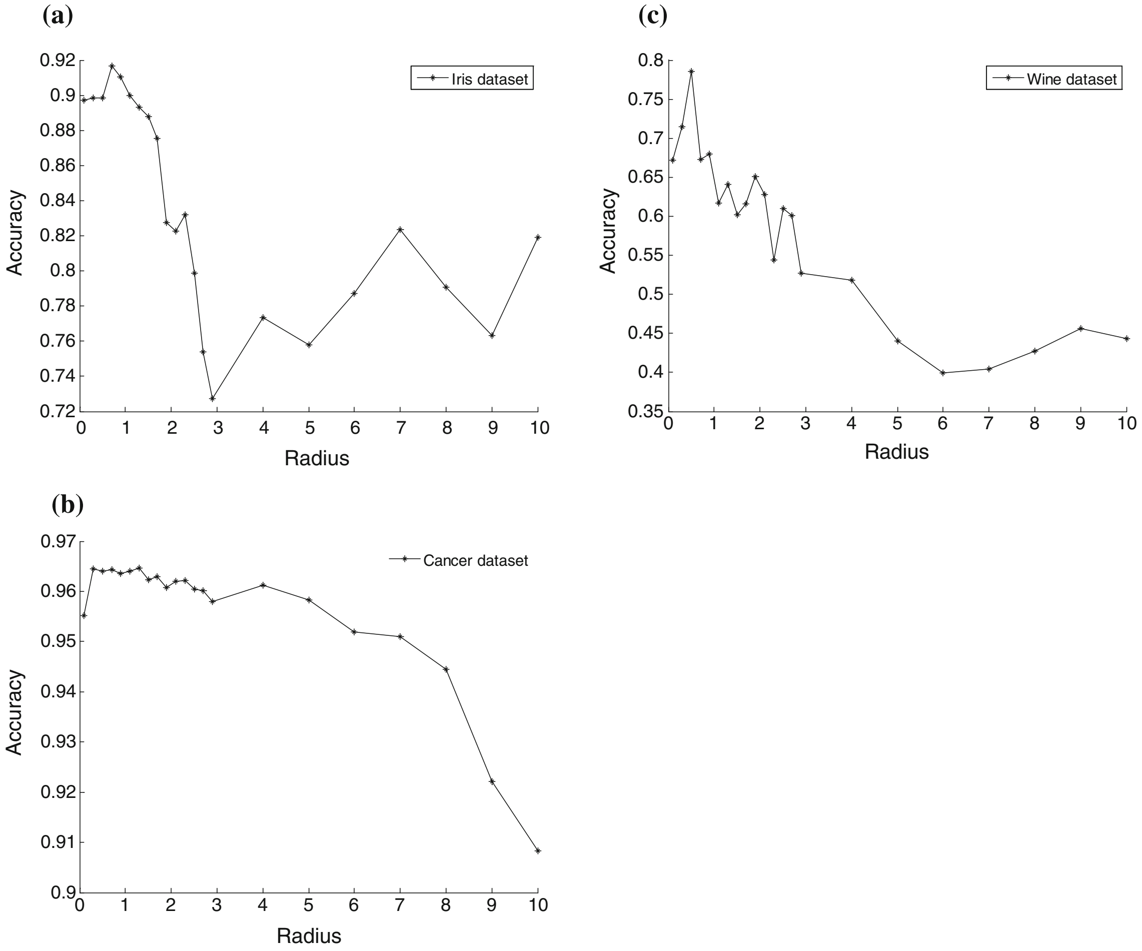


Fig. 4 The changing curve of clustering accuracy with the change of local radius R on Iris dataset (**a**), Cancer dataset (**b**) and Wine dataset (**c**) when the number of iterations is 200

Table 2 The average clustering accuracy and the standard deviation of accuracy for different combinations

Dataset	Accuracy	Radius = 0.5/ iterations = 200	Radius = 3/ iterations = 600	Radius = 5/ iterations = 800	Radius = 7/ iterations = 1,000	Radius = 9/ iterations = 1,000
Iris	Average (%)	91.27	90	90	90	89.93
	Standard deviation	0.0027	0.0021	0.0031	0.0034	0.0034
Cancer	Average (%)	96.49	96.47	96.49	96.44	96.43
	Standard deviation	4.6300e−004	4.6300e−004	4.6300e−004	9.8821e−004	0.0012
Wine	Average (%)	70.96	70.79	71.18	71.24	70.73
	Standard deviation	0.0040	0.0079	0.008	0.0079	0.0049

R can also be attributed to the limited iterations. From Table 2, it can be seen that when the iterations increases with the raising of radius R, the average clustering accuracy is almost consistent to that with lower iteration number for small radius R and the standard deviation is also very small. It means that a good and stable clustering accuracy can be obtained when the number of iterations is enough for the determined bigger radius R. For balancing the clustering accuracy and computationally cost effective in the experiments, the radius R is randomly selected as 0.5 in the interval [0.1, 1] and the number of iterations is set as 200.

The common control parameters of these algorithms are population size (P) and the number of maximum generation (max_g). In order to compare K-clustering, GA-clustering, PSO-clustering and MOA-clustering fairly, these

Table 3 the fitness of clustering results with the application of different clustering algorithm on six datasets

Datasets	Fitness	K-clustering	PSO-clustering	GA-clustering	MOA-clustering
Haberman's Survival	Best	2096.5	2,050.2	2,050.2	2,050.2
	Average	2,143.6	2,050.2	2,050.2	2,050.2
	Worst	2,553.9	2,050.2	2,050.2	2,050.2
	Standard deviation	140.31	0	0	0
Iris	Best	97.33	96.66	97.66	96.66
	Average	106.24	97.80	98.11	96.78
	Worst	123.97	110.79	98.54	97.28
	Standard deviation	12.46	3.12	0.36	0.21
Vertebral Column	Best	7,274	7,154.8	7,154.8	7,154.8
	Average	7,176.8	7,797.1	7,411.1	7,154.8
	Worst	7,279.6	9,750.1	9,717.6	7,154.9
	Standard deviation	2.87	1,141.5	788.79	0.05
Cancer	Best	2,987	2,976.3	2,965.8	2,965.3
	Average	2,988	3,194.7	2,968.2	2,966.5
	Worst	2,988.4	3,543.2	2,972.5	2,968.8
	Standard deviation	0.69	41.11	2.11	0.98
CMC	Best	4,425.8	4,420	4,419.1	4,418.1
	Average	4,427.1	4,424.5	4,421.9	4,418.5
	Worst	4,428.3	4,444.6	4,427.5	4,420.5
	Standard deviation	1.24	6.22	2.45	0.5
Wine	Best	76.72	75.50	76.50	76.13
	Average	78.46	75.60	76.51	76.49
	Worst	85.44	75.87	76.52	76.90
	Standard deviation	3.57	0.12	0.01	0.12

Table 4 The accurate rate of clustering results with the application of different clustering algorithm on six datasets

Datasets	Accurate rate	K-clustering (%)	PSO-clustering (%)	GA-clustering (%)	MOA-clustering (%)
Haberman's survival	Best	51.96	51.96	51.96	51.96
	Average	48.43	51.96	51.96	51.96
	Worst	24.18	51.96	51.96	51.96
Iris	Best	89.33	90	92.67	92.67
	Average	75.63	89.40	91.20	91.27
	Worst	48.67	78	90	90
Vertebral column	Best	67.1	71.29	71.61	71.61
	Average	66.94	70.32	70.94	71.31
	Worst	66.77	67.42	67.42	71.29
Cancer	Best	96.19	96.63	96.48	96.63
	Average	96.09	95.95	96.45	96.49
	Worst	96.05	93.41	96.19	96.34
CMC	Best	38.42	38.49	38.49	38.49
	Average	37.92	38.35	38.34	38.41
	Worst	37.47	37.95	37.95	38.02
Wine	Best	70.22	71.91	71.91	71.91
	Average	67.61	71.45	71.66	70.96
	Worst	56.74	70.79	71.35	70.22

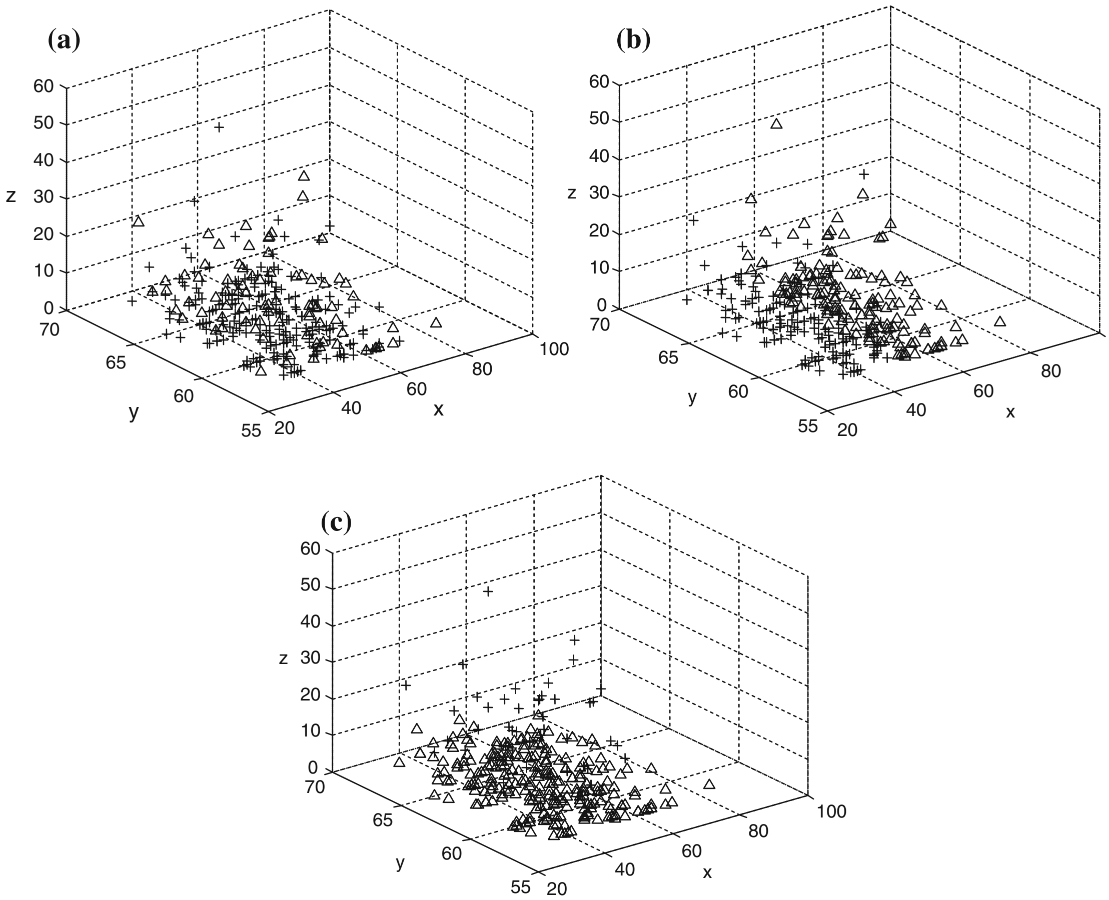
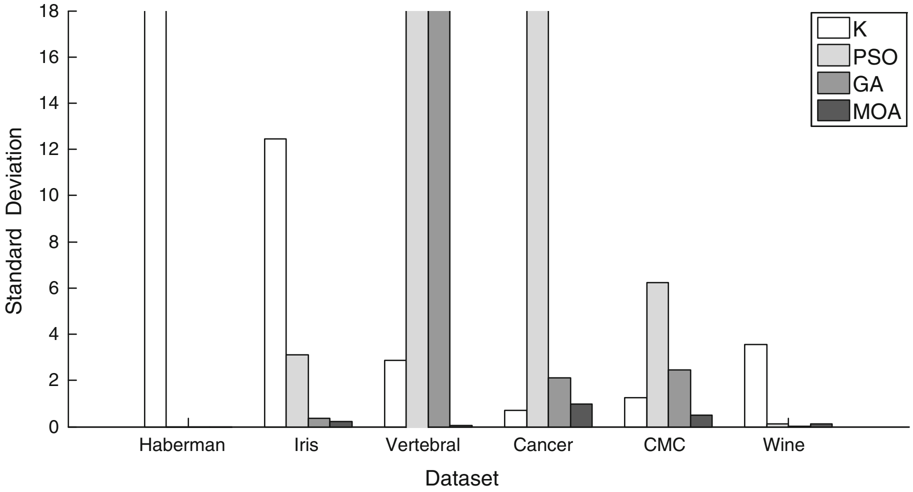


Fig. 5 The results of four algorithms on Haberman's Survival dataset before (**a**) and after (**b**, **c**) clustering

Fig. 6 The standard deviation of fitness



methods use the same common control parameter values which are denoted as $P = 60$, $max_g = 200$. Other control parameters of GA-clustering and PSO-clustering are presented below.

For genetic algorithms, we have used the standard version with no elitism, a mutation probability of 0.05 and a crossover probability of 0.95 [8]. For PSO, we have also used the standard version with the inertia weight is 0.7. The learning factors are set to 2 without the inertia correction [17]. We have used a fixed population size of $n = 60$ in all our simulations for all methods except K-clustering.

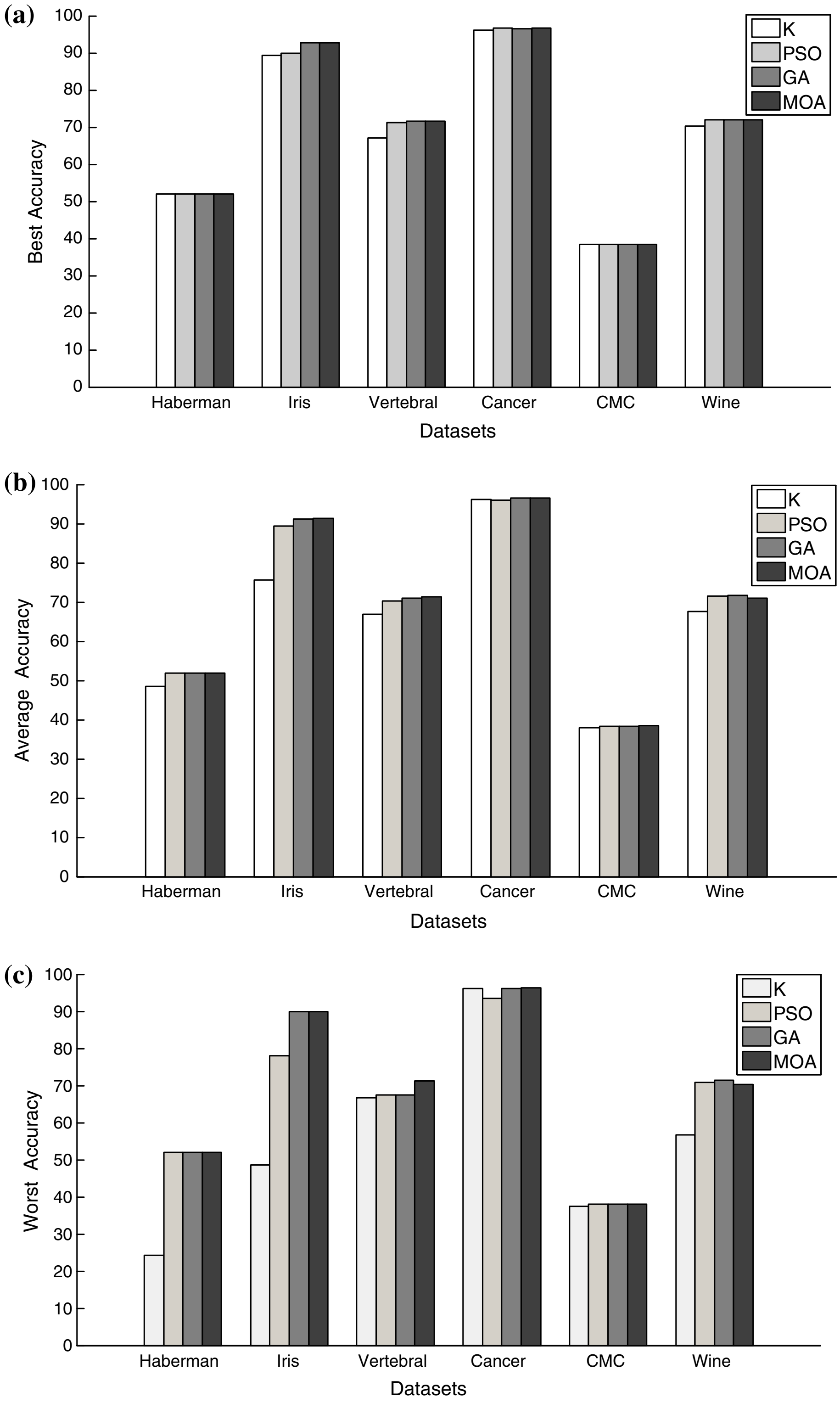


Fig. 7 The best accurate rate (**a**), the average accurate rate (**b**) and the worst accurate rate (**c**)

4.3 Experimental results and discussion

In order to evaluate the proposed algorithm, two criteria are used: the fitness as defined in Eq. (5) and the accurate rate as defined in Eq. (7).

$$Accuracy = \left(\sum_{i=1}^{N} \begin{cases} 1\, Ai = Ai^{*} \\ 0\, Ai \neq Ai^{*} \end{cases} \right) \Big/ N \times 100\% \qquad (7)$$

where N is the total number of data points. A_i and Ai^{*} are the i-th data point before and after clustering.

The performance of MOA-clustering and other clustering methods is compared by the two criteria. The fitness of all four algorithms on six datasets is summarized in Table 3, including the best fitness, the average fitness, the worst fitness, and the standard deviation of fitness over 20 simulations. Figure 6 shows the standard deviation of fitness with bar chart. Table 4 summarizes the best accurate rate, the average accurate rate and the worst accurate rate obtained from the four clustering algorithms on six datasets over 20 simulation runs. Figure 7a–c shows the best accurate rate, the average accurate rate and the worst accurate rate respectively with bar charts. The clustering results of four algorithms on Haberman's Survival dataset are shown in Fig. 5. Figure 5a shows the position of the dataset in 3-dimension space before clustering, Fig. 5b shows the result of MOA-clustering, GA-clustering and PSO-clustering on this dataset after clustering with 51.96 % accurate rate, Fig. 5c presents the result of K-clustering on this dataset after clustering with 24.18 % accurate rate.

According to the results displayed in Table 3 and Fig. 6, K-clustering is very unstable because of its least standard deviation on Cancer dataset and most standard deviation on Haberman's Survival, Iris and Wine datasets. PSO-clustering is not very stable because it has a large fluctuation on Vertebral Column and Cancer datasets in terms of standard deviation. GA-clustering and MOA-clustering proposed in this paper are relative stable, however MOA-clustering outperforms GA-clustering on Vertebral Column, Iris Cancer and CMC datasets according to the standard deviation of fitness. So on the whole, MOA-clustering is more stable than other algorithm.

The results in Table 4 and Fig. 7 clearly show that the MOA-clustering has the ability to locate the optimal solution, because the proposed method can get the best accuracy on all datasets in comparison of other algorithms, which means it can locate the best cluster centers. According to the accurate rate of clustering results, MOA-clustering obviously outperforms K-clustering on all datasets. What's more, MOA-clustering has competitive performance to GA-clustering and PSO-clustering in terms of accurate rate on Haberman's Survival and Wine datasets, and even has better accurate rate than GA -clustering and PSO-clustering on other remaining datasets.

To sum up, MOA -clustering is capable of reaching a high accurate rate and stability in clustering problems compared with K-means, GA, and PSO on the fitness and accurate rate of clustering results.

5 Conclusion

This paper provides a new clustering method based on Multivariant Optimization Algorithm (MOA). Six real-life datasets are used to investigate the performance of MOA-clustering. The experimental results demonstrate that the proposed clustering algorithm is an effective and feasible method to reach a high accurate rate and stability in clustering problems.

References

1. Evangelou IE, Hadjimitsis DG, Lazakidou AA, Clayton C (2001) Data mining and knowledge discovery in complex image data using artificial neural networks. Workshop on Complex Reasoning an Geographical Data, Cyprus
2. Selim SZ, Ismail MA (1984) K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. IEEE Trans Pattern Anal Mach Intell 6:81–87
3. Hitendra Sarma T, Viswanath P, Eswara Reddy B (2013) A hybrid approach to speed-up the k-means clustering method. Int J Mach Learn Cybernet 4(2):107–117
4. Jinxin D, Minyong Qi (2009) A new algorithm for clustering based on particle swarm optimization and K-means. IEEE Int Conf Artif Intell Comput Intell 4:264–268
5. Kao YT, Zahara E, Kao IW (2008) A hybridized approach to data clustering. Expert Syst Appl 34(3):1754–1762
6. Filho JLR, Treleaven PC, Alippi C (1994) Genetic algorithm programming environments. IEEE Comput 27:28–43
7. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks (ICW). vol IV, Perth, Australia, pp 1942–1948
8. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3–2:95–99
9. Maulik U, Bandyopadhyay S (2000) Genetic algorithm based clustering technique. Pattern Recogn 33:1455–1465
10. Chiou YC, Lan LW (2001) Theory and methodology genetic clustering algorithms. Eur J Oper Res 135:413–427
11. Merwe VD, Engelbrecht AP (2003) Data clustering using particle swarm optimization. In: Proceedings of IEEE congress on evolutionary computation 2003 (CEC 2003), Canbella, Australia, pp 215–220
12. Rana S, Jasola S, Kumar R (2013) A boundary restricted adaptive particle swarm optimization for data clustering. Int J Mach Learn Cybernet 4(4):391–400
13. Cui XH, Potok TE, Palathingal P (2005) Document clustering using particle swarm optimization. IEEE swarm intelligence symposium 2005. Pasadena, California, pp 185–191
14. Chen CY, Ye F (2004) Particle swarm optimization algorithm and its application to clustering analysis. International conference on networking, sensing control Taipei, Taiwan, March 21–23
15. Omran M, Engelbrecht AP, Salman A (2005) Particle swarm optimization method for image clustering. Int J Pattern Recogn Artif Intell 19(3):297–322
16. Chuang LY, Hsiao CJ, Yang CH (2011) Chaotic particle swarm optimization for data clustering. Expert Syst Appl 38(12): 14555–14563
17. Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings, evolutionary computation