SCIENCE CHINA
Technological Sciences

Special Issue: Celebrating the 70th anniversary of *Science China*
•Article•

https://doi.org/10.1007/s11431-020-1587-y

Multi-objective clustering analysis via combinatorial pigeon inspired optimization

CHEN Lin[1], DUAN HaiBin[1,2*], FAN YanMing[3] & WEI Chen[1]

[1] *Bio-inspired Autonomous Flight Systems Research Group, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China;*
[2] *Peng Cheng Laboratory, Shenzhen 518000, China;*
[3] *Shenyang Aircraft Design and Research Institute, Aviation Industry Corporation of China (AVIC), Shenyang 110035, China*

Received January 3, 2020; accepted March 31, 2020; published online May 20, 2020

Multi-objective data clustering is an important issue in data mining, and the realization of data clustering using the multi-objective optimization technique is a significant topic. A combinatorial multi-objective pigeon inspired optimization (CMOPIO) with ring topology is proposed to solve the clustering problem in this paper. In the CMOPIO, a delta-locus based coding approach is employed to encode the pigeons. Thus, the length of pigeon representation and the dimension of the search space are significantly reduced. Thereby, the computational load can be effectively depressed. In this way, the pigeon inspired optimization (PIO) algorithm can be discretized with an auxiliary vector to address data clustering. Moreover, an index-based ring topology with the ability of contributing to maintain flock diversity is adopted to improve the CMOPIO performance. Comparative simulation results demonstrate the feasibility and effectiveness of our proposed CMOPIO for solving data clustering problems.

multi-objective data clustering, combinatorial multi-objective pigeon inspired optimization, delta-locus based coding, pigeon representation

Citation: Chen L, Duan H B, Fan Y M, et al. Multi-objective clustering analysis via combinatorial pigeon inspired optimization. Sci China Tech Sci, 2020, 63, https://doi.org/10.1007/s11431-020-1587-y

1 Introduction

Data clustering is one of the most significant technologies to discovery new knowledge hidden in a massive amount of data generated by current applications. Clustering analysis is the process of finding homogeneous groups of data and separating heterogeneous objects without any training samples and priori knowledge. As an active subject, it provides excellent perspectives in many applications, such as marketing [1], patient stratification [2], image processing [3], network clustering [4], cluster-based routing protocol for highly mobile unmanned aerial vehicles (UAVs) network [5,6], and received signal strength indicator-based clustering for UAV integrated wireless sensor networks [7], etc.

In recent years, considerable clustering approaches have been developed for data analysis in different fields. These available methods are usually classified into partition-based clustering [8], hierarchical-based clustering [9], density-based clustering, and model-based clustering. These clustering algorithms play an important role in fostering the development of data clustering technique. A recent survey about the data clustering can be found in ref. [10].

Essentially, data clustering can be modeled as an optimization problem, so that some optimization algorithms, such as genetic algorithm (GA), differential evolution, particle swarm optimization [11], and ant colony optimization [12], can be employed to solve it. For example, discrete particle swarm optimization has been successfully employed for a

*Corresponding author (email: hbduan@buaa.edu.cn)

single objective clustering in ref. [8]. In some applications, it should be beneficial to consider several conflicting validity indices simultaneously to capture different characteristics of the datasets. In recent years, many multi-objective evolutionary algorithms have been proposed for solving the clustering problem, such as the multi-objective clustering (MOC) with automatic *k*-determination [13], multi-objective evolutionary clustering ensemble algorithm [14], MOC algorithms [15], and multi-objective *k*-means GA [16]. As mentioned above, most of the MOC algorithms are based on the multi-objective GAs [17]. Besides, some other optimization approaches including but not limited to the algorithms mentioned above have also been successfully employed for solving the MOC problem [18,19].

However, with the development of technology, the dimension and scale of the data dramatically increase. This may bring great challenges to the MOC optimization algorithms, when it comes to analyse the large-scale data, these algorithms may suffer from the curse of dimensionality. Hence it is necessary to develop effective multi-objective optimization methods to address the MOC optimization problem.

Pigeon inspired optimization (PIO) is a novel swarm intelligence technique originating from the studies of pigeon homing behaviour [20]. The search process of PIO is divided into two stages: the first stage employs the map and compass operator to search, while the landmark operator is used in the second stage. The segmented search strategy can effectively balance the explorative ability and the exploitative ability. As a population-based algorithm, PIO is able to explore multiple optima in a single iteration. Besides, compared with the most classical optimization methods, PIO has a significant advantage that it makes very few requirements about the solved problem. Hence, after the presentation of PIO, it has gained increasing attention as an efficient, simple and robust technique for solving optimization problems. Many variants of PIO have been derived and broadly applied to various fields [21], such as echo state networks for image restoration [22], explicit nonlinear model predictive control for quadrotor [23], enhanced active disturbance rejection control for the attitude deformation system of a self-developed mobile robot [24], path planning of the UAV [25], parameter design of the brushless direct current motor [26], UAV flocking control with obstacle avoidance [27], longitudinal parameters tuning of the UAV's automatic landing system [28], and energy management for parallel hybrid electric vehicle [29].

PIO has achieved great success in solving continuous optimization problems. However, as far as we known, there are few studies on solving discrete optimization problems with PIO. A binary PIO algorithm was proposed for solving multi-dimensional knapsack problem (MKP) in ref. [30]. A discrete PIO algorithm with the Metropolis acceptance criterion was proposed for large-scale traveling salesman problems [31]. These applications prove that the PIO is one of the effective methods for addressing the combinatorial optimization problems.

Motivated by the aforementioned introduction, this paper regards the data clustering as an optimization problem, then proposes a combinatorial multi-objective PIO (CMOPIO) approach with ring topology to solve it. The contributions of the paper are concluded as follows.

(1) When we extend the optimization algorithm for the MOC optimization problem, the representation of the clustering solution is the first issue to be solved. Because the popular locus-based adjacency representation methods still suffer the limitation that the encoding length increases linearly with the size of the data, the reduced-length pigeon representation is realized using the delta-locus encoding method [32]. This operation significantly depresses the search space's dimension and makes the proposed CMOPIO be an available technique to address MOC optimization problem.

(2) To address the MOC optimization problem, the CMOPIO algorithm is proposed. In the CMOPIO, a discretization technique is used to transform the classical PIO into the combinatorial PIO to foster its application to clustering analysis. Based on the combinatorial PIO, the CMOPIO approach with ring topology is developed, where the index-based ring topology is employed to enhance the ability of identifying the Pareto-optimal pigeon and promote flock diversity. The proposed CMOPIO can determine the appropriate number of clusters as well as achieve well-separated, connected and compact clusters.

The rest of the paper is organized as follows. Section 2 gives descriptions of the MOC problem. Section 3 overviews the general PIO, describes the pigeon representation approach and shows the clustering criteria for evaluating the clustering solution. Most importantly, the implementation of the CMOPIO and its application on the MOC optimization problem are provided in this section. In Section 4, simulations are carried out to demonstrate the effectiveness of the proposed approach. Finally, our concluding remarks are contained in Section 5.

2 Preliminaries and problem statement

2.1 Multi-objective clustering

Given a dataset **D** with N_{data} data in D_{dim}-dimensional space, the partition clustering problem can be modeled as

$$\begin{aligned} & c_i \neq \varnothing, i = 1, 2, \ldots, N, \\ & c_i \cap c_j = \varnothing, i \neq j, \\ & \cup_{i=1}^{k} c_i = \mathbf{D} \text{ or } \textstyle\sum_{i=1}^{k} \|c_i\| = N_{\text{data}}, \end{aligned} \tag{1}$$

where $\mathbf{C}=[c_1, c_2, \ldots, c_N]$ denotes the clustering result, N is the number of clusters, $\cup$ represents the union of two partitions, $\cap$ indicates the intersection of two partitions, $\varnothing$ is the null set, $\|c_i\|$ calculates the number of entries in c_i.

In the perspective of optimization, data clustering refers to group homogeneous data and separate heterogeneous data to optimize some criteria. As stated above, a single validity measure may be unable to trade off the whole desired performance of clustering task. Then it contributes to the MOC by considering various conflict objectives. The MOC optimization can be stated as the problem of finding a set of clustering solutions **C** such that

$$\min \mathbf{f}(\mathbf{C}) = [f_1(\mathbf{C}),\ f_2(\mathbf{C}),\ \dots,\ f_m(\mathbf{C})], \tag{2}$$

where $\mathbf{f}(\mathbf{C}) \in R^m$ is the objective function vector with $f_i(\mathbf{C})$, $i \in \{1, 2, \dots, m\}$ representing the ith clustering criterion. m is the number of objective functions. All clustering solutions satisfying corresponding constraint eq. (1) make up the solution space.

For the MOC optimization problem, with the exemplification of Figure 1, the following concepts are defined.

(1) Dominance [33]: the clustering solution $\mathbf{C}_1$ is said to dominate the solution $\mathbf{C}_2$ if and only if

$$\begin{aligned} &\forall i \in \{1, 2, \dots, m\} \quad f_i(\mathbf{C}_1) \le f_i(\mathbf{C}_2), \\ &\quad \text{and} \ \ \exists i: \ \ f_i(\mathbf{C}_1) < f_i(\mathbf{C}_2). \end{aligned} \tag{3}$$

(2) Pareto optimal solution: the clustering solution $\mathbf{C}_i$ is regarded as a Pareto optimal solution if there are not any solutions dominate it. The Pareto solution set consists of these Pareto solutions.

(3) Pareto front (PF): the maps of the non-dominated clustering solution set to the objective space constitute the Pareto front, which describes the trade-off among a variety of conflict validity measure functions.

2.2 Clustering criteria

The main issue of employing multi-objective optimization algorithm to address the MOC problem is the selection of appropriate fitness functions. In general, connectivity and compactness are the common validity measure indexes to evaluate a clustering solution [19]. In this paper, both of them are introduced as the fitness functions. Firstly, to express the concept of clustering compactness, the intracluster variance is calculated using the following equation:

$$\begin{aligned} &f_1(\mathbf{C}) = \frac{1}{N} \sum_{c_j \in \mathbf{C}} V(c_j), \\ &\text{with } V(c_j) = \sum_{i \in c_j} d(i, \mu(c_j))^2, \end{aligned} \tag{4}$$

where $\mu(c_j)$ is the center of cluster c_j, $d(i, \mu(c_j))$ refers to the Euclidean distance between the cluster's center and the node i inside in cluster c_j. Afterwards, the following equation is given to describe the clustering connectivity [32]:

$$\begin{aligned} &f_2(\mathbf{C}) = \sum_{i=1}^{N} \sum_{j=1}^{L} \eta(i,j), \\ &\text{with } \eta(i,j) = \begin{cases} j^{-1}, & \text{if no } c_k \in \mathbf{C}, \\ & \quad \text{s.t. } i \in c_k \cap nn_{ij} \in c_k, \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{5}$$

where L is the pre-designated parameter representing the size of considered neighborhood, nn_{ij} indicates the jth nearest neighbor of the node i.

Based on the above descriptions, the CMOPIO algorithm will be developed in next section to quickly locate the clustering solutions that perform excellent ability to trade off the two different clustering criteria.

3 Multi-objective clustering using CMOPIO

3.1 Basic PIO with transition factor

The basic PIO is an effective population-based optimization algorithm by stimulating the behavior of pigeon homing. It was originally proposed by Duan and Qiao [20]. In PIO, a flock of N_p pigeons is considered, each pigeon is represented by a position vector $\mathbf{p}_i=[p_{i,1},\ p_{i,2},\ \dots,\ p_{i,\text{dim}}]$ and a velocity vector $\mathbf{v}_i=[v_{i,1},\ v_{i,2},\ \dots,\ v_{i,\text{dim}}]$. The search procedure of the solution relies on two independent cycles, map and compass operator and landmark operator. The two significant opera-

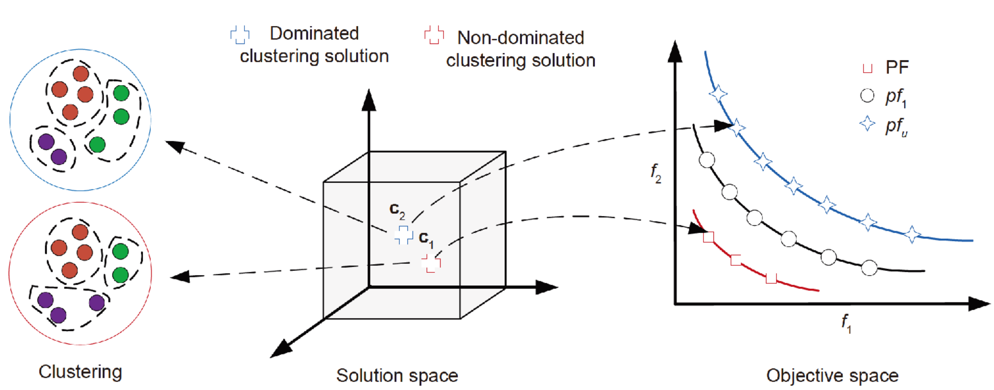

Figure 1 (Color online) Mapping of a 3D solution space to the 2D objective space.

tors of the basic PIO are merged using a transition factor α in ref. [34]. The mathematical expression of this process can be given as

$$\begin{cases} \mathbf{v}_i(t+1) = \mathrm{e}^{-R(t+1)}\mathbf{v}_i(t) \\ \qquad + \alpha r_1(1-\log_{T_{\max}}(t+1))(\mathbf{p}_{\mathrm{gbest}}(t)-\mathbf{p}_i(t)) \\ \qquad + \alpha r_2 \log_{T_{\max}}(t+1)(\mathbf{p}_{\mathrm{center}}(t)-\mathbf{p}_i(t)), \\ \mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{v}_i(t+1), \\ N_{\mathrm{p}}(t+1) = N_{\mathrm{p}}(t) - N_{\mathrm{d}}, \\ \mathbf{p}_{\mathrm{center}} = \dfrac{\sum_{j=1}^{N_{\mathrm{p}}(t)} \mathbf{p}_j(t+1)}{N_{\mathrm{p}}(t+1)}, \end{cases} \tag{6}$$

where R is the map and compass factor to control the impact of the previous velocity on the current one, t is the iteration number, r_1 and r_2 refer to the random numbers that subject to the normal distribution, $\mathbf{p}_{\mathrm{gbest}}=[p_{\mathrm{gbest},1}, p_{\mathrm{gbest},2}, \ldots, p_{\mathrm{gbest,dim}}]$ is the global historical best position of the flock. The memory $\mathbf{p}_{\mathrm{gbest}}$ and the center position $\mathbf{p}_{\mathrm{center}}=[p_{\mathrm{center},1}, p_{\mathrm{center},2}, \ldots, p_{\mathrm{center,dim}}]$ of the pigeon flock play a significant role in guiding the movement of the pigeon flock in the solution space. N_{d} is the number of the pigeons rejected in each iteration, $T_{\max}$ is the maximum number of iterations, $\log_{T_{\max}}(t)$ is the logarithmic function. As t increased, $\mathbf{p}_i(t)$ relies more on $\mathbf{p}_{\mathrm{center}}$ rather than $\mathbf{p}_{\mathrm{gbest}}$. Under the action of α, the transition between the two operators is completed smoothly.

3.2 Pigeon representation

There are several significant issues to be solved when extend the PIO for data clustering. The first issue is the representation of the clustering solution, which is described by the position of the pigeon. In recent, a locus-based adjacency representation is firstly proposed in ref. [35], its less redundant compared with the other representation approaches makes it a prevalent method for encoding. However, as described in ref. [36], the encoding method still suffers the limitation that the encoding length increases linearly with the size of the data. This may result in heavy computational burden. To address the problem, a reduced-length encoding algorithm referring as the delta-locus encoding is proposed in ref. [32] based on the obtained minimum spanning trees (MSTs) information. Motivated by this method, the pigeon representation is carried out by following the next steps.

(1) Generation of the MST

The MST is the spanning tree of a graph with the sum of all the edges' weights being the smallest. It is an effective tool for building cable networks and studying the evolutionary relations of gene sequences [37]. In this paper, the MST is applied to describe the cluster membership of the data by classifying the interconnected nodes into the same cluster. The distances between each node are used to represent the dissimilarity of the nodes. Let

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{(x_{i,1}-x_{j,1})^2 + \cdots + (x_{i,D_{\dim}} - x_{j,D_{\dim}})^2}$$

be the Euclidean distance between the node i and j. The smaller the distance d_{ij} is, the higher the similarity between these two nodes is. Calculating the distances between every node of the dataset, then sort these distances in ascending order to obtain the following adjacency matrix:

$$\mathbf{N}_{\mathrm{nearest}} = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1N_{\mathrm{data}}} \\ n_{21} & n_{22} & \cdots & n_{2N_{\mathrm{data}}} \\ \vdots & \vdots & \cdots & \vdots \\ n_{N_{\mathrm{data}}1} & n_{N_{\mathrm{data}}2} & \cdots & n_{N_{\mathrm{data}}N_{\mathrm{data}}} \end{bmatrix}, \tag{7}$$

where the tuple $[n_{j1}\ n_{j2}\ \cdots\ n_{jN_{\mathrm{data}}}]$, j=1, 2, …, N_{data}, represents the index vector of each node that differs from the node j in ascending order. According to the adjacency matrix $\mathbf{N}_{\mathrm{nearest}}$, the MST of the dataset with seven nodes can be established as shown in Figure 2(a) using the Prim's algorithm.

As shown in Figure 2(a), all the connected nodes will be clarified into the same cluster to obtain the cluster membership of the data. Then we can get the full-length representation of the clustering result as $\mathbf{r}=[i, j, \ldots, k]$, $\|\mathbf{r}\|=N_{\mathrm{data}}$. The index of the clustering result $\mathbf{r}$ indicates the number of every link's start node, and the elements of $\mathbf{r}$ represent the number of the end node for each link. For instance, the link 2→6 can be encoded as $\mathbf{r}(2)=6$.

(2) Reduced-length representation of the pigeon

Before moving on to the next step, we give the concept of similarity weight of the link $i \to j$ firstly [32]

$$sw(i \to j) = \min(nn_i(j), nn_j(i)) + d^*(i,j), \tag{8}$$

where $nn_i(j)$ indicates the ranking of the node j in the nearest neighbors of the node i. Then $a=nn_i(j)$ stands for the node j is the ath nearest neighbor of the node i. $d^*(i,j)$ is the normalized dissimilarity between the nodes i and j.

According to the similarity weight of the MST edges, the ranking of the MST edges is carried out as exemplified in Figure 2(b). Meanwhile, a parameter λ is introduced to classify the MST links into the relevant links set $\mathbf{E}_{\mathrm{r}}$ and fixed links set $\mathbf{E}_{\mathrm{f}}$. The first $\lambda\times(N_{\mathrm{data}}-1)$, $0\le\lambda\le1$ edges with higher similarity weight will be divided into $\mathbf{E}_{\mathrm{r}}$, while the rest $(1-\lambda)\times(N_{\mathrm{data}}-1)$ edges will be classified into $\mathbf{E}_{\mathrm{f}}$. In subsequent processing, only the relevant links are considered, while the fixed links are considered to be invariable, which provides general information $\mathbf{C}_{\mathrm{general}}$ for all the pigeons. Therefore, as shown in Figure 2(c), the deterministic representation of clustering solution can be obtained by removing all the relevant links. The start nodes and the end nodes of the relevant links are stored in the set $\mathbf{S}_{\mathrm{r}}$ and $\mathbf{T}_{\mathrm{r}}$, respectively. Then, a pigeon with reduced-length from N_{data} to $\lambda\times(N_{\mathrm{data}}-1)$ would be constructed by redesignating the end

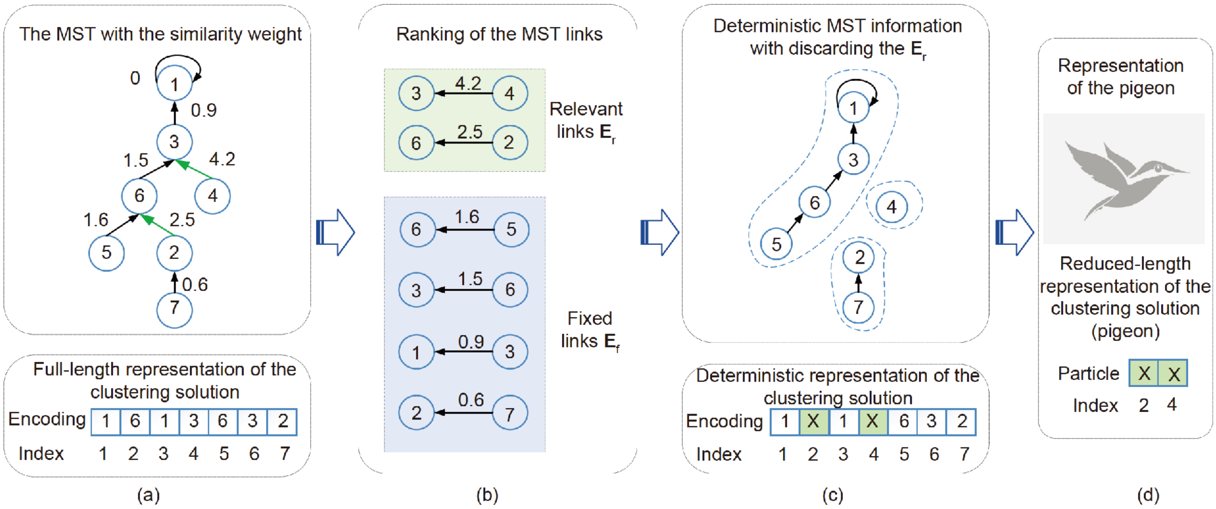

Figure 2 (Color online) Encoding of the clustering solution (pigeon).

nodes to the start nodes $\mathbf{S}_r$ of the relevant links (Figure 2(d)). The length of the pigeon can be changed by adjusting the parameter λ to satisfy different needs.

The redesignating rule can be concludes as $\forall j \in \mathbf{S}_r$, let $L<N_{data}$ be the size of considered neighborhood, randomly select a node from its neighborhood $\mathbf{N}_{nearest}(j,:)$, the selected node l will be treated as the end note corresponding to the node j if the generated link $j\rightarrow l$ satisfies $(j \rightarrow l) \notin \mathbf{E}_r$, i.e. the removed relevant link $j\rightarrow i$ will be replaced with a novel link $j\rightarrow l$, herein l is the nearest neighbor of the node j except the node i. The detailed implement of the generation of the reduced-length pigeon is given as Algorithm 1.

Algorithm 1: GeneratePigeon ($\mathbf{S}_r$, $\mathbf{T}_r$, $\mathbf{N}_{nearest}$, N_p)

Input: The sets of the relevant links' start nodes $\mathbf{S}_r$ and end nodes $\mathbf{T}_r$, adjacency matrix $\mathbf{N}_{nearest}$, the number of the pigeon flock N_p.
Parameter initialization: the size of considered neighborhood L.
for i=1:N_p
  for j=1:size($\mathbf{S}_r$)
    $p_{i,j} = N_{nearest}(\mathbf{S}_r(j), \mathrm{ceil}(L\times \mathrm{rand}(1)))$ if $p_{i,j} \neq \mathbf{T}_r(j)$
  end for
end for
Output: the position of pigeon flock $\mathbf{p}_i$, i=1, 2, …, N_p.

3.3 Pigeon evaluation

(1) Pre-evaluation of the fixed links set

It should be noted that the definition of the fixed links set allows the pre-evaluation of the common part $\mathbf{C}_{general}$ available for all clustering solutions (pigeon). Only the reduced-length pigeon needs to be evaluated during each iteration. This mechanism provides a well access to lower the computation load. Meanwhile, the reduction of search space's dimension not only depresses the difficulty of exploration, but also guarantees the searching ability and searching speed.

As given is Figure 2(c), the general cluster membership of the data can be obtained by classifying the fixed interconnected nodes into the same cluster. Then the connectivity and compactness (clustering fitness functions) of the deterministic part $\mathbf{C}_{general}$ can be obtained as $f_1(\mathbf{C}_{general})$ and $f_2(\mathbf{C}_{general})$ using eqs. (4) and (5).

(2) Evaluation of the pigeon

As shown in Figure 3, combining the deterministic representation and the reduced-length pigeon, a pigeon with full-length encoding can be reconstructed as Figure 3(c). Then a clustering result will be obtained by decoding the full-length pigeon. Since the deterministic representation is available for all the generated pigeons, after completing the pre-evaluation of the common part of the clustering solution, only the evaluation of the generated reduced-length pigeon has to be carried out by dealing with the missing information.

The position of the pigeon stores the new end nodes that has been reassigned for the relevant links' start nodes. The generation of the ith pigeon means the merging of some different clusters that are obtained in the pre-evaluation (Figure 3(b) and (c)). For a new clusters $c_m = c_i \cup c_j$ created by merging the clusters c_i and c_j, we can obtain that

$$V(c_m) = V(c_j) + \|c_j\| d(\mu(c_m), \mu(c_j)) + V(c_i) + \|c_i\| d(\mu(c_m), \mu(c_i)), \quad (9)$$

where $V(c_j)$ and $V(c_i)$ have been calculated when pre-evaluating the deterministic part of the clustering solution. They can be employed directly when we evaluate the pigeon. $\mu(c_m) = \left(\|c_j\|\mu(c_j) + \|c_i\|\mu(c_i)\right)\|c_m\|^{-1}$ is the centroid of the new cluster c_m. The connectivity fitness function $f_1(\mathbf{p}_i)$ of the pigeon i can be updated by subtracting $V(c_j)$ and $V(c_i)$ from $f_1(\mathbf{C}_{general})$, and then adding the $V(c_m)$ of the union cluster c_m to the $f_1(\mathbf{C}_{general})$. Repeating the above procedures until there are not any clusters to be merged, the pigeon's connectivity fitness function can be obtained as $f_1(\mathbf{p}_i)$.

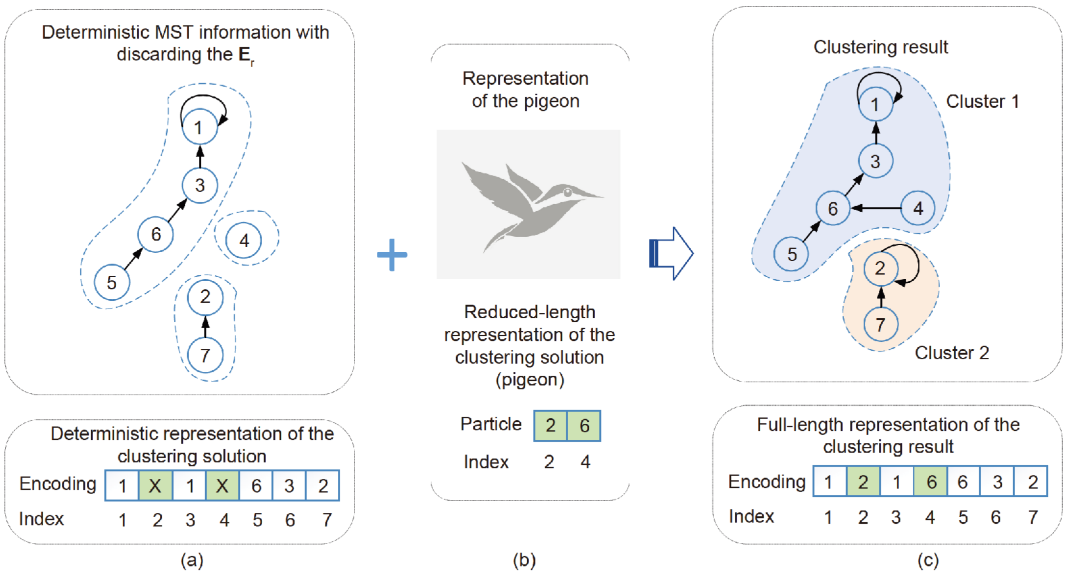

Figure 3 (Color online) Reconstruction of the full-length clustering solution.

Considering the compactness fitness function, $f_2(\mathbf{C}_{\text{general}})$ is the sum of the penalty value that the nodes and their nearest neighbors within the designated range, are not clarified into the same cluster. As mentioned before, the ith pigeon means the merging of some different clusters, which results in the consequence, some nodes and their neighbors within the specified range that does not belong to the same cluster are classified into the same category. Then the penalty imposed on these nodes should be subtracted from the $f_2(\mathbf{C}_{\text{general}})$ to get the pigeon's compactness fitness function $f_2(\mathbf{p}_i)$.

3.4 Implementation of CMOPIO

(1) Discretization of PIO

To address the MOC problem, the combinational PIO is the essential tool. In this subsection, to bridge the continuous space and the discrete space, an auxiliary vector $\boldsymbol{\zeta}_i=[\zeta_{i,1}, \zeta_{i,2}, \ldots, \zeta_{i,\text{dim}}]$ with $\zeta_{i,j} \in \{-1, 0, 1\}$, corresponding to the ith pigeon's position vector $\mathbf{p}_i$ is introduced as

$$\zeta_{i,j}(t) = \begin{cases} -1, & \text{if } p_{i,j}(t) = p_{\text{center},j}(t), \\ 1, & \text{if } p_{i,j}(t) = p_{\text{gbest},j}(t), \\ 0, & \text{otherwise}. \end{cases} \tag{10}$$

From the definition of $\boldsymbol{\zeta}_i$, the velocity update in eq. (6) can be re-expressed as

$$\begin{aligned} \mathbf{v}_i(t+1) = {} & \mathrm{e}^{-R(t+1)}\mathbf{v}_i(t) \\ & +\alpha r_1(1-\log_{T_{\max}}(t+1))(1-\boldsymbol{\zeta}_i(t)) \\ & +\alpha r_2 \log_{T_{\max}}(t+1)(-1-\boldsymbol{\zeta}_i(t)). \end{aligned} \tag{11}$$

Then the vector $\boldsymbol{\zeta}_i(t+1)$ for the next iteration time will be obtained as

$$\zeta_{ij}(t+1) = \begin{cases} 1, & \text{if } \zeta_{i,j}(t) + v_{i,j}(t+1) > \beta, \\ -1, & \text{if } \zeta_{i,j}(t) + v_{i,j}(t+1) < -\beta, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $\beta>0$. Finally, the following rule is gained to identify the entries of the new position vector of pigeon i.

$$p_{i,j}(t+1) = \begin{cases} p_{\text{gbest},j}(t), & \text{if } \zeta_{i,j}(t+1) = 1, \\ p_{\text{center},j}(t), & \text{if } \zeta_{i,j}(t+1) = -1, \\ \kappa, & \text{otherwise}, \end{cases} \tag{13}$$

where κ is randomly selected from the nearest neighbor of the node $p_{i,j}(t)$.

In the combinational PIO, the introduction of auxiliary vector $\boldsymbol{\zeta}_i$ associated with the position vector $\mathbf{p}_i$ allows the transition from the combinatorial state to the continuous state and vice versa.

(2) CMOPIO with the ring topology

In order to alleviate premature convergence and enhance the flock diversity of PIO, a ring topology is introduced in this subsection. Based on the topological structure, the CMOPIO based on the Pareto dominance ranking [38] and the crowded distance selection mechanism [39] are constructed. In the studies carried out in ref. [40], it was demonstrated that a ring topology is capable of contributing to stable niching behavior.

As shown in Figure 4, a ring topology with six pigeons is exemplified. Each pigeon interacts only with its immediate neighbors and every pigeon i possesses a local memory to store its neighborhood best position $\mathbf{p}_{\text{nbest}\_i}$. The neighborhood of the pigeon i is only composed of the pigeons $i-1$ and $i+1$, if i is the first pigeon, then its neighborhood includes the second pigeon and the last pigeon, if i is the last pigeon, then

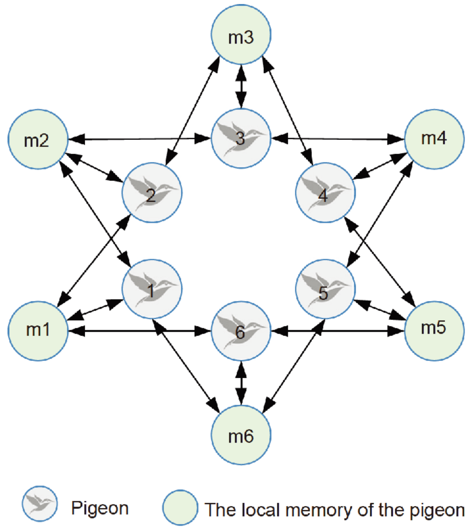

Figure 4 (Color online) CMOPIO with ring topology

that of the pigeon i consists of the first pigeon and the pigeon N_p-1. Such a ring structure effectively limits the transmission speed of information between the pigeons, hence alleviates the dilemma of premature convergence.

After the introduction of ring topology in CMOPIO, the update eqs. (10)–(13) of the ith pigeon's position and velocity will be rewritten by replacing the flock's historical best position $\mathbf{p}_{\text{gbest}}$ with its neighborhood historical best position $\mathbf{p}_{\text{nbest}\_i}=[p_{\text{nbest}\_i,1}, p_{\text{nbest}\_i,2}, \dots, p_{\text{nbest}\_i,\text{dim}}]$ and substituting the central position $\mathbf{p}_{\text{center}}$ with the local neighborhood center position $\mathbf{p}_{\text{center}\_i}=[p_{\text{center}\_i,1}, p_{\text{center}\_i,2}, \dots, p_{\text{center}\_i,\text{dim}}]$. Therefore, eqs. (10) and (13) should be re-expressed as

$$\zeta_{i,j}(t)=\begin{cases}-1, & \text{if } p_{i,j}(t)=p_{\text{center}\_i,j}(t),\\ 1, & \text{if } p_{i,j}(t)=p_{\text{nbest}\_i,j}(t),\\ 0, & \text{otherwise},\end{cases} \tag{14}$$

$$p_{i,j}(t+1)=\begin{cases}p_{\text{nbest}\_i,j}(t), & \text{if } \zeta_{i,j}(t)=1,\\ p_{\text{center}\_i,j}(t), & \text{if } \zeta_{i,j}(t)=-1,\\ \kappa, & \text{otherwise}.\end{cases} \tag{15}$$

The typical characteristic of multi-objective optimization problems is that they have several conflicting objectives and their feasible solution is not unique. There is one issue to be solved when applying CMOPIO to address the MOC problem, that is the selection of the historical personal best position $\mathbf{p}_{\text{pbest}\_i}$ and the historical neighborhood historical best position $\mathbf{p}_{\text{nbest}\_i}$. In the CMOPIO, both the Pareto dominance technique and the crowded distance selection mechanism are combined to select the best position. A neighborhood best archive NBA_i with the maximum capacity of N_{nba} is provided to store the non-dominated neighborhood historical best position within the ith pigeon's neighborhood [41]. The whole framework of the CMOPIO for solving the MOC optimization problem is summarized as Algorithm 2.

Algorithm 2: Solving framework for the MOC using the CMOPIO

Parameter initialization: map and compass factor R, the number of pigeons N_p, neighborhood best archive NBA_i, i=1,2,…, N, the maximum capacity N_{nba} of archive, historical personal best position $\mathbf{p}_{\text{pbest}\_i}$, i=1,2,…, N_p, transition factor α, maximum number of iterations $T_{\max}$, specified size of the node's nearest neighborhood L.
Input: The dataset with N_{data} nodes in D_{dim}-dimensional space.
1 Calculate the dissimilarity d_{ij} between the nodes i and j, meanwhile, rank it to generate the adjacency matrix $\mathbf{N}_{\text{nearest}}$.
2 Construct the MST using the Prim's algorithm based on $\mathbf{N}_{\text{nearest}}$.
3 Compute the similarity weight of each MST edge using eq. (8).
4 Classify these MST edges into the relevant links set $\mathbf{E}_r$ and the fixed links set $\mathbf{E}_f$ by ranking the edges' similarity weight in ascending order. Meanwhile, store the start nodes and end nodes of the relevant links in the set $\mathbf{S}_r$ and $\mathbf{T}_r$, respectively.
5 For $\mathbf{E}_f$, generate the deterministic representation $\mathbf{C}_{\text{general}}$ for all the pigeons, and carry out the pre-evaluation of $\mathbf{C}_{\text{general}}$.
6 Initialize the pigeon flock using Algorithm 1. GeneratePigeon ($\mathbf{S}_r$, $\mathbf{T}_r$, $\mathbf{N}_{\text{nearest}}$, N_p).
7 Evaluate the pigeon flock $\mathbf{f}_i=[f_1(\mathbf{p}_i), f_2(\mathbf{p}_i)]$, i=1, 2, …, N_p as described in Section 3.3.
8 Select the pigeon to be saved in NBA_i, i=1, 2, …, N_p and $\mathbf{p}_{\text{pbest}\_i}$, i=1, 2, …, N_p based on the Pareto non-dominant ranking mechanism and the crowded distance. Sort NBA_i using the Pareto dominance technique, discard the redundant tuples if its size exceeds N_{nba}.

if i=1, then $temp\_NBA_i=(\mathbf{p}_{\text{pbest}\_N_p}\cup\mathbf{p}_{\text{pbest}\_i}\cup\mathbf{p}_{\text{pbest}\_i+1})$,

if i=N_p, then $temp\_NBA_i=(\mathbf{p}_{\text{pbest}\_1}\cup\mathbf{p}_{\text{pbest}\_i}\cup\mathbf{p}_{\text{pbest}\_i-1})$,

otherwise, $temp\_NBA_i=(\mathbf{p}_{\text{pbest}\_i+1}\cup\mathbf{p}_{\text{pbest}\_i}\cup\mathbf{p}_{\text{pbest}\_i-1})$,

NBA_i=NonDominatedSort($NBA_i\cup temp\_NBA_i$),

if size(NBA_i) > N_{nba}, then $NBA_i=NBA_i(1:N_{\text{nba}},:)$,

$\mathbf{p}_{\text{nbest}\_i}$=NonDominatedSort ($\mathbf{p}_{\text{pbest}\_i}\cup\mathbf{p}_i$).

9 Update each pigeon's leader $\mathbf{p}_{\text{nbest}\_i}$, the first tuple of NBA_i is selected as $\mathbf{p}_{\text{nbest}\_i}$. NBA_i =NonDominatedSort(NBA_i), $\mathbf{p}_{\text{nbest}\_i}$=$NBA_i$(1,:).
10 Calculate the local neighborhood center position $\mathbf{p}_{\text{center}\_i}$.

if i=1, then $\mathbf{p}_{\text{center}\_i}=\text{ceil}\left((\mathbf{p}_{N_p}+\mathbf{p}_i+\mathbf{p}_{i+1})/3\right)$,

if i=N_p, then $\mathbf{p}_{\text{center}\_i}=\text{ceil}\left((\mathbf{p}_{N_p-1}+\mathbf{p}_i+\mathbf{p}_1)/3\right)$,

otherwise, $\mathbf{p}_{\text{center}\_i}=\text{ceil}\left((\mathbf{p}_{i-1}+\mathbf{p}_i+\mathbf{p}_{i+1})/3\right)$.

11 Update the velocity $\mathbf{v}_i$ and the position $\mathbf{p}_i$ of all the pigeons using eqs. (11) and (15).
12 Return to Step 7 to continue until the termination condition $t\geq T_{\max}$ is reached.
13 Output the Pareto front NBA and the Pareto optimal solution set, reconstruct and decode the Pareto optimal solution. Then the cluster of the dataset can be obtained.

In the proposed solving framework, the following main parameters play a significant role in the convergence rate and optimization results.

(1) Map and compass factor R. As given in eq. (11), R is used to control the impact of the previous velocity on the current one. The smaller R is, the larger e^{-Rt} is. This means the pigeon gains more initial velocity, which is beneficial to improve the global search ability. Conversely, a larger R indicates greater local exploration ability.

(2) Transition factor α. α is used to realize the smooth transition from the map and compass operator to the landmark operator. Meanwhile, α represents the proportion that pigeon relies on the central position of the flock $\mathbf{p}_{\text{center}}$ and individual's globally optimal location $\mathbf{p}_{\text{gbest}}$, rather than the previous velocity to adjust the position.

(3) λ is a problem-specific parameter, which determines the encoding length of the pigeons, the smaller the value, the shorter the length of pigeons. The reduced-length pigeon becomes full-length pigeon when λ=1.

(4) N_{nba} is used to limit the capacity of the neighborhood best archive.

(5) In adjacency matrix $\mathbf{N}_{\text{nearest}}$, the differences among each node are sorted in ascending order. Hence, for a given node, the similarity between it and the first L nodes is stronger than that of the rest nodes. L is used as the available size of the node's nearest neighborhood when generating a novel link to replace the removed relevant link.

4 Simulation experiments

4.1 Simulation results

In this section, simulation experiments on the clustering analysis of a series of datasets with different types and properties using the proposed CMOPIO are carried out. The spherical-type dataset, irregular-type dataset, and shape-type dataset are considered [42]. The parameters are given as in Table 1. A total of five independent executions for each problem instance are performed. Each independent execution of the clustering methods generates a set of Pareto optimal solution candidates. Some clustering solutions from the set are chosen to illustrate in the following simulation figures.

The clustering results of five different datasets are shown in Figures 5–9. Figure 5 shows the clustering result of the smile dataset, where Figure 5(a) is the exhibition of the data with the label, Figure 5(f) is the curve of the Pareto front representing the set of the optimal clustering fitness function value. According to the requirements of the clustering indexes, different optimal clustering results can be obtained. Figure 5(b) describes the clustering result corresponding to point A on the Pareto front curve that has been given in Figure 5(f). Figure 5(c)–(e) are the clustering results corresponding to points B, C and D on the Pareto front curve, respectively. From the clustering results of the smile dataset (Figure 5), the spiral dataset (Figure 6), the first square dataset (Figure 7), the second square dataset (Figure 8), and the third square dataset (Figure 9), it can be observed that the proposed clustering algorithm can effectively realize the data clustering.

To illustrate the effect of the parameter λ (relating to the length of the pigeon representation) on the performance of the CMOPIO, simulations on some of the above datasets with λ=0.2, λ=0.6, λ=0.9 are carried out. The simulation platform is MATLAB 2019a. The other parameters have the same values as those in Table 1. The running time of the program is used as the evaluation index. The results are given in Table 2. From Table 2, it can be observed that the larger the value of λ, the longer it takes to run the program. Hence, it can be proved that the employment of the reduced-length pigeon representation approach and the pre-evaluation of the fixed MST edges effectively depress the computation load.

4.2 Comparisons with the other clustering algorithms

The performance comparisons of the proposed CMOPIO algorithm with two existing prevalent cluster algorithms, the *k*-means algorithm [43] and the *k*-medoids algorithm [44], and the most popular multi-objective optimization algorithm, non-dominated sorting GA-II (NSGA-II) [45] are carried out in this subsection. The parameters such as the population size and the maximum number of iterations of NSGA-II are the

Table 1 The parameters in the proposed algorithm

Symbol	Meanings	Value
λ	The parameter for identify the length of the pigeon	0.2
N_{p}	The size of the pigeon flock	20
m	The number of fitness functions	2
$T_{\max}$	The maximum number of iterations	80
N_{nba}	The maximum capacity of the neighborhood best archive	15
α	Transition factor	3
L	The specified size of the node's nearest neighborhood	$N_{\text{data}}/6$
R	The map and compass factor	0.3

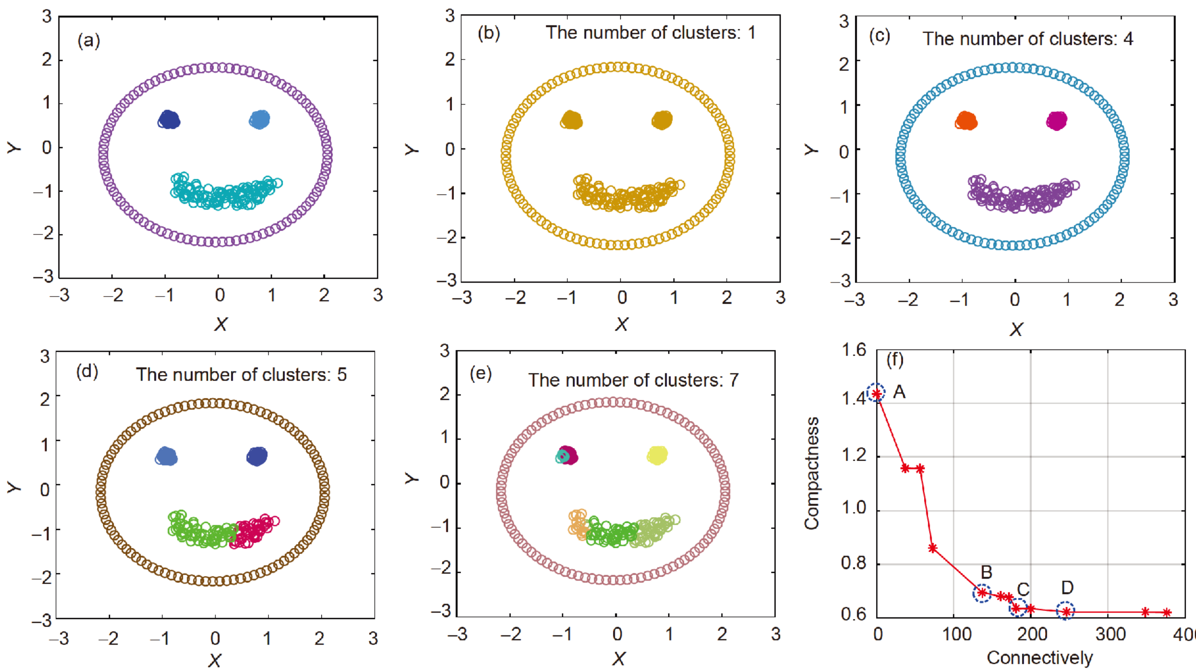

Figure 5 (Color online) Clustering results of the smile dataset. (a) The exhibition of the data with the label; (b) the clustering result corresponding to point A in (f); (c) the clustering result corresponding to point B in (f); (d) the clustering result corresponding to point C in (f); (e) the clustering result corresponding to point D in (f); (f) the curve of the Pareto front.

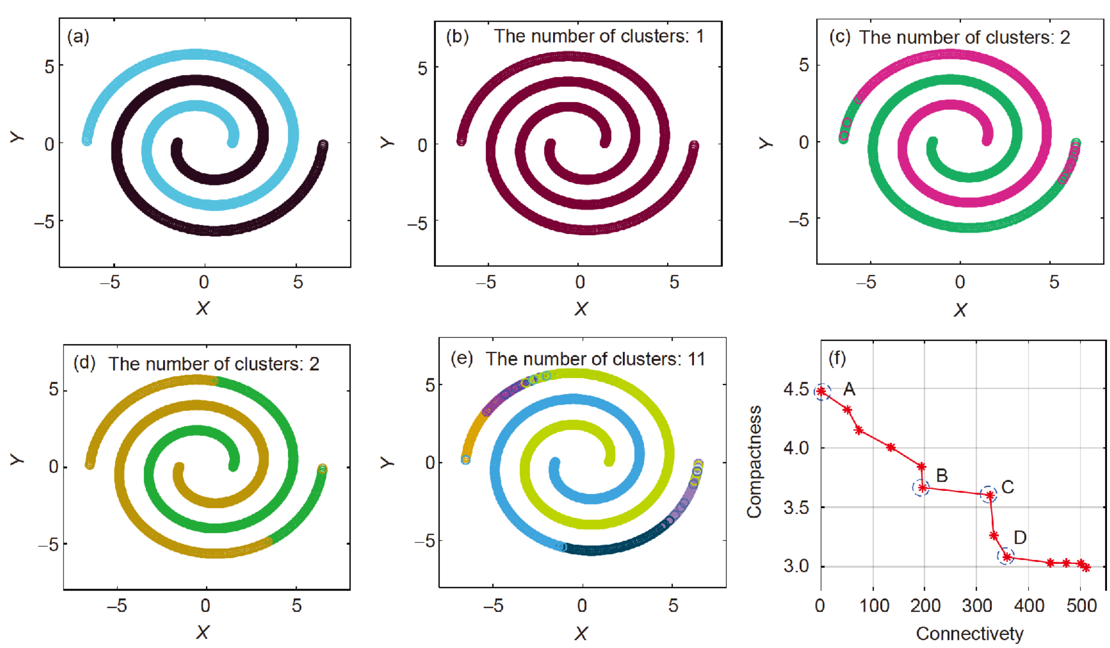

Figure 6 (Color online) Clustering results of the spiral dataset. (a) The exhibition of the data with the label; (b) the clustering result corresponding to point A in (f); (c) the clustering result corresponding to point B in (f); (d) the clustering result corresponding to point C in (f); (e) the clustering result corresponding to point D in (f); (f) the curve of the Pareto front.

same as those in Table 1. The datasets mentioned above are still employed as the test sources, which are repetitively listed in Table 3. Two evaluation indexes, the adjusted rand index (ARI) and the average silhouette coefficient (ASC), are used to evaluate the clustering results for four different clustering algorithms. The average value of the ARI and the ASC of 20 independent runs are calculated as the quantitative evaluation of every approach. The experimental results on five datasets are given in Table 3. All the four algorithms have good clustering ability when solving the clustering of these datasets.

According to Table 3, it can be obtained that although the

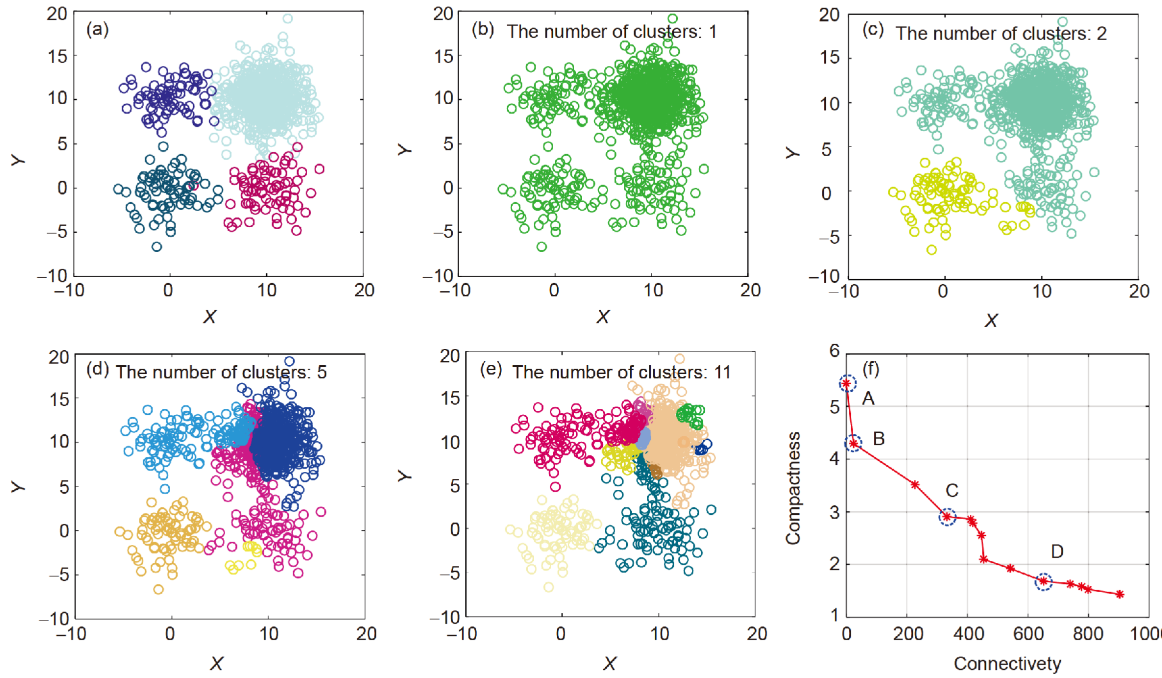

Figure 7 (Color online) Clustering results of the first square dataset. (a) The exhibition of the data with the label; (b) the clustering result corresponding to point A in (f); (c) the clustering result corresponding to point B in (f); (d) the clustering result corresponding to point C in (f); (e) the clustering result corresponding to point D in (f); (f) the curve of the Pareto front.

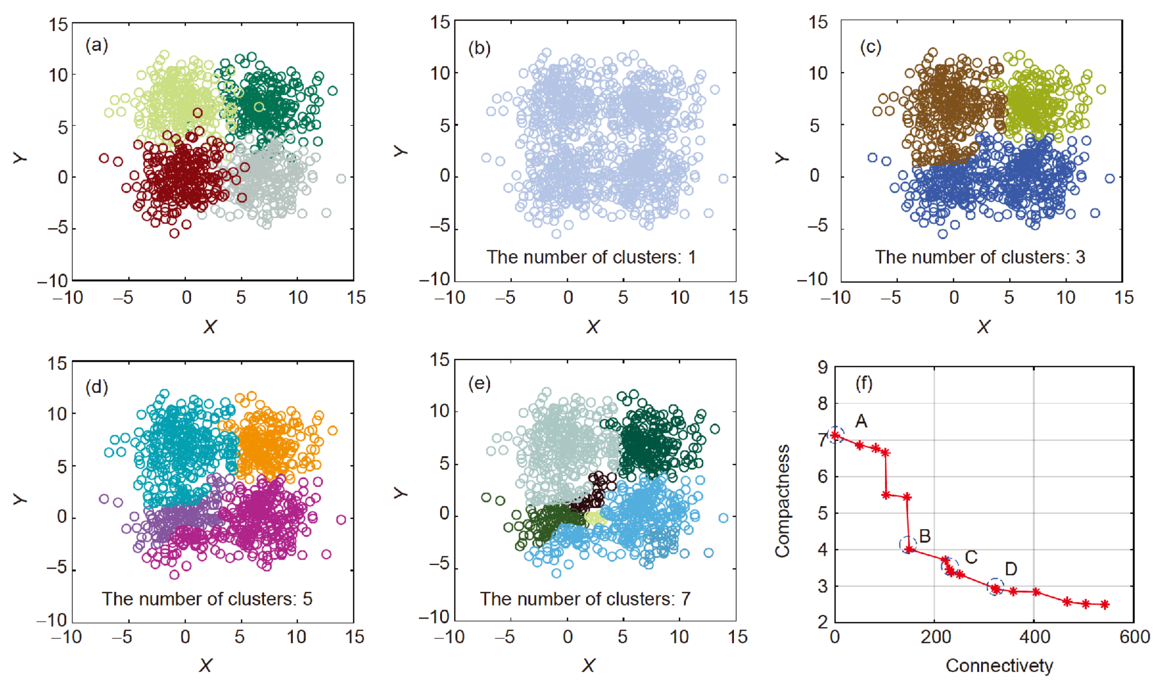

Figure 8 (Color online) Clustering results of the second square dataset. (a) The exhibition of the data with the label; (b) the clustering result corresponding to point A in (f); (c) the clustering result corresponding to point B in (f); (d) the clustering result corresponding to point C in (f); (e) the clustering result corresponding to point D in (f); (f) the curve of the Pareto front.

values of the ARI for the first square dataset and the second square dataset of the CMOPIO are smaller than that of the other two approaches. Our proposed CMOPIO algorithm has better performance than the *k*-means, NSGA-II and the *k*-medoids in terms of the ARI and the ASC.

5 Conclusions

This paper focuses on addressing the MOC problem using the proposed CMOPIO approach. The employment of the reduced-length pigeon representation approach and the pre-

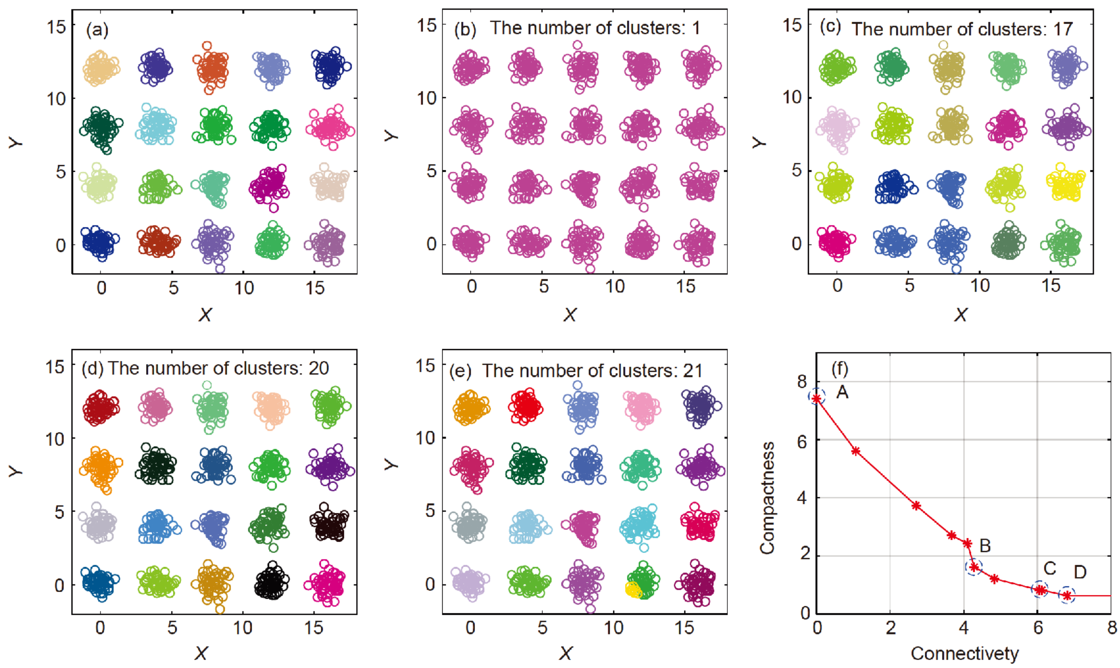

Figure 9 (Color online) Clustering results of the third square dataset. (a) The exhibition of the data with the label; (b) the clustering result corresponding to point A in (f); (c) the clustering result corresponding to point B in (f); (d) the clustering result corresponding to point C in (f); (e) the clustering result corresponding to point D in (f); (f) the curve of the Pareto front.

Table 2 The time it takes to run the program with λ=0.2/0.6/0.9

Dataset	N_{data}	N	Time (s)		
			λ=0.2	λ=0.6	λ=0.9
Smile	400	4	21.1	90.9	163.8
Spiral	1000	2	90.1	469.0	803.2
The first square	1000	4	106.9	505.3	875.9
The third square	1000	20	113.3	526.1	890.9

Table 3 ARI and ASC measurement on the clustering results of the CMOPIO, *k*-means, *k*-medoids and NSGA-II

Dataset	N_{data}	N	CMOPIO		*k*-means		*k*-medoids		NSGA-II	
			ARI	ASC	ARI	ASC	ARI	ASC	ARI	ASC
Smile	400	4	**1.0000**	0.6019	0.5449	0.6922	0.5819	0.5849	0.5725	0.6557
Spiral	1000	2	**0.8948**	0.1254	0.0359	0.5116	0.0375	0.5116	0.0522	0.4811
The first square	1000	4	0.4336	0.5045	**0.4409**	0.5032	0.2326	0.4401	0.4224	0.3571
The second square	1000	4	0.6211	0.5838	0.8324	0.5115	0.8372	0.6798	**0.9062**	0.5254
The third square	1000	20	**1.0000**	0.9185	0.8744	0.7547	0.6856	0.5511	0.3621	0.6584

evaluation of the fixed MST edges effectively depress the computation load and hence make it possible to be applied even for the large-scale datasets. Based on the discretization of PIO, the CMOPIO with a ring topology is developed. In the CMOPIO, pigeons only interact with the pigeons in their neighborhood. Meanwhile, the update of the pigeon's position and velocity relies on each pigeon's neighborhood rather than the global best position. These improvements allow the CMOPIO identify a variety of Pareto optimal clustering solutions. Finally, the tests on some different datasets verify the effectiveness of the CMOPIO algorithm.

Our future work will focus on some CMOPIO-based clustering algorithms for organizing the communication topology of the UAVs when multiple UAVs collaborate to perform the complicated missions.

This work was supported by the Science and Technology Innovation 2030-Key Project of "New Generation Artificial Intelligence" (Grant No. 2018AAA0102303), and the National Natural Science Foundation of China (Grant Nos. 91948204, U1913602, and U19B2033).

1 Xu D K, Tian Y J. A comprehensive survey of clustering algorithms.

Ann Data Sci, 2015, 2: 165–193
2 Li X T, Wong K C. Evolutionary multiobjective clustering and its applications to patient stratification. IEEE Trans Cybern, 2019, 49: 1680–1693
3 Zhao F, Fan J L, Liu H Q, et al. Noise robust multiobjective evolutionary clustering image segmentation motivated by the intuitionistic fuzzy information. IEEE Trans Fuzzy Syst, 2019, 27: 387–401
4 Gong M G, Cai Q, Chen X W, et al. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. IEEE Trans Evol Computat, 2014, 18: 82–97
5 Arafat M Y, Moh S. Localization and clustering based on swarm intelligence in UAV networks for emergency communications. IEEE Internet Things J, 2019, 6: 8958–8976
6 Cooper C, Franklin D, Ros M, et al. A comparative survey of VANET clustering techniques. IEEE Commun Surv Tutorials, 2017, 19: 657–681
7 Okcu H, Soyturk M. Distributed clustering approach for UAV integrated wireless sensor networks. Int J Ad Hoc Ubiquitous Comput, 2014, 15: 106–120
8 Jarboui B, Cheikh M, Siarry P, et al. Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. Appl Math Computat, 2007, 192: 337–345
9 Xu T S, Chiang H D, Liu G Y, et al. Hierarchical *k*-means method for clustering large-scale advanced metering infrastructure data. IEEE Trans Power Deliver, 2017, 32: 609–616
10 Mukhopadhyay A, Maulik U, Bandyopadhyay S, et al. Survey of multiobjective evolutionary algorithms for data mining: Part II. IEEE Trans Evol Computat, 2014, 18: 20–35
11 Shi Y H, Eberhart R C. A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation Conference, 4–9 May, 1998, Anchorage, AK, USA. 69–73
12 Martínez-Morales J, Quej-Cosgaya H, Lagunas-Jiménez J, et al. Design optimization of multilayer perceptron neural network by ant colony optimization applied to engine emissions data. Sci China Tech Sci, 2019, 62: 1055–1064
13 Handl J, Knowles J. An evolutionary approach to multiobjective clustering. IEEE Trans Evol Computat, 2007, 11: 56–76
14 Wahid A, Gao X Y, Andreae P. Multi-objective clustering ensemble for high-dimensional data based on strength Pareto evolutionary algorithm (SPEA-II). In: 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 19–21 October, 2015, Paris, France
15 Kirkland O, Rayward-Smith V J, Iglesia B D L. A novel multi-objective genetic algorithm for clustering. In: International Conference on Intelligent Data Engineering & Automated Learning, 7–9, September, 2011. 317–326
16 Özyer T, Zhang M, Alhajj R. Integrating multi-objective genetic algorithm based clustering and data partitioning for skyline computation. Appl Intell, 2011, 35: 110–122
17 Mukhopadhyay A, Maulik U, Bandyopadhyay S. A survey of multi-objective evolutionary clustering. ACM Comput Surv, 2015, 47: 1–46
18 İnkaya T, Kayalıgil S, Özdemirel N E. Ant Colony Optimization based clustering methodology. Appl Soft Computing, 2015, 28: 301–311
19 Alam S, Dobbie G, Koh Y S, et al. Research on particle swarm optimization based clustering: A systematic review of literature and techniques. Swarm Evolary Computat, 2014, 17: 1–13
20 Duan H B, Qiao P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. Int Jnl Intel Comp Cyber, 2014, 7: 24–37
21 Duan H B, Qiu H X. Advancements in pigeon-inspired optimization and its variants. Sci China Inf Sci, 2019, 62: 070201
22 Duan H B, Wang X H. Echo state networks with orthogonal pigeon-inspired optimization for image restoration. IEEE Trans Neural Netw Learn Syst, 2017, 27: 2413–2425
23 Xian N, Chen Z L. A quantum-behaved pigeon-inspired optimization approach to explicit nonlinear model predictive controller for quadrotor. Int Jnl Intel Comp Cyber, 2018, 11: 47–63
24 Hai X S, Wang Z L, Feng Q, et al. Mobile robot ADRC with an automatic parameter tuning mechanism via modified pigeon-inspired optimization. IEEE/ASME Trans Mechatron, 2020, 24: 2616–2626
25 Zhang B, Duan H B. Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment. IEEE/ACM Trans Comput Biol Bioinf, 2017, 14: 97–107
26 Deng Y M, Zhu W R, Duan H B. Hybrid membrane computing and pigeon-inspired optimization algorithm for brushless direct current motor parameter design. Sci China Tech Sci, 2015, 59: 1435–1441
27 Qiu H X, Duan H B. A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles. Inf Sci, 2020, 509: 515–529
28 Duan H B, Huo M Z, Yang Z Y, et al. Predator-prey pigeon-inspired optimization for UAV ALS longitudinal parameters tuning. IEEE Trans Aerosp Electron Syst, 2019, 55: 2347–2358
29 Pei J Z, Su Y X, Zhang D H. Fuzzy energy management strategy for parallel HEV based on pigeon-inspired optimization algorithm. Sci China Tech Sci, 2017, 60: 425–433
30 Bolaji A L, Balogun, S Babatunde, et al. Adaptation of binary pigeon-inspired algorithm for solving multi-dimensional knapsack problem. In: Pant M, Ray K, Sharma T, et al., eds. Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing, vol 583. Singapore: Springer, 2018. 743–751
31 Zhong Y W, Wang L J, Lin M, et al. Discrete pigeon-inspired optimization algorithm with Metropolis acceptance criterion for large-scale traveling salesman problem. Swarm Evol Computat, 2019, 48: 134–144
32 Garza-Fabre M, Handl J, Knowles J. An improved and more scalable evolutionary approach to multiobjective clustering. IEEE Trans Evol Computat, 2018, 22: 515–535
33 Rachmawati L, Srinivasan D. Multiobjective evolutionary algorithm with controllable focus on the knees of the Pareto front. IEEE Trans Evol Computat, 2009, 13: 810–824
34 Duan H B, Qiu H X, Fan Y M. Unmanned aerial vehicle close formation cooperative control based on predatory escaping pigeon-inspired optimization (in Chinese). Sci Sin Tech, 2015, 45: 559–572
35 Agustín-Blas L E, Salcedo-Sanz S, Jiménez-Fernández S, et al. A new grouping genetic algorithm for clustering problems. Expert Syst Appl, 2012, 39: 9695–9703
36 Rothlauf F, Goldberg D E. Redundant representations in evolutionary computation. Evol Comput, 2003, 11: 381–415
37 Cheriton D, Tarjan R E. Finding minimum spanning trees. SIAM J Comput, 1976, 5: 724–742
38 Voorneveld M. Characterization of Pareto dominance. Operations Res Lett, 2003, 31: 7–11
39 Chen B L, Zeng W H, Lin Y B, et al. A new local search-based multiobjective optimization algorithm. IEEE Trans Evol Computat, 2015, 19: 50–73
40 Li X D. Niching without niching parameters: Particle swarm optimization using a ring topology. IEEE Trans Evol Computat, 2010, 14: 150–169
41 Yue C T, Qu B Y, Liang J. A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. IEEE Trans Evol Computat, 2018, 22: 805–817
42 Garcia-Piquer A, Fornells A, Bacardit J, et al. Large-scale experimental evaluation of cluster representations for multiobjective evolutionary clustering. IEEE Trans Evol Computat, 2014, 18: 36–53
43 Likas A, Vlassis N, J. Verbeek J. The global *k*-means clustering algorithm. Pattern Recognit, 2003, 36: 451–461
44 Park H S, Jun C H. A simple and fast algorithm for K-medoids clustering. Expert Syst Appl, 2009, 36: 3336–3341
45 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Computat, 2002, 6: 182–197