Harmonious Genetic Clustering

Faliang Huang, Xuelong Li, *Fellow, IEEE*, Shichao Zhang, *Senior Member, IEEE*, and Jilian Zhang

***Abstract*—To automatically determine the number of clusters and generate more quality clusters while clustering data samples, we propose a harmonious genetic clustering algorithm, named HGCA, which is based on harmonious mating in eugenic theory. Different from extant genetic clustering methods that only use fitness, HGCA aims to select the most suitable mate for each chromosome and takes into account chromosomes gender, age, and fitness when computing mating attractiveness. To avoid illegal mating, we design three mating prohibition schemes, i.e., no mating prohibition, mating prohibition based on lineal relativeness, and mating prohibition based on collateral relativeness, and three mating strategies, i.e., greedy eugenics-based mating strategy, eugenics-based mating strategy based on weighted bipartite matching, and eugenics-based mating strategy based on unweighted bipartite matching, for harmonious mating. In particular, a novel single-point crossover operator called variable-length-and-gender-balance crossover is devised to probabilistically guarantee the balance between population gender ratio and dynamics of chromosome lengths. We evaluate the proposed approach on real-life and artificial datasets, and the results show that our algorithm outperforms existing genetic clustering methods in terms of robustness, efficiency, and effectiveness.**

***Index Terms*—Data clustering, eugenic theory, genetic clustering, mating operator.**

I. Introduction

DATA clustering is one of the classic mainstream research topics in data mining and machine learning, which partitions a set of objects into different clusters, such that intracluster objects are more similar to each other than intercluster objects. From the optimization point of view, clustering N objects into K clusters can be considered as a particular kind of NP-hard problem. In the past few years, tremendous research effort has been put in population-based-optimization to address the NP-hard problem [1]–[4] for its excellent search ability [5]–[8]. However, most of the existing clustering techniques accept the number of clusters K as an input parameter, instead of determining it on-the-fly. In real world applications, it is often impossible for users to set an appropriate value of K in advance, because most users have zero knowledge about their datasets that are to be mined. In general, an improper K can easily mislead the clustering process and result in poor clustering outcome. For example, while clustering a set of documents returned from a search engine, the number of clusters K varies across different queries. Also, if the data set is represented in high-dimensional feature vectors, it may be impossible in practice to visualize the data for tracking its number of clusters.

To tackle the issue, we propose a harmonious genetic clustering algorithm, called HGCA, based on eugenic theory [9]. In HGCA, a clustering solution is represented by a real-coded chromosome with variable length. To improve clustering performance, we propose a harmonious mating operator to balance population exploration and exploitation, in order to guide the population to a better search path by self-adaptive control of population diversity. Specifically, we first compute mating attractiveness of candidate chromosomes based on some of their features, including gender, age, and fitness, instead of only considering fitness. And then, we design a novel mating operator, which integrates three mating prohibition schemes, i.e., no mating prohibition (NMP), mating prohibition based on lineal relativeness (MPLR) and mating prohibition based on collateral relativeness (MPCR), as well as three mating strategies, namely, greedy eugenics-based mating strategy (Greedy-EMS), eugenics-based mating strategy based on weighted bipartite matching (WBM-EMS), and eugenics-based mating strategy based on unweighted bipartite matching (UBM-EMS), in order to select the most suitable mate for each chromosome. To improve traditional single-point crossover operator, we design a novel crossover operator, called variable-length-and-gender-balance crossover (VGC), which can probabilistically guarantee balance between ratio of population gender and dynamics of chromosome lengths. In HGCA, the number of clusters, K, is determined adaptively by the crossover operator VGC. Specifically, K is implicitly represented as the length of chromosome, variability of which can be probabilistically guaranteed, that is, K is a hidden parameter to be optimized in the search process of HGCA.

Manuscript received November 5, 2015; revised May 1, 2016; accepted November 1, 2016. Date of publication January 5, 2017; date of current version November 15, 2017. The work of S. Zhang was supported in part by the China 973 Program under Grant 2016YFB1000905, in part by the China 1000-Plan National Distinguished Professorship, in part by the China Key Research Program under Grant 2013CB329404, in part by the Natural Science Foundation of China under Grant 61672177, in part by the Guangxi Bagui Teams for Innovation and Research, in part by the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing. The work of J. Zhang was supported by the NSFC under Grant 61363009. This paper was recommended by Associate Editor H. Yin. *(Corresponding author: Jilian Zhang.)*

F. Huang is with the Fujian Engineering Research Center of Public Service Big Data Mining and Application, Faculty of Software, Fujian Normal University, Fuzhou 350007, China (e-mail: faliang.huang@gmail.com).

X. Li is with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

S. Zhang is with the Guangxi Key Laboratory of MIMS and College of Computer Science and Information Technology, Guangxi Normal University, Guilin 541004, China (e-mail: zhangsc@gxnu.edu.cn).

J. Zhang is with the Guangxi Key Laboratory of Cross-Border E-Commerce Intelligent Information Processing, Guangxi University of Finance and Economics, Nanning 530003, China (e-mail: zhangjilian@yeah.net.).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2016.2628722

2168-2267 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

The contributions of this paper mainly include the following.

1) We devise a novel crossover operator VGC for automatic determination of the optimal number of clusters for any unlabeled dataset.
2) We propose a harmonious mating operator to improve search performance of genetic algorithm (GA) optimization-based clustering algorithm.

The rest of this paper is organized as follows. In Section II, we review related work. Our HGCA approach is presented in Section III. Section IV reports the experimental results. Finally, we conclude this paper in Section V.

II. Related Work

A. Metaheuristic Techniques to Determine K

As a challenging problem, determining the number of clusters has been attracting much attention from population-based optimization research community. We first review some strategies related to GAs for automatically determining K. As early as 2002, Bandyopadhyay and Maulik [10] attempted to use GA to automatically determine the number of clusters K. Following the line to utilize excellent search ability of GAs, some researchers [11], [12] made some efforts to determine the number of clusters K automatically by using chromosomes with variable lengths. And the idea of divide-and-conquer was also integrated into detecting the number of clusters, for instance, Sheng *et al.* [13] proposed a hybrid niching GA to automatically evolve the proper number of clusters and the appropriate partitioning of dataset by using weighted sum of several normalized cluster validity indexes. Shin *et al.* [14] proposed a two-leveled symbiotic evolutionary clustering algorithm to divide a clustering problem with unknown K into two subproblems: 1) finding the number of clusters and 2) grouping the data into clusters accordingly. He and Tan [15] proposed a dynamic genetic clustering algorithm (TGCA) with two-staged selection and mutation to find the optimal values for both the number of clusters and the cluster centers. Different from pure metaheuristic techniques, Liu *et al.* [16] proposed an approach to combine multiobjective GA with K-means to determine K. Moreover, another noteworthy technique to produce an appropriate K is a data-driven genetic clustering algorithm, automatic genetic clustering for unknown K [17], which applied noising selection and division-absorption mutation to automatically determine the optimal number of clusters.

In addition to GAs-based techniques, other population-based optimization techniques, such as particle swarm optimization (PSO), differential evolution (DE), etc., are also frequently used to automatically determine optimal K. Omran *et al.* [18] proposed a dynamic clustering approach DCPSO, which uses binary PSO to select the best number of clusters with minimal user interference. Masoud *et al.* [19] proposed a dynamic clustering algorithm improved version of combinatorial particle swarm optimization to automatically find the best number of clusters and categorizes data objects simultaneously. Das *et al.* [20] presented an automatic clustering algorithm ACDE to apply DE techniques for automatic determination of the optimal number of clusters for any unlabeled data set.

Similar to the work in [10]–[17], the proposed HGCA is also a genetic clustering algorithm (GCA), which attempts to exploit search capability of GA for automatically evolving the number of clusters as well as proper clustering of any data set. However, to the best of our knowledge, existing GCAs for automatic determination of K, including the above algorithms, have no mating operator to improve the quality of clustering result.

B. Mating Techniques

Mating is a genetic operator to construct the most suitable parent chromosome pairs to perform crossover, which involves evaluation of mating attractiveness and creation of mating prohibition rules. The mating mechanism in natural evolution is complex [21]. For example, the factors affecting human marriage usually include wealth, health, and physical appearance. This is why most researchers are reluctant to take any mating operator into account while devising GAs. And few studies on mating strategy are classified into exploitation priority and exploration priority.

Exploitation priority mating strategy is proposed to avoid generating low population diversity that is caused by too low population exploitation rate. It can make population incapable of finding some potentially valuable solutions hidden in fitness landscape, and the search algorithm may converge to locally optimal solution. Eshelman and Schaffer [22] proposed to prohibit incest to refrain from premature caused by excessive assimilation, and encouraged chromosomes with short hamming distance to mate. Matsui [23] presented a correlative tournament selection operator to choose candidate chromosomes with the highest correlation as parents. Craighurst and Martin [24] put forward the family tree to disallow incest by the ancestry-based incest law.

Compare to the exploitation priority mating strategy, exploration priority mating strategy attempts to make full use of the information hidden in the current population in order to speed up convergence. De *et al.* [25] proposed to confine mating of two chromosomes with hamming distance greater than a minimum. Fernandes *et al.* [26] applied different assortative mating strategies to address vector quantization. Ochoa *et al.* [27] draw a conclusion that GA efficiency is closely related to mating strategies.

Its worth pointing out that researchers have presented some approaches by making compromise between the above two strategies. Ting *et al.* [28] proposed a tabu GA to prevent inbreeding. Fernandes and Rosa [29] presented a variable dissortative mating GA to automatically mate candidate parent chromosomes based on the number of newborn chromosomes and population diversity.

The studies above show that optimization performance of GAs can be enhanced to some extent via addition of mating operator, and this motivates us to introduce mating operator into the search process for optimal cluster structure hidden in dataset.

However, most existing mating operators only choose fitness and distance of chromosomes to evaluate mating attractiveness

TABLE I
NOTATIONS USED IN THE HGCA ALGORITHM

Symbol	Description	Symbol	Description
D	Dataset	F	Female population
o	Object in D	M	Male population
h	#features of an object	x	Chromosome
C	Set of clusters	f	Female chromosome
C_i	The i^{th} cluster in C	m	Male chromosome
O_i	Center of cluster C_i	Q	Mating prohibition record
P	Population	p	Mating prohibition scheme

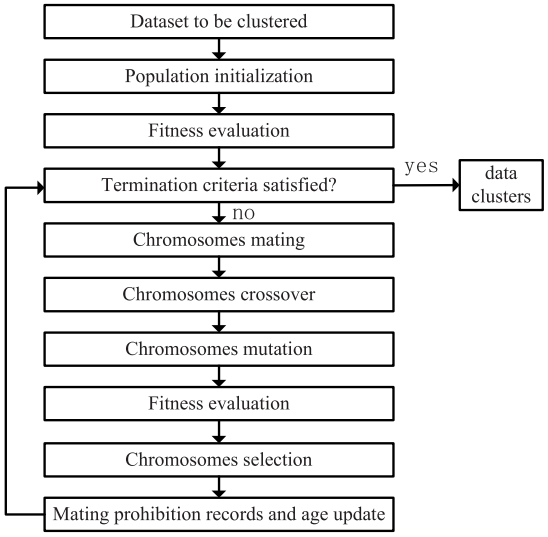

Fig. 1. Flowchart of HGCA.

of candidate chromosome pairs, paying no attention to other important attributes such as gender and age. These result in poor improvement on convergence rate for GAs. To the best of our knowledge, we are the first to utilize GA with mating operator for automatic data clustering.

III. PROPOSED ALGORITHM

To facilitate presentation, we summarize the symbols used in Table I. We sketch our method HGCA in the flowchart in Fig. 1. For a dataset to be clustered, we first randomly initialize a population using real-encoding scheme and assign gender and age to each chromosome, where each chromosome corresponds to a clustering solution for the dataset. Then we search for an optimal clustering of the dataset in a loop, performing operators one by one, i.e., mating (creating potential suitable mate pairs using mating strategies), crossover (generating new population with gender balance and variable chromosome length), mutation, evaluation, and selection. Finally, we get the optimal clustering solution by decoding the chromosome with the best fitness.

A. Chromosome Encoding

In GCAs, popular methods to represent a partition of a dataset include binary-encoding, integer-encoding, and real-encoding [1]. Here, we adopt a variable-length and real-encoding representation scheme, as shown in Fig. 2.

Given a dataset $D = (o_1, o_2, \ldots, o_{|D|})$, where each object o in D can be regarded as a point in Euclidean space R^h. In particular, if the input dataset D is a document corpus or an

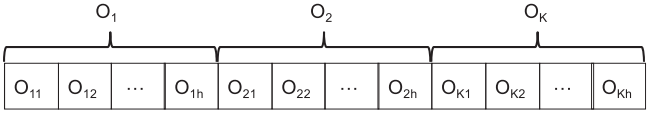

Fig. 2. Real-encoding chromosome.

image dataset, some preprocessing techniques such as representation and feature selection may be applied to vectorize the data objects, e.g., documents in corpus or handwritten letters in image dataset. Assume D is divided into a set C of K clusters, $C = (C_1, C_2, \ldots, C_K)$, the corresponding real-encoding chromosome can be vectorized as $x = (O_1, O_2, \ldots, O_K)$, where $O_i = (O_{i1}, O_{i2}, \ldots, O_{ih})$ denotes the center of cluster C_i.

B. Population Initialization

Studies show that the appropriate initial values of cluster centers greatly affect the quality of partition clustering [30]. There are many methods for population initialization, among which random sampling is the most widely used one.

Observation 1: In our preliminary experiment results, we discover that random sampling may produce some chromosomes, in which there may exist some cluster centers that do not dominate any objects. We call such cluster center *invalid cluster center (ICC)*.

Definition 1 (ICC): Let $O = (O_1, O_2, \ldots, O_K)$ be the set of cluster centers of chromosome x. For cluster center $O_i \in O$ and any object $o \in D$, there always exists another cluster center $O_j \in O$ such that $\text{dist}(o, O_i) > \text{dist}(o, O_j)$ holds. The cluster center O_i is referred to as ICC.

To avoid producing ICCs during population initialization, we adopt a maximum attribute range partition method for population initialization [15].

C. Chromosome Attributes

Darwinian evolution indicates that individuals exhibit different attributes during natural evolution, and attribute difference in the individuals plays an important role in their survival and reproduction. Hence, we endow the chromosomes with fitness, age, and gender in order to depict their mating behaviors more reasonably.

1) Chromosome Fitness: In general, chromosome fitness is used to measure chromosome viability and is evaluated by the objective function in a clustering problem. The objective of the clustering problem considered in this paper is to maximize similarity within each cluster and dissimilarity between clusters. Many measurements to evaluate clustering results have been proposed [31]. In this paper, three different criteria, i.e., Davies—Bouldin index (DBI) [32], cluster similarity (CS) [33], and variance ratio criterion (VRC) [34], are separately used to compute the fitness of a chromosome. We experimentally analyze the impact of different fitness measurements on clustering quality in Section IV-B1. We review the definitions of DBI, CS, and VRC below.

DBI is used to find clusters that are compact and well separated by minimizing the intracluster distance while maximizing the intercluster distance. The DBI index of a chromosome

x with K clusters is defined as follows:

$$\text{fitness}(x) = \frac{1}{K}\sum_{i=1}^{K} \max_{j=1,\ldots,K, j\neq i} \frac{S_i + S_j}{\text{dist}(O_i, O_j)} \tag{1}$$

where $S_i = (1/|C_i|)\sum_{o\in C_i} \text{dist}(o, O_i)$ is the average intracluster distance, o is an object in cluster C_i, $|C_i|$ is the cardinality of C_i, $\text{dist}(\cdot,\cdot)$ is the Euclidean distance between two vectors.

CS measure is a function of the ratio of the sum of within-cluster scatter to between-cluster separation, which is formalized as below

$$\text{CS} = \frac{\sum_{i=1}^{K}\left[\frac{1}{N_i}\sum_{o_i\in C_i}\max_{o_q\in C_i}\{\text{dist}(o_i, o_q)\}\right]}{\sum_{i=1}^{K}\min_{j\in K, j\neq i}\{\text{dist}(O_i, O_q)\}}. \tag{2}$$

VRC is another criterion used for cluster validation. It considers both the intracluster and intercluster distances. Formally, VRC is defined as

$$\text{VRC} = \frac{\text{BCSS}}{K-1} \times \frac{n-K}{\text{WCSS}} \tag{3}$$

where BCSS is the between cluster sum-of-squares, WCSS is the within cluster sum-of-squares, and n is the number of data samples.

2) Chromosome Gender: Chromosome gender is an important factor affecting mating behavior of chromosomes. Dividing a population according to gender has been investigated extensively and it can bring several advantages for GAs [35], [36]. For example, Vrajitoru [36] showed that "sexual reproduction with completely separating male and female organisms has proven to provide several advantages in nature." To obtain high quality chromosomes, we propose to attach a unique gender to each chromosome based on the concept of gonochorism [37]. Gonochorism describes the situation in which individuals of a species are of one of the two distinct sexes and retain that sex throughout their lives. Gonochorous individuals always produce offspring with sexual reproduction.

We initialize the gender of chromosomes as follows. Let P be an initial population. We randomly select $\lfloor(|P|/2)\rfloor$ chromosomes from P to construct female chromosome set F, and the rest of the population form the male chromosome set M. It is clear that $P = F \cup M$ and $F \cap M = \emptyset$. Random selection operator is chosen to construct female chromosomes on account of the following consideration. In GAs, selection pressure controls the selection of individuals from one population to the next generation, and random selection operator with weak selection pressure can maintain high population diversity and allow for exploration in the initial stage of population evolution [50].

In GAs, genetic operators such as crossover and mutation can reproduce newborn chromosomes. For the newborn chromosomes, we propose to determine the gender of the offspring like this: the gender of the offspring resulting from crossover is determined by the proposed crossover operator VGC (details given in Section III-E), and the gender of the offspring resulting from mutation remains the same as that of their corresponding parent.

3) Chromosome Age: It is widely known in eugenics that for all species that practice reproduction, there is a best fertile period (BFP) in which mating will be more likely to reproduce high quality offsprings. And studies indicate that utilization of chromosome age information is helpful to boost search ability [38], [39]. Hence, we propose a new approach to measure the influence the chromosome age exerts on mating.

Definition 2 (BFP): Let S be an ordered series of ages of chromosomes, the BFP can be quantified as an age interval $[lb, ub]$, where lb and ub are manually set to be the 15th and 85th percentile of the series S, respectively.

Based on the definition, a new piecewise function fec(.) is introduced to quantify a chromosome's ability to reproduce high quality offspring during its BFP, that is

$$\text{fec}(x) = \begin{cases} \frac{x.\text{age}-lb}{\text{mid}-lb}, & lb \le x.\text{age} < \text{mid} \\ 1, & x.\text{age} = \text{mid} \\ \frac{ub-x.\text{age}}{ub-\text{mid}}, & \text{mid} < x.\text{age} \le ub \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where mid is the age when chromosomes are most likely to reproduce optimal offspring, and x.age is the age of chromosome x, which can be computed according to the following two rules.

Rule 1: If chromosome x is a newborn, then $x.\text{age} = 0$.

Rule 2: If chromosome x is intact from the last generation, then $x.\text{age} = x.\text{age} + 1$.

D. Mating Operator

Most existing mating operators [23]–[25] only choose the fitness and distance of chromosomes to evaluate the mating attractiveness of candidate chromosome pairs, paying no attention to other important attributes such as gender and age. This results in poor improvement on the convergence rate of GAs. On the other hand, studies in eugenics [9] indicate that proper parent chromosomes can improve the quality of their offsprings. Therefore, we devise a novel mating operator. In particular, we first introduce chromosome attributes, i.e., gender, fitness, and age, to evaluate chromosome mating attractiveness. Then, we represent chromosome mating repulsiveness with mating prohibition rules. Finally, we present three mating strategies based on mating attractiveness and mating repulsiveness.

Compare to the state-of-the-art mating operators, our proposed mating operator has following advantages: 1) it can provide suitable mating pairs that may produce better offsprings, i.e., it may improve performance of GCAs and 2) the proposed mating operator is more flexible, since it provides users with three alternative mating strategies, among which the users can pick up a proper one according to their clustering tasks. However, just as every coin has two sides, the diversity of mating strategy in the proposed mating operator brings forward another new problem—how to select a proper mating strategy for a clustering task. We discuss this in detail in Sections IV-B2 and IV-B3.

1) Mating Attractiveness: Eugenic studies show that it is a process of choosing and attracting each other that dioecious chromosomes look for their suitable mates. Motivated by this, we propose to integrate age, gender, and fitness into the process of scoring chromosome mating attractiveness.

First, we consider the impact of age on mating attractiveness. We assume that for two heterosexual chromosomes in their BFP, it is more likely for them to reproduce higher quality offspring if the age difference between them is small. Therefore, the impact on mating attractiveness between a heterosexual chromosome pair (m, f) can be formalized as follows:

$$\text{AgeAttract}(m, f) = \frac{\text{fec}(m) * \text{fec}(f)}{|m.\text{age} - f.\text{age}|}. \tag{5}$$

Second, chromosome fitness is taken into account while computing mating attractiveness, because a chromosome tends to choose other chromosomes with high fitness as its potential mates. So, the contribution of fitness to mating attractiveness between the heterosexual chromosome pair (m, f) can be formulated as a function proportional to the fitness of m and f. Here, a simple average function is adopted for this purpose

$$\text{FitAttract}(m, f) = \frac{1}{2}(\text{fitness}(m) + \text{fitness}(f)). \tag{6}$$

Finally, the above two mating attractiveness measures can be integrated into a combined index, shown below.

$$\begin{aligned} &\text{Attract}(m, f) = w_1 \cdot \text{FitAttract}(m, f) + w_2 \cdot \text{AgeAttract}(m, f) \\ &\text{s.t.} \begin{cases} w_1 + w_2 = 1 \\ w_1, w_2 \in [0, 1] \end{cases} \end{aligned} \tag{7}$$

where w_1 and w_2 are weights on fitness attractiveness and age attractiveness, respectively. Strategies to set the above two weights play an important role in convergence of clustering algorithms. A large w_1 value can lead to premature result, whereas a small w_1 value may result in slow convergence rate. Obviously, it is very difficult for users to determine appropriate value of w_1. In this paper, we apply an adaptive strategy to determine the value of w_1. Specifically, if the fitness of the best chromosome in population has not been improved in the successive T_{max} generations, then we set $w_1 = \tau w_1$, where $\tau \in [0, 1]$. In addition, our preliminary results reveal that influence of age on mating attractiveness is very small during early period of the population evolution. Thus, we ignore age influence in this period. In the implementation, we do not take age influence into account in the first T_w generations.

2) Mating Prohibition: According to Eugenic theory, frequent inbreeding can decrease genetic diversity and increase gene expression of bad recessive. To refrain from frequent inbreeding, we propose two prohibition schemes for chromosome mating, i.e., MPLR and MPCR. In MPLR, chromosomes cannot mate with their parents, and in MPCR, chromosomes cannot mate with their parents and their grandparents. Obviously, prohibition of MPCR is stricter than MPLR. To implement the above two prohibition schemes, we allocate a queue Q for each chromosome x to inherit and update its mating prohibition records. Specifically, in MPLR, $x.Q$ is used to store the parents of x, and in MPCR, $x.Q$ is used to store the parents and grandparents of x. To save space, we set $|x.Q| = 2$ in MPLR and $|x.Q| = 6$ in MPCR. So, MPLR and MPCR can be expressed by following rule: for male chromosome m and female chromosome f, if $m \in f.Q$ and $f \in m.Q$, then m cannot mate with f.

If we further take into account NMP, then we have a mating prohibition scheme set MPS = {NMP, MPLR, MPCR}. To check whether mating between m and f violates a given mating prohibition scheme p, we define a function mps as

$$\text{mps}(m, f, p) = \begin{cases} 1, & \text{if } m \text{ can mate with } f \text{ w.r.t scheme } p \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $p \in$ MPS denotes a mating prohibition scheme.

Given the dynamics of chromosome mating prohibition records, updating the records is realized by enqueue and dequeue operator on a queue Q. Specifically, if crossover between chromosome m and f reproduces their offspring x and y, then updating record x requires two operations *enqueue*$(x.Q, m)$ and *enqueue*$(x.Q, f)$ in MPLR, and four operations *dequeue*$(x.Q)$, *dequeue*$(x.Q)$, *enqueue*$(x.Q, m)$, and *enqueue*$(x.Q, f)$ in MPCR. Clearly, updating record y needs to perform similar operations.

3) Mating Strategies: Based on mating attractiveness and mating prohibition, it is a challenging problem to construct a set of suitable mate pairs that can offer current parent population a better chance of reproducing higher quality offspring. The problem can be formalized as a chromosome mating optimization problem, as follows:

$$\max \sum_{f_i \in F, m_j \in M, \text{mps}(f_i, m_j, p) = 1} \text{Attract}(f_i, m_j) \tag{9}$$

where f_i and m_j denote a female chromosome in set F and a male chromosome in set M, respectively.

Obviously, solution to the above optimization problem can be formalized as a pair set FMate, which is a subset of Cartesian product $M \times F$. To solve this problem, we propose three mating strategies, i.e., Greedy-EMS, WBM-EMS, and UBM-EMS.

a) Greedy-EMS: Greedy-EMS is a greedy strategy to select a mate with maximal mating attractiveness for a chromosome under a certain mating prohibition scheme. Greedy-EMS strategy can be formulated as

$$\text{mate}(f_i) = \underset{m_j \in \{m_j | \text{mps}(f_i, m_j, p) = 1\}}{\arg\max} \text{Attract}(f_i, m_j). \tag{10}$$

Obviously, it is not difficult to construct FMate using this strategy. Greedy-EMS strategy is simple and easy to implement, but it has some shortcomings, such as gene drift and diversity loss.

Example 1: Consider a male chromosome set $M = \{a, b, c, d, e\}$ and a female chromosome set $F = \{1, 2, 3, 4, 5\}$. Mating prohibition scheme NMP can be used between M and F. According to chromosome mating attractiveness in Fig. 3(a) and Formula (8), we have a mating pair set FMate = $\{(a, 2), (b, 2), (c, 2), (d, 2), (e, 2)\}$. By using population gene diversity formula GD = (# locuses with different value/# total locuses), we have GD = 45/50 before crossover, and GD = 24/50 after crossover. Comparing these two GDs, we can see that Greedy-EMS strategy leads to population diversity loss.

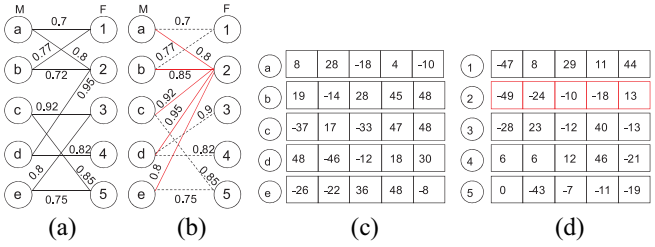

Fig. 3. Diversity loss of Greedy-EMS. (a) Mating attractiveness between female and male chromosomes, where an edge exists if the mating prohibition conditions are satisfied, and edge weight denotes mating attractiveness between chromosomes. (b) Mating set constructed by Greedy-EMS, where red edge denotes the legality between chromosomes. (c) Male chromosome set. (d) Female chromosome set.

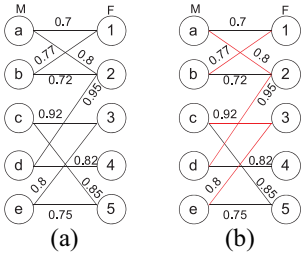

Fig. 4. Population diversity of WBM-EMS. (a) Mating attractiveness between female and male chromosome, where an edge exists if the mating prohibition conditions are satisfied, and edge weight denotes mating attractiveness between chromosomes. (b) Mating set constructed by WBM-EMS, where red edges denote the legality between chromosomes.

b) WBM-EMS: To overcome the shortcomings of Greedy-EMS strategy, we propose to model the chromosome mating optimization problem as a weighted bipartite graph matching problem. The model is described as follows.

Suppose $G = \langle V, E, W\rangle$ is a weighted bipartite graph with vertex set $V = F' \cup M'$, where

$$F' = \bigcup_{i=1}^{|M|} \text{mate}(m_i)$$
$$M' = M - \{m_i | \text{mate}(m_i) = \emptyset, m_i \in M\}$$
$$\text{mate}(m_i) = \left\{f_j \middle| \text{mps}\left(f_j, m_i, p\right) = 1 \wedge f_j \in F\right\}$$

and $E = \{e_{ij}(f_i \leftrightarrow m_j) | f_i \in F' \wedge m_j \in \text{mate}(f_i)\}$ is the edge set with weight set $W = \{w_{ij} | w_{ij} = \text{Attract}(f_i, m_j)\}$.

It is not difficult to prove that the chromosome mating optimization problem is equivalent to the maximum matching problem in the weighted bipartite graph G, that is

$$PMat = \arg\max_{PM_k \in PM} \sum \left(w_{ij} | w_{ij} \in PM_k\right)$$

where PM denotes matching collection of G, PM_k denotes the kth matching in PM. According to graph theory, we have $|PM| = \max(|F|!, |M|!)$. The maximum matching problem in weighted bipartite is widely studied and many algorithms have been proposed. Here, we adopt Hungarian algorithm [40] to acquire the optimal mating pair set FMate. Based on the time complexity of Hungarian algorithm, we can deduce that time complexity of WBM-EMS is $O(|PM| * |E|^2)$.

Example 2: Suppose chromosome attractiveness between $M = \{a, b, c, d, e\}$ and $F = \{1, 2, 3, 4, 5\}$ is shown in Fig. 3(a). After applying Hungarian algorithm to Fig. 4(a), we have the optimal mating pair set $\{(a, 2), (b, 1), (c, 3), (d, 2), (e, 3)\}$, represented by the red edges in Fig. 4(b). Similar to Example 1, we compute gene diversity of the newborn population, which is GD $= 32/50$. Comparing WBM-EMS (Fig. 4) with Greedy-EMS (Fig. 3), we find that WBM-EMS can successfully maintain population diversity level by avoiding too many male chromosomes mating with several high quality female chromosomes.

4) UBM-EMS: The time complexity of WBM-EMS is high, which severely slows down convergence rate of the optimization algorithm, although it can effectively circumvent undue loss of population diversity. To improve computational efficiency, we further propose UBM-EMS strategy to transform the weighted bipartite model to an unweighted bipartite model.

Mating relation of heterosexual chromosomes can be moderately simplified by virtue of a threshold function, defined in Formula (11). That is, the threshold function is used to filter out the edges with weight smaller than threshold λ in the weighted bipartite graph constructed by WBM-EMS, and equally assign 1 to the weight of left edges. Then we have a reduced weighted bipartite G. Comparing Fig. 4(a) with (b), we can find that the number of deleted edges is 3

$$w_{ij} = \begin{cases} 1, & \text{if Attract}\left(f_i, m_j\right) > \lambda \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

If we optimize FMate in WBM-EMS by applying Hungarian algorithm to the reduced G, we may obtain performance gain to some extent. However, Hungarian-based algorithms essentially cannot decrease the time complexity, even though the reduced G has fewer edges. Since the edge weight in the reduced G equals to one, by adding in a source node s and a sink node t we can further transform the reduced G into a flow net graph, as follows:

$$G' = \left(V', E', W'\right), \text{ where } V' = V \cup \{s, t\}$$
$$E' = E \cup \left\{\left(s, v'_i\right) : v'_i \in F\right\} \cup \left\{\left(v'_j, t\right) : v'_j \in M\right\}$$
$$\forall w_{ij} \in W', w_{ij} = 1.$$

Now, the optimization problem has been transformed from a maximum matching problem on weighted bipartite to a max-flow problem on flow net graph. And its time complexity can be further reduced to $O(|PM| * |E| * \log|PM|)$ by using Goldberg algorithm [41]. However, given that all the edge weights are equal, it is possible to generate multiple flow nets with equal maximal flow. That is, this property poses another issue, i.e., how to select the best one from the final flow nets as a solution to the chromosome mating optimization problem. To address the issue, we propose a simple greedy approach below.

Let the final flow nets be NET $= \{\text{net}_1, \text{net}_2, \ldots, \text{net}_k\}$, $\text{net}_i = (V'_i, E'_i)$, $V'_i \subseteq V'$, $E'_i \subseteq E'$, the final optimal flow net is $\text{net}^* = \arg\max_{net_i \in \text{NET}} |V'_i|$.

Example 3: Suppose chromosome set M and F with mating attractiveness are depicted in Fig. 5(a). An unweighted bipartite graph [Fig. 5(b)] is obtained by removing the edges with

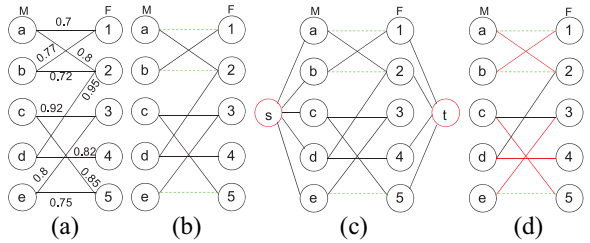

Fig. 5. Diversity loss of Greedy-EMS. (a) Mating attractiveness between female and male chromosomes, where an edge exists if the mating prohibition conditions are satisfied, and weights denote mating attractiveness between chromosomes. (b) Mating set constructed by Greedy-EMS, red edges denote the legality between chromosomes. (c) Male chromosome set. (d) Female chromosome set.

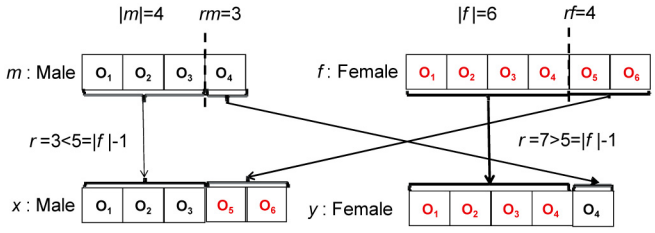

Fig. 6. Crossover process of VGC operator.

weight smaller than threshold $\lambda = 0.75$. By further adding in a source node s and a sink node t, we have a flow net (Fig. 5(c)) and an optimal mating pair set represented by red edge in Fig. 5(d), which is obtained by applying Goldberg algorithm to the flow net. Compared to Greedy-EMS and WBM-EMS, UBM-EMS completely eliminates the loss of population gene diversity, because GD index remains GD $= 41/50$ before performing crossover.

The aforementioned three mating strategies have different computational complexity and optimization effectiveness. In current HGCA, these strategies are selected by data analyst according to clustering tasks at hand, aiming to facilitate the selection. In Section IV-B5, we experimentally compare mating strategies in terms of efficiency and effectiveness. Indeed, research on how to adaptively choose the most appropriate strategy for clustering tasks is important and interesting, and it is one of our future work.

E. Crossover Operator

In general, after applying mating strategies in constructing chromosome pairs, GCAs proceed to perform population evolution for the emergence of high quality chromosomes which may contain better clustering outcomes. Crossover operator is one of the important subsequent promoters. Unfortunately, existing crossover operators have some limitations. First, most existing crossover operators are fixed-length, presuming that the number of clusters is known in advance. However, as discussed in introduction, the presumption often fails in real-life situations. Second, observation 1 shows that evolutionary population may reproduce chromosomes with ICCs. Removal of all the ICCs is helpful in improving algorithm performance, but this will result in chromosomes with unequal length, making the fixed-length crossover operators unsuitable for performing unequal length crossover. Third, although there are a few variable-length crossover operators, they all ignore gender attribute of chromosomes, resulting in unsatisfactory convergence rate [10]–[15].

To address the above limitations, we propose a VGC, described in Algorithm 1. Theorems 1 and 2 (see details in the Appendix) show that different from the existing crossover operators, VGC is a single point crossover operator that can guarantee both the variability of chromosome length and gender balance of offspring in probability. That is, the above

Algorithm 1 VGC

Input: parent chromosome pair (m, f)
Output: child chromosome x and y
Step 1: Generate random crossover point rm for m and rf for f
1.1: rm=$rand(1, |m| - 1)$
1.2: rf=$rand(1, |f| - 1)$
Step 2: Perform concatenation to produce x and y:
2.1 : x=$concat(m[1: rm], f[rf + 1: |f|])$
2.2 : y=$concat(f[1: rf], m[rm + 1: |m|])$
Step 3: Determine gender of child chromosome:
3.1 : for $\alpha \in \{x, y\}$ do
3.2 : if $|m| \geq |f|$ then{
3.3 : if $rm \geq rf$ then
3.4 : $r = rand(1, 2|m| - |f|)$
3.5 : if $r \leq |m| - 1$ then $\alpha.gender = male$
3.6 : else $\alpha.gender = female$
3.7 : else $\alpha.gender = male$
3.8 : }else {
3.9 : if $rm \leq rf$ then
3.10: $r = rand(1, 2|f| - |m|)$
3.11: if $r \leq |f| - 1$ then $\alpha.gender = male$
3.12: else $\alpha.gender = female$
3.13: else $\alpha.gender = male$ }

two theorems can provide theoretical basis for both avoiding population search ability degradation incurred by gender imbalance and maintaining dynamics of the number of clusters K. However, determining offspring gender in VGC may lead to higher computational cost.

We illustrate the details of VGC operator in Example 4. It is worth mentioning that the valid value of chromosome length is not an arbitrary integer, but a multiple of the length of cluster center vector.

Example 4: Suppose male parent chromosome m and female parent chromosome f partition a given set of objects into 4 clusters and 6 clusters, respectively, i.e., $|m| = 4$ and $|f| = 6$ (see Fig. 6). A random number generator generates random number $rm = 3$ as the crossover point for m and $rf = 4$ as the crossover point for f. Then child chromosome x and y are created by performing step 2 in Algorithm 1. Given $t|m| < |f|$ and $rm < rf$, step 3 in Algorithm 1 needs another two random numbers to determine the gender of x and y. Here, we have $r = 3$ for x, and $r = 7$ for y, so the gender of x is male and y is female.

F. Algorithm Description and Analysis

To facilitate understanding the above discussion, we describe HGCA in Algorithm 2. From the description, we can see that HGCA mainly consists of two major steps. Step 1 concerns initializing related parameters such as population size, selection probability, etc., and randomly partitioning the input dataset with population initialization, while step 2 is responsible for searching the optimal partition of the dataset by a loop that mainly includes temporary population generation, mating, crossover, mutation, evaluation, selection adaptive adjustment of weight, and obtaining the dataset partition by decoding the chromosome with best fitness.

Time complexity of HGCA can be computed as follows. We first compute CPU cost of step 1. Apparently, time complexity in parameter initialization in step 1.1 is $O(1)$, and let the maximum number of clusters resulting from generating random partitions be K, the time complexity of population initialization in step 1.2 is $O(N*K)$. Thus, the overall time complexity of step 1 is $O(N*K)$. And time complexity of step 2 is relatively complicated. With the definition of DBI, we can compute time complexity of step 2.1 as $O(K*|D|*N)$. CPU cost of temporary population generation in step 2.2 is simply $O(N)$. Although there are three candidate mating strategies, we choose candidate Greedy-EMS for simplicity, and time complexity of step 2.3 can be calculated as $O(|F'_{\text{gen}}|^2)$ according to Section III-D3. CPU cost of VGC operator in step 2.4 can be estimated as $O(|M^c_{\text{gen}}|)$. Similarly, time complexity of mutation operator in step 2.5, fitness evaluation in step 2.6, selection operator in step 2.7 and age update in step 2.9 can be estimated, respectively, as below

$$
\begin{aligned}
&O\left(\left|M_{\text{gen}} \cup F_{\text{gen}}\right|\right) \approx O(N)\\
&O\left(K*|D|*\left|\left(M^c_{\text{gen}}+M^m_{\text{gen}}+F^c_{\text{gen}}+F^m_{\text{gen}}\right)\right|\right)\\
&\qquad\qquad \approx O(K*|D|*N)\\
&O\left(\left|M_{\text{gen}} \cup M^c_{\text{gen}} \cup M^m_{\text{gen}}\right|\right) \approx O(N)\\
&O\left(\left|\left(M_{\text{gen}} \cap M_{\text{gen}+1}\right) \cup \left(F_{\text{gen}} \cap F_{\text{gen}+1}\right)\right|\right) \approx O(N).
\end{aligned}
$$

Therefore, time complexity of decoding the best chromosome in step 2.11 is $O(N*K)$. Hence, time complexity of step 2 is $O(K*|D|*N*G_m)$. From the above analysis, we conclude that time complexity of HGCA is $O(K*|D|*N*G_m)$.

Algorithm 2 HGCA

Input: dataset D

Output: clusters of D

Step 1: Parameters and population initialization;

1.1: parameters initialization for N,G_m,P_c,P_m,T_{max},T_w,λ,τ, and current iteration $gen = 0$ (Details in Table III);

1.2: Population initialization: $P = F_0 \cup M_0$ with N chromosomes, where M_0 and F_0 denote male population and female population respectively, and $|F_0| = |M_0|$, for each chromosome $x \in P$, let $x.age = 0$;

Step 2: optimal partition search:

2.1: Compute fitness of chromosomes in population P;

2.2: Temporary population generation: select chromosomes from male population M_{gen} and female population F_{gen} to construct candidate male mating population M'_{gen} and candidate female mating population F'_{gen} with probability p_c;

2.3: Mating operator: select a harmonious mating strategy, i.e. Greedy-EMS, WBM-EMS or UBM-EMS, to construct *FMate* according to F'_{gen} and M'_{gen};

2.4: Crossover operator: for every pair in *FMate* perform VGC operator and produce male offspring M^c_{gen} and female offspring F^c_{gen}, update mating prohibition record of chromosomes in $M^c_{gen} \cup F^c_{gen}$;

2.5: Mutation operator: mutate chromosomes in $M_{gen} \cup F_{gen}$ with probability p_m and construct male offspring M^m_{gen} and female offspring F^m_{gen};

2.6: Evaluation of chromosomes: compute fitness of chromosomes in $\left(M^c_{gen}+M^m_{gen}+F^c_{gen}+F^m_{gen}\right)$ and remove the ICCs in chromosomes;

2.7: Selection operator: select $\lfloor N \times 0.4 \rfloor$ best chromosomes from $\left(M_{gen}+M^c_{gen}+M^m_{gen}\right)$ to construct M_{gen+1} and randomly select $\lfloor N \times 0.1 \rfloor$ chromosomes from the rest of $\left(M_{gen}+M^c_{gen}+M^m_{gen}\right)$ into M_{gen+1}, similarly, F_{gen+1} is constructed;

2.8: Adaptive adjustment of weight: if $gen < T_w$ then $w_1 = 1$, elseif fitness of the best chromosome has not been improved in successive $T_{\max}$ generations then $w_1 = \tau w_1$;

2.9: Update of chromosome age: for each chromosome $x \in \left(M_{gen} \cap M_{gen+1}\right) \cup \left(F_{gen} \cap F_{gen+1}\right)$ do $x.age = x.age + 1$;

2.10: $gen = gen + 1$;

2.11: Repeat from step 2.1 to step 2.10 until $gen >= G_m$, and return the clustering outcome corresponding to the chromosome with best fitness.

IV. Experimental Study

In this section, we empirically evaluate the proposed algorithm from two aspects. First, we apply our novel algorithm to automatically search the number of clusters and cluster centers, and subsequently evaluate its effectiveness and efficiency. Second, we experimentally investigate the impact of mating prohibition schemes and mating strategies on the quality of clustering result.

A. Experiment Datasets and Setup

We evaluate HGCA using artificial datasets [8], [43] and real-world datasets [44], [45], [49]. Statistics of the datasets are summarized in Table II. From Fig. 7, we can see that there are different degrees of overlapping between clusters for each artificial dataset with Gaussian or ellipsoid distribution. And real-world datasets are used as evaluation benchmarks when comparing with some state-of-the-art clustering methods. Dataset Iris and Letter are both from the UCI repository. Iris contains 50 instances from three classes, where each class refers to a type of iris plant and one class is linearly separable from the other two, and the two classes are not linearly separable from each other. Letter contains 20 000 character images, where each corresponds to one of the 26 capital letters in the English alphabet. 20Newsgroups is a collection of 18 846 newsgroup documents partitioned almost evenly across 20 different newsgroups [26]. Vowel, consisting of 871

TABLE II
DESCRIPTION OF THE EXPERIMENT DATASETS

Type	Dataset	#instances	#attributes	#clusters
Real-life	Iris	150	4	3
	Vowel	871	36	6
	C-Cube	10000	34	10
	20Newsgroups	18846	50	20
	Letter	20000	16	26
	Animal(PHOG)	30475	30	50
Artificial	DS1	300	2	6
	DS2	500	2	9
	DS3	5000	2	15
	DS4	5000	2	15

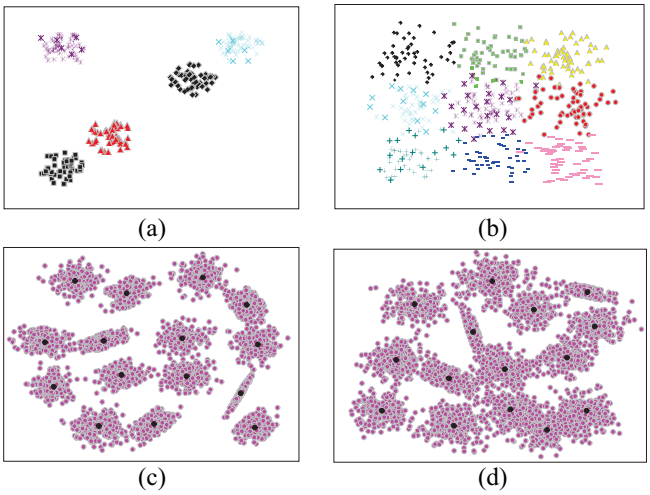

Fig. 7. Artificial datasets. (a) DS1. (b) DS2. (c) DS3. (d) DS4.

Indian Telugu vowel sounds, has three features corresponding to the first, second, and third vowel frequencies, and there are six overlapping classes. Animal(PHOG) is a large-scale dataset of animal images, which consists of six features, 50 classes, and 30 475 samples, where we choose visual feature PyramidHOG(PHOG) to represent each sample. Moreover, in our experiments the dimensionality of 20newsgroups and Animal(PHOG) are reduced to 50 and 30, respectively, by latent semantic indexing and convolutional neural network techniques in a preprocessing step. Since all the clustering algorithms depend on initializations, we repeat all the methods 50 times using random initialization and report the average performance. The parameters of HGCA are given in Table III.

It is well-known that how to reasonably determine and adjust parameters for GAs, for example, population size, crossover probability, mutation probability, etc., is an open problem in evolutionary computation [52], [53]. We give the detailed parameter setting in Table III. The rationale for choosing the values in Table III is that through extensive preliminary experiments on widely-used datasets, including small datasets such as Wine, Glass, Seeds, Iris, Lungcancer, Parkinsons, etc., and small samples from large scale datasets such as C-Cube, 20Newsgroups, Letter, and Animal(PHOG), we observe that although different datasets have different parameter settings that produce high-quality clusters, the parameter setting in Table III can produce better clusters in most cases.

TABLE III
PARAMETER SETTINGS

Parameter	Value	Description
N	60	Population size
G_m	200	Number of iteration
P_c	0.75	Selection probability
P_m	0.05	Mutation probability
T_{max}	20	Number of successive generations in which fitness of the best chromosome has not been improved
T_w	10	Number of the previous generations where age influence is ignored
λ	0.75	Threshold for UBM-EMS
τ	0.85	Adjustment factor for fitness attractiveness weight

B. Experimental Results

1) Comparison of Clustering Performance: In order to empirically evaluate clustering performance of HGCA, we focus on three major aspects: 1) the ability to find the optimal number of clusters; 2) quality of the solution measured by DBI, CS, and VRC; and 3) computational cost to find the solution. And we empirically compare our HGCA algorithm with four state-of-the-art metaheuristic clustering algorithms (MCAs), i.e., ACDE [20], genetic clustering for unknown K (GCUK) [10], TGCA [15], DCPSO [18], and two non-MCAs, i.e., Shi and Malik (SM) [46] and K-means.

a) Comparison of search ability for K: In this section, we evaluate the search ability for the number of clusters K, by comparing HGCA with the four MCAs only, since K-means accepts K as an input parameter specified by the user. To make comparison quantitatively, we introduce an index R_c, as defined in (12), to denote the ratio of how many times the correct number of clusters K is identified. The results on real-world and artificial datasets are given in Tables IV and V, respectively

$$R_c = \frac{\text{Runs having the correct number of clusters}}{\text{Total runs}} \times 100. \tag{12}$$

From the results we can see that first, for datasets with clear cluster boundaries, such as artificial dataset DS1, R_c of HGCA is similar to those of the four MCAs. However, for datasets with moderately overlapping clusters such as DS3 and Iris, HGCA achieves a similar R_c as that of TGCA, which is greater than those of ACDE, DCPSO, and GCUK. For instance, we find that GCUK and DCPSO yield two clusters on average on Iris, i.e., one of the clusters corresponds to the Setosa class, whereas the other corresponds to the combination of Veriscolor and Virginica. For those datasets with heavily blurry cluster boundaries, such as artificial dataset DS4 and real-life datasets Vowel, C-Cube, 20Newsgroups, Letter, and Animal(PHOG), we find that R_c of HGCA, without exception, is greater than those R_cs of the other four MCAs.

Second, no matter which measure we adopted, e.g., DBI, CS, or VRC, for chromosome fitness, R_c of HGCA on most datasets is larger than those of the other four MCAs, although different measures may have different influence on R_c of all the five clustering algorithms. Compared with the other four MCAs, HGCA is more robust in chromosome fitness selection

TABLE IV
EXPERIMENT RESULTS ON REAL-LIFE DATASETS USING FITNESS INDEX DBI, CS, AND VRC

Dataset	Algorithm	DBI		CS		VRC	
		R_c	Fitness	R_c	Fitness	R_c	Fitness
Iris	HGCA	**93**	0.414±0.038	**91**	0.613±0.075	**90**	571.28±8.45
	TGCA	91	0.466±0.042	91	0.631±0.098	86	560.63±12.19
	GCUK	23	0.536±0.051	25	0.734±0.104	24	554.77±14.62
	DCPSO	21	0.495±0.129	22	0.746±0.186	27	545.29±22.81
	ACDE	90	0.478±0.088	87	0.666±0.098	85	555.24±11.73
Vowel	HGCA	**70**	0.632±0.031	**71**	1.021±0.045	**69**	593.35±12.25
	TGCA	62	0.702±0.036	65	1.393±0.048	66	585.25±14.07
	GCUK	20	0.743±0.043	23	1.287±0.053	31	580.27±13.4
	DCPSO	22	0.767±0.059	27	1.335±0.054	37	545.49±21.13
	ACDE	65	0.684±0.078	63	1.404±0.043	61	595.61±17.05
C-Cube	HGCA	**76**	0.834±0.034	**79**	1.551±0.084	**70**	593.61±38.06
	TGCA	69	0.905±0.035	68	1.987±0.097	66	552.02±41.25
	GCUK	43	1.101±0.036	48	2.373±0.108	42	517.43±56.72
	DCPSO	32	0.972±0.047	52	2.506±0.129	37	523.95±86.88
	ACDE	64	0.945±0.037	70	2.287±0.077	65	541.38±36.49
20Newsgroups	HGCA	**66**	1.183±0.189	**61**	1.145±0.297	**65**	584.73±26.37
	TGCA	60	1.309±0.226	56	1.555±0.097	53	518.07±49.16
	GCUK	24	1.356±0.246	27	1.962±0.879	31	488.24±78.23
	DCPSO	20	1.582±0.254	31	1.788±0.955	29	504.76±95.57
	ACDE	63	1.247±0.208	54	1.644±0.694	57	536.69±52.04
Letter	HGCA	**62**	1.235±0.204	**60**	1.332±0.282	**59**	592.22±29.58
	TGCA	52	1.399±0.215	49	1.461±0.135	46	527.61±51.45
	GCUK	24	1.472±0.237	31	1.994±0.657	30	479.03±86.05
	DCPSO	20	1.634±0.266	34	1.686±0.863	35	523.29±90.92
	ACDE	48	1.278±0.213	52	1.573±0.705	49	545.81±68.27
Animal(PHOG)	HGCA	**53**	1.825±0.189	**54**	1.963±0.297	**52**	652.85±59.82
	TGCA	32	2.006±0.226	34	2.208±0.097	35	613.51±72.46
	GCUK	26	2.182±0.246	25	2.818±0.879	30	571.44±78.23
	DCPSO	28	2.424±0.254	29	2.367±0.955	31	610.05±86.89
	ACDE	35	1.985±0.208	34	2.199±0.694	33	621.18±62.04

TABLE V
EXPERIMENT RESULTS ON ARTIFICIAL DATASETS USING FITNESS INDEX DBI, CS, AND VRC

Dataset	Algorithm	DBI		CS		VRC	
		R_c	Fitness	R_c	Fitness	R_c	Fitness
DS1	HGCA	**97**	0.3517±0.036	95	0.3625±0.029	**96**	523.48±11.32
	TGCA	97	0.3524±0.035	94	0.3787±0.032	96	514.78±21.25
	GCUK	95	0.4325±0.038	**96**	0.4361±0.031	95	487.79±20.74
	DCPSO	94	0.4117±0.043	95	0.4208±0.048	94	502.82±29.97
	ACDE	94	0.3724±0.029	95	0.3876±0.034	93	516.06±13.46
DS2	HGCA	**66**	0.5802±0.035	**65**	0.5935±0.041	**63**	388.79±14.57
	TGCA	62	0.6057±0.038	61	0.6007±0.039	58	361.51±20.16
	GCUK	49	0.6116±0.036	47	0.6329±0.037	44	340.23±18.75
	DCPSO	57	0.6005±0.042	52	0.6251±0.046	54	357.77±25.59
	ACDE	61	0.6021±0.033	58	0.6014±0.036	55	376.01±15.34
DS3	HGCA	**62**	0.5438±0.047	56	0.8864±0.056	**58**	2376.49±195.51
	TGCA	62	0.6057±0.038	**61**	0.6007±0.039	55	2294.72±219.33
	GCUK	45	0.5951±0.059	43	0.9711±0.073	46	2024±274.68
	DCPSO	49	0.5825±0.076	51	0.9096±0.082	52	1983.75±236.44
	ACDE	52	0.5901±0.049	48	0.8723±0.053	51	2250.13±208.84
DS4	HGCA	**51**	0.644±0.033	**49**	0.695±0.036	**47**	2186.49±156.54
	TGCA	46	0.652±0.044	44	0.7209±0.045	45	2039.11±190.19
	GCUK	35	0.692±0.047	34	0.9626±0.049	39	1967±197.06
	DCPSO	48	0.666±0.051	43	0.714±0.059	47	2015.58±213.35
	ACDE	49	0.668±0.045	47	0.785±0.047	40	2112.67±194.26

when detecting the optimal cluster number. In order to better analyze robustness, we average R_cs of the five MCAs on all datasets and the results are depicted in Fig. 8. From Fig. 8, we can see that in terms of search ability for optimal number of clusters, HGCA significantly outperforms other four MCAs in terms of fitness robustness.

Third, R_c on real-life datasets gradually decreases with the number of true clusters hidden in those datasets, but R_c of HGCA increases slowly compared with the other four MCAs. For instance, in Letter with 26 clusters R_c of HGCA with fitness DBI is 62 and the largest R_c value of other four MCAs is 48; while in Animal(PHOG) with 50 clusters HGCA's R_c is 53 and the largest R_c of the other four MCAs is 35.

From the above analysis, we conclude that HGCA is superior to the state-of-the-art MCAs for identifying the optimal number of clusters. This means that introduction of genetic operators into clustering algorithms is beneficial to finding the correct number of clusters of datasets. Nevertheless, we admit that it is tricky to identify the real number of

TABLE VI
t-TEST RESULTS OF NMI OF HGCA AND THE OTHER FOUR MCAS ON REAL-LIFE DATASETS

Dataset	Criterion	HGCA	TGCA	GCUK	DCPSO	ACDE	P	95% CI	Sig.
Iris	DBI	**0.9337±0.019**	0.9012±0.022	0.8621±0.023	0.8735±0.048	0.8891±0.025	<0.0001	[0.0294, 0.0405]	√
	CS	**0.9354±0.019**	0.8943±0.029	0.8524±0.026	0.8635±0.046	0.8883±0.0093	<0.0001	[0.0385, 0.0512]	√
	VRC	**0.9214±0.0181**	0.9158±0.031	0.8485±0.025	0.8396±0.037	0.8726±0.011	0.0307	[-0.0067, 0.0216]	×
Vowel	DBI	**0.6523±0.021**	0.6136±0.025	0.5708±0.023	0.5963±0.026	0.6022±0.027	<0.0001	[0.0289, 0.0507]	√
	CS	**0.6537±0.021**	0.6201±0.028	0.5884±0.026	0.5776±0.034	0.5971±0.025	<0.0001	[0.0301, 0.0528]	√
	VRC	**0.6479**±0.019	0.6054±0.021	0.5655±**0.018**	0.5726±0.029	0.5969±0.027	<0.0001	[0.0213,0.0429]	√
C-Cube	DBI	**0.5452**±0.021	0.5037±0.031	0.4526±0.027	0.4871*±0.031	0.4852±**0.019**	<0.0001	[0.0074, 0.0253]	√
	CS	**0.5208±0.026**	0.4823±0.044	0.4807±0.035	0.5018±0.076	0.5014±0.032	<0.0001	[0.0215, 0.0404]	√
	VRC	**0.5313**±0.041	0.4908±0.037	0.5168±0.057	0.5025*±0.088	0.4807±**0.034**	<0.0001	[0.0167, 0.0369]	√
20Newsgroups	DBI	**0.5143±0.056**	0.4915±0.074	0.4634±0.062	0.4806±0.089	0.4857±0.061	<0.0001	[0.0114, 0.0436]	√
	CS	**0.4986**±0.052	0.4709±**0.043**	0.4627±0.048	0.4719±0.078	0.4814±0.0502	<0.0001	[0.0118, 0.0367]	√
	VRC	**0.4822±0.056**	0.4662±0.069	0.4551±0.077	0.4532±0.082	0.474±0.064	0.1307	[-0.0034, 0.0197]	×
Letter	DBI	**0.5207±0.055**	0.4977±0.078	0.4801±0.065	0.4847±0.082	0.4922±0.063	<0.0001	[0.0107, 0.0385]	√
	CS	**0.5104**±0.053	0.4888±0.051	0.4627±0.056	0.4781±0.066	0.4991±**0.049**	<0.0001	[0.0126, 0.0352]	√
	VRC	**0.5018±0.059**	0.4806±0.062	0.4775±0.071	0.4841±0.077	0.4972±0.061	0.1458	[-0.0042, 0.0181]	×
Animal(PHOG)	DBI	**0.5025±0.071**	0.4773±0.083	0.4358±0.084	0.4712±0.092	0.4803±0.078	<0.0001	[0.0098, 0.0246]	√
	CS	**0.4901±0.073**	0.4711±0.075	0.4317±0.088	0.4732±0.089	0.4793±0.091	<0.0001	[0.0222, 0.0419]	√
	VRC	**0.4903±0.072**	0.4685±0.077	0.4265±0.073	0.4594±0.083	0.4838±0.074	<0.0001	[0.0037, 0.0246]	√

TABLE VII
t-TEST RESULTS OF NMI OF HGCA AND THE OTHER FOUR MCAS ON ARTIFICIAL DATASETS

Dataset	Criterion	HGCA	TGCA	GCUK	DCPSO	ACDE	P	95% CI	Sig.
DS1	DBI	**0.93±0.019**	0.928±0.027	0.907±0.028	0.918±0.034	0.9224±0.021	0.3256	[0.0004, 0.0126]	×
	CS	**0.9224±0.014**	0.921±0.025	0.9153±0.0208	0.912±0.023	0.907±0.023	0.2986	[-0.0013, 0.0115]	×
	VRC	**0.9212**±0.021	0.919±0.031	0.911±**0.014**	0.9164±0.025	0.9105±0.028	0.4834	[-0.0017, 0.0097]	×
DS2	DBI	**0.884**±0.033	0.841±0.0016	0.8306±**0.011**	0.8421±0.035	0.8513±0.0182	<0.0001	[0.0209, 0.0444]	√
	CS	**0.8759±0.0249**	0.843±0.027	0.835±0.025	0.841±0.031	0.8429±0.0208	<0.0001	[0.0215, 0.0414]	√
	VRC	**0.8687**±0.026	0.856±**0.012**	0.8395±0.014	0.8431±0.0195	0.8407±0.026	<0.0001	[0.0145, 0.0348]	√
DS3	DBI	**0.946**±0.02	0.931±0.024	0.926±**0.018**	0.935±0.028	0.9377±0.022	0.108	[-0.0019, 0.0187]	×
	CS	**0.9414±0.0281**	0.929±0.0263	0.921±0.0258	0.9276±0.0234	0.918±0.0247	0.0190	[0.0021, 0.0249]	×
	VRC	**0.9417±0.0209**	0.9304±0.0275	0.9305±0.0231	0.9315±0.0246	0.928±0.0211	0.0294	[0.001, 0.0193]	×
DS4	DBI	**0.9314±0.019**	0.917±0.031	0.908±0.023	0.911±0.045	0.911±0.0119	<0.0001	[0.0098, 0.0247]	√
	CS	**0.919**±0.0219	0.892±0.038	0.894±0.038	0.902±0.048	0.896±**0.024**	<0.0001	[0.0084, 0.0263]	√
	VRC	**0.929**±0.021	0.902±0.027	0.881±0.032	0.903±0.028	0.906±**0.013**	<0.0001	[0.0151, 0.0306]	√

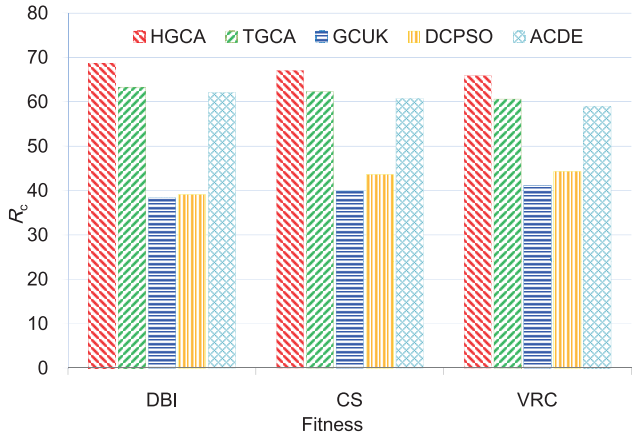

Fig. 8. R_c of HGCA, TGCA, GCUK, DCPSO, and ACDE with respect to fitness.

clusters, given the unsatisfactory performance of all algorithms on datasets with various degree of overlapping between clusters.

b) Comparison of clustering quality: A good clustering algorithm not only can automatically identify the optimal number of clusters, but obtain high-quality clusters. To evaluate quality of data clusters resulting from clustering algorithms, we use the normalized mutual information (NMI) in [47], given in Formula (13), to measure the agreement between user-labeled clusters and the calculated clusters. Obviously, NMI value is 1 when a clustering solution perfectly matches the user-labeled clusters, and NMI approaches to 0 in the case of random partitioning

$$\mathrm{NMI}(C^r, C^t) = \frac{-2\sum_{C_m\in C^r}\sum_{C_n\in C^t}\frac{|C_m\cap C_n|}{|\mathrm{D}|}\log\left(\frac{|\mathrm{D}||C_m\cap C_n|}{|C_m||C_n|}\right)}{\sum_{C_m\in C^r}\frac{|C_m|}{|\mathrm{D}|}\log\left(\frac{|C_m|}{|\mathrm{D}|}\right)+\sum_{C_n\in C^t}\frac{|C_n|}{|\mathrm{D}|}\log\left(\frac{|C_n|}{|\mathrm{D}|}\right)} \quad (13)$$

where C^r and C^t denote the resulting clusters and the true clusters of a dataset, respectively, and C_m denotes a cluster.

To verify the clustering quality, we use *t*-test to compare the means of the results produced by HGCA and the best one among the four MCAs and the two non-MCAs. *t*-test assumes that the data have been sampled from a normally distributed population. From the central limit theorem, one may note that as sample size increases, the sampling distribution of the mean approaches a normal distribution regardless of the shape of the original population. A sample size around 40 allows the normality assumptions conducive to performing the *t*-test [48]. The experiment results on all of the datasets are shown in Tables VI–VIII. From these tables, we obtain the following observations.

First, compared with the other four MCAs, HGCA has larger NMIs on most datasets. And *t*-test result indicates that HGCA shows an overwhelming advantage over those four MCAs on datasets with fuzzy cluster boundaries, e.g., on dataset C-Cube, Vowel, etc., although its clustering quality is not significantly better than those four MCAs on four

TABLE VIII
t-TEST RESULTS OF THE NMIS BETWEEN HGCA AND TWO NON-MCAS ON DATASETS

Dataset	HGCA	K-Means	SM	95% CI	P	Sig.
Iris	**0.935**±0.017	0.728±0.066	0.754±**0.014**	[0.1556, 0.1992]	< 0.0001	√
Vowel	**0.652**±0.021	0.547±0.017	0.616±**0.015**	[0.0778, 0.0943]	< 0.0001	√
C-Cube	**0.536**±0.036	0.423±0.049	0.481±**0.037**	[0.0643, 0.0861]	< 0.0001	√
20Newsgroups	**0.512±0.043**	0.386±0.065	0.459±0.051	[0.0415, 0.0637]	< 0.0001	√
Letter	**0.521**±0.054	0.354±0.052	0.473±**0.046**	[0.0552, 0.0824]	< 0.0001	√
Animal(PHOG)	**0.501**±0.071	0.343±0.064	0.438±**0.061**	[0.0529, 0.0965]	< 0.0001	√
DS1	**0.93**±0.019	0.927±0.057	0.928±**0.013**	[-0.015, 0.0216]	0.7244	×
DS2	**0.884**±0.033	0.831±0.037	0.852±**0.027**	[0.0365, 0.0679]	< 0.0001	√
DS3	**0.946±0.02**	0.934±0.025	0.941±0.021	[0.0057, 0.0262]	0.0037	×
DS4	**0.931±0.019**	0.902±0.021	0.913±0.023	[0.0214, 0.0376]	< 0.0001	√

datasets with clear cluster structures, e.g., on artificial dataset DS1 and DS3.

Second, results in Table VIII also show that in most datasets, i.e., all real-life datasets and artificial datasets DS2 and DS4, HGCA outperforms the non-MCAs, i.e., *K*-means and SM, in terms of clustering quality. Interestingly, SM is overwhelmingly stable in clustering. The explanation is that low-dimensional manifold embedded in the high-dimensional vector space produced by spectral mapping makes cluster boundary more clear, which improves the clustering stability of SM.

Third, all MCAs, i.e., HGCA, TGCA, DCPSO, ACDE, and GCUK, can produce better clustering quality than *K*-means on most datasets, but there are some exceptions where *K*-means performs better, for example, GCUK and ACDE on DS4, and GCUK on DS3. This demonstrates that population intelligence-based optimization techniques can facilitate search for optimal cluster structure.

Fourth, different fitness evaluation functions also play an important role in searching for clusters hidden in datasets, for instance, DBI fitness outperforms the other two fitness indexes on most of the datasets.

Finally, comparing NMI in Tables VI–VIII and R_c in Tables IV and V, we can see that determining optimal assignment of data objects to clusters and detecting optimal number of clusters are different, since larger R_c does not necessarily correspond to larger NMI.

From the aforementioned observations, it is not difficult to conclude that: 1) MCAs are better in discovering cluster structure of datasets than non-MCAs, meaning that introduction of metaheuristic operators can improve clustering quality and 2) comparing with TGCA, ACDE, DCPSO, and GCUK, our approach performs the best in terms of clustering quality.

c) Comparison of CPU time: In this section, we investigate the efficiency of HGCA, by comparing running time of HGCA and the other four evolutionary algorithms, i.e., TGCA, GCUK, ACDE, and DCPSO. For the sake of fairness, we first need to introduce a time measurement. Obviously, the number of iterations or generations cannot be used as a time measure, since the algorithms perform different amount of work in their inner loops and they have different population sizes. Hence, we choose the elapsed CPU time as a measure instead of number of generations or iterations. Moreover, we use a maximum number of iterations, specified by user, as termination condition for the four algorithms. We record the running time of the three algorithms according to Rule 3, shown below.

TABLE IX
RUNNING TIME OF HGCA, TGCA, GCUK, ACDE, AND DCPSO ON DATASETS

Dataset	TGCA	GCUK	ACDE	DCPSO	HGCA
Iris	18.54	19.36	19.26	**18.34**	18.71
Vowel	116.08	129.87	110.63	135.52	**103.46**
C-Cube	1806.42	2105.93	1828.41	1826.85	**1749.43**
20Newsgroups	3829.55	3978.82	3943.65	3875.14	**3536.48**
Letter	3016.74	3108.44	2925.15	2833.61	**2625.02**
Animal(PHOG)	5205.23	5697.85	5167.94	5038.62	**4874.27**
DS1	15.57	16.94	15.63	**14.86**	15.29
DS2	32.09	35.18	29.76	28.23	**24.01**
DS3	690.58	725.43	703.57	685.66	**682.15**
DS4	836.69	914.73	859.04	827.55	**778.98**

Rule 3: If the termination condition of the algorithm has been satisfied but its clustering quality index NMI has not achieved the corresponding value in Tables VI and VII, then we do not record the running time. In other words, if its clustering quality index NMI has achieved the corresponding value in Tables VI and VII before the termination condition is met, then we break the loop and record the time used in the loop.

The experiment results are listed in Table IX. From the table, we can see that CPU time of HGCA is as good as TGCA, GCUK, ACDE, and DCPSO on datasets with easily discernable clusters, such as Iris, DS1, and DS3. However, HGCA exhibits remarkable advantages on datasets with indistinguishable clusters such as Letter, C-Cube, 20Newsgroups, and Animal(PHOG). Explanation to this observation is obvious. For datasets with easily discernable clusters, even algorithms with fair optimization ability can efficiently discover the cluster structures with given NMI, since it is difficult for HGCA to show its time performance advantage. However, datasets with complex and ambiguous cluster structure will benefit the optimization ability of HGCA. From the above observations, we conclude that for datasets with either clear cluster structures or ambiguous cluster structures, HGCA performs as good as or even better than the other four MCAs in terms of CPU time used in clustering.

2) Effectiveness of Crossover Operator VGC: In this section, we examine whether and to what extent VGC affects clustering quality of HGCA. Given that VGC operator is closely related to gender attribute of chromosomes, we remove gender information of chromosomes in population, and replace mating strategy and VGC with traditional selection operator (roulette wheel selection) and crossover operator (two-point crossover), respectively. We call this simplified HGCA SHGCA. From Table X, we can see that for both artificial and real-life

TABLE X
EFFECTIVENESS OF VGC

Dataset	HGCA	SHGCA
Iris	0.9337±0.019	0.8724±0.021
Vowel	0.6523±0.021	0.5813±0.049
C-Cube	0.5452±0.021	0.4698±0.028
20Newsgroups	0.5143±0.056	0.4784±0.061
Letter	0.5207±0.055	0.4902±0.063
Animal(PHOG)	0.5025±0.071	0.4405±0.075
DS1	0.93±0.019	0.928±0.021
DS2	0.884±0.033	0.8415±0.013
DS3	0.946±0.02	0.925±0.019
DS4	0.9314±0.019	0.911±0.022

TABLE XI
EFFECTIVENESS OF HARMONIOUS MATING STRATEGY

Dataset	HGCA	SHGCA2
Iris	0.9337±0.019	0.8736±0.022
Vowel	0.6523±0.021	0.5907±0.071
C-Cube	0.5452±0.021	0.4714±0.034
20Newsgroups	0.5143±0.056	0.4812±0.065
Letter	0.5207±0.055	0.4881±0.067
Animal(PHOG)	0.5025±0.071	0.4433±0.086
DS1	0.93±0.019	0.915±0.024
DS2	0.884±0.033	0.854±0.017
DS3	0.946±0.02	0.924±0.018
DS4	0.9314±0.019	0.918±0.026

TABLE XII
IMPACT OF PROHIBITION SCHEMES ON THE CLUSTERING QUALITY

Dataset	NMP	MPLR	MPCR
Iris	0.9118±0.0645	0.9274±**0.0381**	**0.9356**±0.042
Vowel	0.611±0.0802	0.645±0.0337	**0.6534±0.028**
C-Cube	0.5165±0.049	0.5304±0.036	**0.5356±0.026**
20Newsgroups	0.4806±0.0517	0.4908±0.0236	**0.5103±0.014**
Letter	0.494±0.0732	0.502±0.068	**0.521±0.054**
Animal(PHOG)	0.4513±0.098	0.485±0.084	**0.501±0.071**
DS1	**0.936**±0.027	0.934±**0.016**	0.935±0.017
DS2	0.841±0.065	**0.884**±0.039	0.876±**0.022**
DS3	0.912±0.0456	**0.946**±0.0333	0.935±**0.0197**
DS4	0.907±0.0561	**0.9314**±0.0308	0.9185±**0.019**

TABLE XIII
IMPACT OF MATING STRATEGIES ON CLUSTERING QUALITY

Dataset	Greedy-EMS	UBM-EMS	WBM-EMS
Iris	0.9054±0.0972	**0.9388±0.0253**	0.9127±0.0306
Vowel	0.617±0.0914	**0.6535±0.0289**	0.6321±0.0304
C-Cube	0.5034±0.0875	**0.5295±0.0486**	0.5284±0.0532
20Newsgroups	0.4751±0.0693	0.5121±0.0472	**0.4708±0.0408**
Letter	0.5026±0.0897	**0.521±0.053**	0.528±0.0471
Animal(PHOG)	0.4689±0.0881	0.501±0.071	**0.5083±0.0653**
DS1	0.931±0.023	0.933±**0.016**	**0.935**±0.018
DS2	0.852±0.094	**0.886±0.063**	0.873±0.065
DS3	0.915±0.0747	**0.942±0.0365**	0.931±0.0402
DS4	0.902±0.0885	**0.9307±0.0462**	0.912±0.0476

datasets, HGCA outperforms SHGCA in terms of average and standard error of NMI. This indicates that dividing chromosome population into male and female subpopulations and introduction of VCG based on gender attribute of chromosome are beneficial to boosting clustering validity and stability of GCAs.

3) Effectiveness of Mating Strategy: Mating strategy is the core feature of HGCA, and it is necessary to empirically justify its effectiveness in clustering performance improvement. To this end, we replace mating strategy used in HGCA with a traditional selection operator, i.e., roulette wheel selection (RWS) [51]. Specifically, we use RWS to generate female parents F and male parents M from current female subpopulation and male subpopulation, respectively, and construct parent chromosome pairs FMate in the order that they appear in F and M. For instance, let $M = \{a, b, c, d, e\}$ and $F = \{1, 2, 3, 4, 5\}$, we have FMate $= \{(a, 1), (b, 2), (c, 3), (d, 4), (e, 5)\}$. We call this simplified HGCA SHGCA2. The experiment results are given in Table XI. From Table XI, we can see that for all the datasets, HGCA can discover much higher quality, in terms of average and standard error of NMI, cluster patterns than SHGCA2. The rationale is that SHGCA2 cannot take mating attractiveness between heterosexual chromosomes into consideration while constructing potential crossover pairs, which degrades its ability to search for optimal cluster structures from datasets.

4) Comparison of Mating Prohibition Schemes: As an important part of HGCA, mating prohibition schemes may affect clustering quality. In this section, we conduct some experiments to verify the impact of the three schemes NMP, MPLR, and MPCR on clustering quality. From Table XII, we can see that in terms of mean value of NMI, NMP is the best on DS1, whereas MPLR performs the best on DS2, DS3, and DS4, and MPCR is the best across all real-life datasets. This means that none of the three schemes consistently performs the best in improving clustering quality. The ability of mating prohibition scheme to improve clustering quality depends on the cluster structure of a dataset. Therefore, how to adaptively select a good mating prohibition scheme is an important issue, which belongs to our future work. On the other hand, when examining the stability of the three schemes using standard deviation (SD), we can see that MPCR is the most stable one, followed by MPLR which is more stable than NMP. However, we do not observe a similar trend on Iris and DS1 though. This suggests that the stricter a mating prohibition scheme, the more stable it will be.

5) Comparison of Mating Strategies: Mating strategies are also very important to clustering quality of HGCA. According to population diversity loss caused by the three mating strategies, we can see that WBM-EMS is able to avoid premature and achieve the optimal clustering result. And we have verified this through experiment results given in Table XIII, where the mean values of NMI of WBM-EMS are the largest almost on all datasets, except for DS1. Also, SD of NMI of the three mating schemes shows that clustering stability of WBM-EMS and UBM-EMS surpasses that of Greedy-EMS. Meanwhile, by comparing WBM-EMS with UBM-EMS we discover that although clustering quality of WBM-EMS is better than that of UBM-EMS, the performance gain is marginal, and clustering stabilities of both methods are roughly the same.

To provide better guidance for users to make a choice among the three mating strategies, we also conduct detailed experiments to investigate the time cost of the three strategies. Table XIV compares the time costs of Greedy-EMS, WBM-EMS, and UBM-EMS. From Table XIV, we can see that in general, CPU time costs increase steadily for Greedy-EMS, UBM-EMS, and WBM-EMS, and the gaps between CPU time of Greedy-EMS, UBM-EMS, and WBM-EMS are relatively small on datasets with easily discernable clusters such as DS1. However, a noticeable gap in time costs occurs

TABLE XIV
RUNNING TIME OF THE THREE MATING STRATEGIES

Dataset	Greedy-EMS	UBM-EMS	WBM-EMS
Iris	16.24	17.49	18.71
Vowel	87.18	95.54	103.46
C-Cube	1519.93	1737.77	2386.92
20Newsgroups	3108.47	3576.55	4789.76
Letter	1886.25	2625.02	4205.49
Animal(PHOG)	2761.51	4874.27	8523.72
DS1	14.58	15.07	15.29
DS2	19.75	22.93	24.01
DS3	591.03	647.59	682.15
DS4	588.23	734.96	778.98

on datasets with fuzzy cluster boundaries such as C-Cube, 20Newsgroups, and DS4. Rationale to the above observation is that the number of mating pairs with attractiveness greater than threshold λ decreases on high dimensional datasets, which degrades the efficiency of UBM-EMS. Moreover, regardless of which mating strategy we use, i.e., Greedy-EMS, UBM-EMS, and WBM-EMS, the time cost of HGCA on larger datasets such as Animal(PHOG) and 20Newsgroups is still a bit high. Hence, improvement of the efficiency of HGCA is left in our future work.

From the above discussion, we can see that it is tricky to choose suitable mating strategy for a clustering task. However, the experiment results suggest that it is more appropriate to choose UBM-EMS, which can obtain good clustering results without incurring high computational cost in most cases.

V. CONCLUSION

In this paper, we have presented a novel GCA to automatically select the number of clusters K, and the generated clusters are better than those produced by the state-of-the-art methods. Our main finding is that traditional variable length coding solution can incur an exponential time complexity due to larger search space. Motivated by eugenic theory, our algorithm uses harmonious mating and crossover operator to guide the population into better convergence path and thus improves clustering quality. We have conducted extensive experiments to evaluate our novel method on real-life and artificial datasets, and the results show that HGCA can divide the data into meaningful clusters without knowing the number of clusters in advance. We have also compared our algorithm with some existing state-of-the-art algorithms, and the results confirm that our approach is more effective and efficient for data clustering.

APPENDIX

Theorem 1: The length of newborn chromosomes produced by VGC is variable with probability $1-(2/\max(|m|, |f|) - 1)$.

Proof: Without loss of generality, suppose x is a newborn chromosome. From concatenation operation in step 2 of Algorithm 2, we can deduce that length of x is $|x| = rm+|f|-rf$. If length of a newborn chromosome is equal, i.e., either $|x| = |f|$ or $|x| = |m|$ holds, then we have the following two derivatives.

1) $|x| = |f| \Rightarrow |x| = rm + |f| - rf \Rightarrow rm = rf$.
2) $|x| = |m| \Rightarrow |x| = rm + |f| - rf \Rightarrow |m| - rm = |f| - rf$.

From derivatives 1) and 2), we conclude that both before and after two random crossover points (rm, rf), the lengths of chromosome segments are equal. And it is not difficult to find that the probabilities of the occurrence of derivative 1) and derivative 2) are the same. Since the probability of the equation $rm = rf$ holds, due to

$$\text{Prob}(rm = rf) = \frac{1}{\max(|m|, |f|) - 1}$$

so we have

$$\text{Prob}((|x| = |f|) \cup (|x| = |m|)) = \frac{2}{\max(|m|, |f|) - 1}.$$

Therefore, the length of a newborn chromosome is variable with probability $1 - (2/\max(|m|, |f|) - 1)$. ■

Theorem 2: The gender proportion of offspring generation generated by VGC operator is equal in probability to the gender proportion of the chromosomes parent generation.

Proof: Without loss of generality, let a and b be two chromosomes of a parent generation, their corresponding clusters satisfy $|a| \geq |b|$, and gene length of a new chromosome of the offspring generation containing a be $a' \in [1, |a| - 1]$ and containing b be $b' \in [1, |b| - 1]$. According to the rule of judging the gender of chromosomes of offspring generation (step 3 of Algorithm 1), to prove Theorem 2 we have to prove that the probability of being a's gender is 0.5. In other words, the probability of being b's gender is 0.5. It is easy to find that there is only one case for a chromosome to be the a's gender, based on which the probability is relatively easy to compute. Therefore, we need to compute the probability of occurrence of the event *"a chromosome of offspring generation being a's gender."*

We denote the probability of occurrence of the event *a chromosome of off-spring generation being a's gender* by p_a, the probability of occurrence of the event *"the gene length of the chromosomes of off-spring generation"* by $p_{a' \geq b'}$, the probability of occurrence of the event *"the chromosome of off-spring generation being a's gender, when $a' \geq b'$ for the gene length of the chromosomes of offspring generation"* by $p_{a|a' \geq b'}$. Then the above probabilities satisfy $p_a = p_{a' \geq b'} \times p_{a|a' \geq b'}$.

According to the crossover rule in Algorithm 1, when the event *"$a' \geq b'$ for the gene length of the chromosomes of off-spring generation"* happened, if a randomly generated number $r \in [1, 2 \times |a| - |b|]$ satisfies $r < |a|$, then the event *the chromosome of off-spring generation being a's gender* happens, i.e., we have the probability $p_{a|a' \geq b'} = (|a| - 1/2 \times |a| - |b|)$.

Since $a' \in [1, |a| - 1]$ and $b' \in [1, |b| - 1]$, there are $(|a| - 1)(|b|-1)$ combinations of the gene length a' and b' of the offspring generation. Consider the gene length a'. Assume that the other gene length of a chromosome a of parent generation is a'', then $a' = |a|-a''$, which means that $a' \geq b'$ is equivalent to $|a| - a'' \geq b'$, i.e., a' is equivalent to $|a| \geq a'' + b'$. The possible combinations of a'' and b' are

$$\begin{array}{ll} b' = 1 & a'' \in [1, |a| - 1] \\ b' = 2 & a'' \in [1, |a| - 2] \\ \vdots & \vdots \\ b' = |b| - 1 & a'' \in [1, |a| - |b| + 1]. \end{array}$$

There are $(((|b|-1)(|a|-1+|a|-|b|+1)/2) = ((|b|-1)(2\times|a|-|b|)/2)$ combinations in total. Then we are able to compute the probability $p_{a\geq b} = (((|b|-1)(2\times|a|-|b|)/2)/(|a|-1)(|b|-1)) = (2\times|a|-|b|/2\times(|a|-1))$, and then compute p_a, which is $p_a = p_{a'\geq b'}\times p_{a|a'\geq b'} = (2\times|a|-|b|/2\times(|a|-1))\times(|a|-1/2\times|a|-|b|) = (1/2) = 0.5$. In other words, the gender proportion of offspring generation generated by VGC operator is equal in probability to the gender proportion of the chromosomes parent generation. ■

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees and the Associate Editor for their valuable comments and suggestions, which have improved this paper vastly.

REFERENCES

[1] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 2, pp. 133–155, Mar. 2009.

[2] S. Das, A. Abraham, and A. Konar, "Metaheuristic pattern clustering—An overview," *Metaheuristic Clustering*. Heidelberg, Germany: Springer, 2009, pp. 1–62.

[3] O. A. M. Jafar and R. Sivakumar, "Ant-based clustering algorithms: A brief survey," *Int. J. Comput. Theory Eng.*, vol. 2, no. 5, pp. 1793–8201, 2010.

[4] R. H. Sheikh, M. M. Raghuwanshi, and A. N. Jaiswal, "Genetic algorithm based clustering: A survey," in *Proc. ICETET* Nagpur, India, 2008, pp. 314–319.

[5] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 40–53, Jan. 2014.

[6] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2202–2215, Dec. 2013.

[7] J. Shen, X. Yang, X. Li, and Y. Jia, "Intrinsic image decomposition using optimization and user scribbles," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 425–436, Apr. 2013.

[8] L. Shao, R. Yan, X. Li, and Y. Liu, "From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1001–1013, Jul. 2014.

[9] F. Galton, *Inquiries into Human Faculty and its Development*. London, U.K.: Dent, 1883.

[10] S. Bandyopadhyay and U. Maulik, "Genetic clustering for automatic evolution of clusters and application to image classification," *Pattern Recognit.*, vol. 35, no. 6, pp. 1197–1208, 2002.

[11] Z. Zhu, P. Han, C. Yu, and L. Li, "A dynamic genetic algorithm for clustering Web pages," in *Proc. SEDM*, Chengdu, China, 2010, pp. 506–511.

[12] U. Maulik and S. Bandyopadhyay, "Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 5, pp. 1075–1081, May 2003.

[13] W. Sheng, S. Swift, L. Zhang, and X. Liu, "A weighted sum validity function for clustering with a hybrid niching genetic algorithm," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1156–1167, Dec. 2005.

[14] K. S. Shin, Y.-S. Jeong, and M. K. Jeong, "A two-leveled symbiotic evolutionary algorithm for clustering problems," *Appl. Intell.*, vol. 36, no. 4, pp. 788–799, 2012.

[15] H. He and Y. H. Tan, "A two-stage genetic algorithm for automatic clustering," *Neurocomputing*, vol. 81, pp. 49–59, Apr. 2012.

[16] Y. Liu, T. Özyer, R. Alhajj, and K. Barker, "Integrating multi-objective genetic algorithm and validity analysis for locating and ranking alternative clustering," *Informatica*, vol. 29, no. 1, pp. 33–40, 2005.

[17] Y. Liu, X. Wu, and Y. Shen, "Automatic clustering using genetic algorithms," *Appl. Math. Comput.*, vol. 218, no. 4, pp. 1267–1279, 2011.

[18] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Anal. Appl.*, vol. 8, no. 4, pp. 332–344, 2006.

[19] H. Masoud, S. Jalili, and S. M. H. Hasheminejad, "Dynamic clustering using combinatorial particle swarm optimization," *Appl. Intell.*, vol. 38, no. 3, pp. 289–314, 2013.

[20] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 218–237, Jan. 2008.

[21] S. T. Emlen and L. W. Oring, "Ecology, sexual selection and the evolution of mating systems," *Science*, vol. 197, no. 4300, pp. 215–223, 1977.

[22] L. J. Eshelman and J. D. Schaffer, "Preventing premature convergence in genetic algorithms by preventing incest," in *Proc. ICGA*, San Mateo, CA, USA, 1991, pp. 115–122.

[23] K. Matsui, "New selection method to improve the population diversity in genetic algorithms," in *Proc. SMC*, vol. 1. Tokyo, Japan, 1999, pp. 625–630.

[24] R. Craighurst and W. N. Martin, "Enhancing GA performance through crossover prohibitions based on ancestry," in *Proc. ICGA*, Pittsburgh, PA, USA, 1995, pp. 130–135.

[25] S. De, S. K. Pal, and A. Ghosh, "Genotypic and phenotypic assortative mating in genetic algorithm," *Inf. Sci.*, vol. 105, nos. 1–4, 1998, pp. 209–226.

[26] C. Fernandes, R. Tavares, C. Munteanu, and A. Rosa, "Using assortative mating in genetic algorithms for vector quantization problems," in *Proc. SAC*, Las Vegas, NV, USA, 2001, pp. 361–365.

[27] G. Ochoa, C. Mädler-Kron, R. Rodriguez, and K. Jaffe, "Assortative mating in genetic algorithms for dynamic problems," in *Proc. EvoWorkshops*, Lausanne, Switzerland, 2005, pp. 617–622.

[28] C.-K. Ting, S.-T. Li, and C. Lee, "On the harmonious mating strategy through tabu search," *Inf. Sci.*, vol. 156, nos. 3–4, pp. 189–214, 2003.

[29] C. Fernandes and A. C. Rosa, "Self-adjusting the intensity of assortative mating in genetic algorithms," *Soft Comput.*, vol. 12, no. 10, pp. 955–979, 2008.

[30] A. Juan and E. Vidal, "Comparison of four initialization techniques for the *K*-medians clustering algorithm," in *Proc. SSPR/SPR*, Alicante, Spain, 2000, pp. 842–852.

[31] S. Saitta, B. Raphael, and I. F. C. Smith, "A comprehensive validity index for clustering," *Intell. Data Anal.*, vol. 12, no. 6, pp. 529–548, 2008.

[32] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[33] C.-H. Chou, M.-C. Su, and E. Lai, "A new cluster validity measure and its application to image compression," *Pattern Anal. Appl.*, vol. 7, no. 2, pp. 205–220, 2004.

[34] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat.*, vol. 3, no. 1, pp. 1–27, 1974.

[35] Z. Kowalczuk and T. Bialaszewski, "Improving evolutionary multi-objective optimization using genders," in *Proc. ICAISC*, Zakopane, Poland, 2006, pp. 390–399.

[36] D. Vrajitoru, "Simulating gender separation and mating constraints for genetic algorithms," Intell. Syst. Lab., Indiana University South Bend, South Bend, IN, USA, Tech. Rep. TR-20050520-1, 2005.

[37] *Gonochorism-Wikipedia*. Accessed on Oct. 15, 2014. [Online]. Available: http://en.wikipedia.org/wiki/Gonochorism

[38] C. A. C. António, "A hierarchical genetic algorithm with age structure for multimodal optimal design of hybrid composites," *Struct. Multidiscipl. Optim.*, vol. 31, no. 4, pp. 280–294, 2006.

[39] N. Kubota, T. Fukuda, F. Arai, and K. Shimojima, "Genetic algorithm with age structure and its application to self-organizing manufacturing system," in *Proc. ETFA*, Tokyo, Japan, 1994, pp. 472–477.

[40] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist.*, vol. 2, nos. 1–2, pp. 83–97, 1955.

[41] A. V. Goldberg and R. E. Tarjan, "A new approach to the maximum-flow problem," *J. ACM*, vol. 35, no. 4, pp. 921–940, 1988.

[42] *UCI Repository of Machine Learning Databases*. Accessed on Mar. 10, 2014. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[43] *Synthetic Datasets*. Accessed on Mar. 10, 2014. [Online]. Available: http://cs.joensuu.fi/sipu/datasets/

[44] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[45] F. Camastra, M. Spinetti, and A. Vinciarelli, "Offline cursive character challenge: A new benchmark for machine learning and pattern recognition algorithms," in *Proc. ICPR*, Hong Kong, 2006, pp. 913–916.
[46] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[47] B. E. Dom, "An information-theoretic external cluster-validity measure," in *Proc. UAI*, Edmonton, AB, Canada, 2002, pp. 137–145.
[48] B. Flury, *A First Course in Multivariate Statistics*. New York, NY, USA: Springer, 2013.
[49] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. CVPR*, Miami, FL, USA, 2009, pp. 951–958.
[50] C. Ansótegui, M. Sellmann, and K. Tierney, "A gender-based genetic algorithm for the automatic configuration of algorithms," in *Proc. CP*, Lisbon, Portugal, 2009, pp. 142–157.
[51] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.
[52] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 16, no. 1, pp. 122–128, Jan. 1986.
[53] K. G. Srinivasa, K. R. Venugopal, and L. M. Patnaik, "A self-adaptive migration model genetic algorithm for data mining applications," *Inf. Sci.*, vol. 177, no. 20, pp. 4295–4313, 2007.

Faliang Huang received the Ph.D. degree in data mining from the South China University of Technology, Guangzhou, China.

He is currently an Associate Professor with the Department of Software Engineering, Fujian Normal University, Fuzhou, China. His current research interests include data mining and natural computing.

Xuelong Li (M'02–SM'07–F'12) received the Ph.D. degree from the University of Science and Technology of China, Hefei, China.

He is a Full Professor with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

Shichao Zhang (SM'05) received the Ph.D. degree in computer science from Deakin University, Geelong, VIC, Australia.

He is currently a China 1000-Plan Distinguished Professor with the Department of Computer Science, Guangxi Normal University, Guilin, China. His current research interests include information quality and pattern discovery.

Jilian Zhang received the Ph.D. degree in information systems from Singapore Management University, Singapore.

His current research interests include data management, query authentication for outsourced databases, data privacy, and data mining.