A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering

Rui Xu, *Member, IEEE*, Jie Xu, *Member, IEEE*, and Donald C. Wunsch, II, *Fellow, IEEE*

***Abstract*—Swarm intelligence has emerged as a worthwhile class of clustering methods due to its convenient implementation, parallel capability, ability to avoid local minima, and other advantages. In such applications, clustering validity indices usually operate as fitness functions to evaluate the qualities of the obtained clusters. However, as the validity indices are usually data dependent and are designed to address certain types of data, the selection of different indices as the fitness functions may critically affect cluster quality. Here, we compare the performances of eight well-known and widely used clustering validity indices, namely, the Caliński–Harabasz index, the CS index, the Davies–Bouldin index, the Dunn index with two of its generalized versions, the I index, and the silhouette statistic index, on both synthetic and real data sets in the framework of differential-evolution–particle-swarm-optimization (DEPSO)-based clustering. DEPSO is a hybrid evolutionary algorithm of the stochastic optimization approach (differential evolution) and the swarm intelligence method (particle swarm optimization) that further increases the search capability and achieves higher flexibility in exploring the problem space. According to the experimental results, we find that the silhouette statistic index stands out in most of the data sets that we examined. Meanwhile, we suggest that users reach their conclusions not just based on only one index, but after considering the results of several indices to achieve reliable clustering structures.**

***Index Terms*—Clustering, differential evolution (DE), particle swarm optimization (PSO), swarm intelligence, validity index.**

I. Introduction

AS ONE OF the most important and primitive activities of human beings, cluster analysis, also known as unsupervised classification or exploratory data analysis, attempts to explore the unknown nature of data by separating a finite data set, with little or no ground truth, into a finite and discrete set of "natural" hidden data structures [1]–[5]. In particular, partitional clustering directly partitions data objects into some prespecified number of clusters, which can be stated mathematically as follows [6].

Manuscript received July 5, 2011; revised November 23, 2011; accepted January 25, 2012. Date of publication March 15, 2012; date of current version July 13, 2012. This work was supported in part by the National Science Foundation, by the Missouri University of Science and Technology Intelligent Systems Center, and by the Mary K. Finley Missouri endowment. This paper was recommended by Editor E. Santos, Jr.

R. Xu is with the Machine Learning Laboratory, GE Global Research, Niskayuna, NY 12309 USA (e-mail: rxu@ieee.org).

J. Xu is with the Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA 22030 USA (e-mail: jxu13@gmu.edu).

D. C. Wunsch, II, is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: dwunsch@mst.edu).

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/TSMCB.2012.2188509

Given a set of N data objects $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_j, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jd}) \in \Re^d$, find a K-partition of $\mathbf{X}$, $C = \{C_1, \ldots, C_K\}$ $(K \leq N)$, such that

$$C_i \neq \phi, \qquad i = 1, \ldots, K \tag{1}$$

$$\bigcup_{i=1}^{K} C_i = \mathbf{X} \tag{2}$$

$$C_i \cap C_j = \phi, \qquad i, j = 1, \ldots, K \text{ and } i \neq j. \tag{3}$$

This problem can be solved by optimizing a predefined criterion function so that data objects are more similar to others in the same cluster than to those in different clusters. Although, in principle, such an optimal partition can be found by enumerating all possibilities, this brute force method is infeasible in practice due to the extremely expensive computation involved. A natural alternative is to use a category of heuristic algorithms, e.g., the K-means algorithm [7], [8], to seek approximate optimal solutions that have a reasonable computational burden. However, such hill-climbing-based clustering algorithms suffer from inherent disadvantages; they are sensitive to the initial partitions and are easily stuck in local minima [9]. For these reasons, more powerful search techniques, such as evolutionary algorithms [10]–[12], differential evolution (DE) [13], [14], and simulated annealing [15], which belong to the category of stochastic optimization methods, and deterministic annealing [16], one of the most typical deterministic search techniques, potentially offer a more effective way to explore the clustering solution space.

In recent years, clustering based on a family of nature-inspired algorithms, including particle swarm optimization (PSO) [17], [18] and ant algorithms [19] under the domain of swarm intelligence [18], has increasingly attracted attention, popularity, and effort from a wide variety of research fields because of its ease of implementation and demonstrated effectiveness in solving complicated combinatorial optimization problems [20]. In the following discussions, we will focus on PSO-based clustering; discussions regarding clustering with ant algorithms can be found in [21]–[23]. Merwe and Engelbrecht used PSO to seek a prespecified number of cluster centroids and also suggested using the results of K-means to initialize one of the particles [24]. Cui *et al.* also considered the combination of PSO and K-means in the context of document clustering, utilizing the clustering results of PSO as the initial seeds to refine the final partition [25]. Implementation of PSO in a variant of K-means, called the K-harmonic-means algorithm, and in the fuzzy c-means algorithm was presented in [26] and [27], respectively. Moreover, in order to alleviate dependence on the user-specified number of clusters, Omran *et al.* defined each

particle as a binary vector whose length equals the maximum number of clusters, which is supposed to be much easier to estimate [28]. Each bit in the particle indicates whether the corresponding cluster centroid is included in the clustering solution. Das *et al.* further designed a PSO particle as an integration of an activation threshold and a set of cluster centroids, thus improving upon the aforementioned method's limitation of taking only one set of cluster centroids into account during each iteration [29].

In such PSO-based clustering applications, as well as in other evolutionary-algorithm-based clustering practices [30]–[32], clustering validity indices usually operate as fitness functions to evaluate the clusters' quality. However, as the validity indices are usually designed to address certain types of data, the selection of different indices may lead to critically different clusters. Based on this consideration, the objective of this paper is to help select appropriate validity indices in swarm-intelligence-based clustering, which could also be used as a reference for other types of evolutionary-algorithm-based clustering, through the investigation and comparison of the performances of eight widely used clustering validity indices. These are the Caliński-Harabasz (CH) index [33], the CS index [34], the Davies-Bouldin (DB) index [35], the Dunn index [36] with two of its generalized versions (GD_{33} and GD_{53}) [37], the I index [30], and the silhouette statistic (SIL) index [38]. These are tested on both synthetic and real data sets in the framework of differential-evolution-particle-swarm-optimization (DEPSO)-based clustering [39].

DE is a simple stochastic function minimizer that evolves individuals in order to increase the convergence to optimal solutions [13], [14]. DEPSO combines the DE operator into the PSO procedure in order to diversify the PSO, thus keeping the particles from falling into a local minimum [40]. Applications of DEPSO in recurrent neural network training [41] and combinatorial logic circuits [42] demonstrate its superior performance to either DE or PSO individually in terms of convergence speed and solution quality. Our previous empirical results, reported in a shorter conference version of this paper [39], also show that the DEPSO-based clustering algorithm achieves better performance than either PSO or DE in terms of the number of epochs required to reach a prespecified cutoff value of the fitness function.

This paper is organized as follows. Section II describes the DEPSO-based clustering algorithms. In Section III, the eight clustering validity indices are summarized. Section IV introduces both the synthetic and real data sets used in the analysis and details the experimental results. Section V concludes this paper.

II. DEPSO-Based Clustering

A. PSO

PSO is a population-based search strategy consisting of a swarm of particles, each of which represents a candidate solution [17], [18], [43]. Each particle i with a position represented as $\mathbf{z}_i$ moves in the multidimensional problem space with a corresponding velocity $\mathbf{v}_i$. The basic idea of PSO is that each particle randomly searches through the problem space by updating itself with its own memory and the social information gathered from other particles. These components are represented in terms of the two best locations during the evolution process; one is the particle's own previous best position, recorded as vector $\mathbf{p}_i$, according to the calculated fitness value, which is measured in terms of clustering validity indices in the context of clustering, and the other is the best position in the entire swarm, represented as $\mathbf{p}_g$. Also, $\mathbf{p}_g$ can be replaced with a local best solution obtained within a certain local topological neighborhood.

The canonical PSO velocity and position equations at iteration t are written as

$$\mathbf{v}_i(t) = w \times \mathbf{v}_i(t-1) + c_1 \times \varphi_1 \times (\mathbf{p}_i - \mathbf{z}_i(t-1)) + c_2 \times \varphi_2 \times (\mathbf{p}_g - \mathbf{z}_i(t-1)) \quad (4)$$

$$\mathbf{z}_i(t) = \mathbf{z}_i(t-1) + \mathbf{v}_i(t) \quad (5)$$

where $\mathbf{z}_i(t-1)$ and $\mathbf{v}_i(t-1)$ denote the particle's position and the associated velocity vector in the search space at generation $t-1$, w is the inertia weight, c_1 and c_2 represent the acceleration constants, and φ_1 and φ_2 are uniform random functions in the range of [0, 1].

PSO displays many desirable characteristics, such as flexibility in balancing global and local searches, computational efficiency for both time and memory, and ease of implementation. Also, the memory mechanism implemented in PSO can retain the information about previous best solutions that may get lost during the population's evolution. As to parameter tuning, an important problem hindering many computational approaches, PSO only needs to determine four major parameters in advance using some useful rules. The inertia weight w is designed as a tradeoff between the global and local searches. Larger values of w facilitate global exploration, while lower values encourage a local search. w can be fixed to some certain value or can vary with a random component, such as

$$w = w_{\max} - \varphi_3/2 \quad (6)$$

where φ_3 is a uniform random function in the range of [0, 1] and $w_{\max}$ is a constant. For example, if $w_{\max}$ is set to one, (6) makes w vary between 0.5 and 1, with a mean of 0.75. c_1 and c_2 are known as the cognitive and social components, respectively, and are used to adjust the velocity of a particle toward $\mathbf{p}_i$ and $\mathbf{p}_g$. A set of commonly used combinations of c_1 and c_2 based on past experience can be found in [17], [18], and [41]. During the evolutionary procedure, the velocity for each particle is restricted to a limit $V_{\max}$, as in velocity initialization. When the velocity exceeds $V_{\max}$, it is reassigned to $V_{\max}$. If $V_{\max}$ is too small, particles may become trapped in local optima; if it is too large, particles may miss some good solutions. $V_{\max}$ is usually set to around 10%–20% of the dynamic range of the variable on each dimension [18].

B. DE

DE primarily involves generating trial parameter vectors by adding a weighted difference of two parameters to a third one. In this way, no separate probability distribution is required, which makes the scheme completely self-organizing [13], [14].

In every generation of DE, for each individual $\mathbf{z}_1$ in the population, three other distinct individuals $\mathbf{z}_2$, $\mathbf{z}_3$, and $\mathbf{z}_4$ are randomly selected. The difference vector between $\mathbf{z}_2$ and $\mathbf{z}_3$ is added to $\mathbf{z}_4$ in order to generate an offspring $\mathbf{o}_1$ for $\mathbf{z}_1$. Of the parent $(\mathbf{z}_1)$ and the offspring $(\mathbf{o}_1)$, the one with greater fitness is selected to be part of the next generation of individuals. The aforementioned DE operation can be summarized by means of

$$o_{1j} = \begin{cases} z_{4j} + \gamma(z_{2j} - z_{3j}), & \text{if } \varphi \leq p_r \text{ or } j = r \\ z_{1j}, & \text{otherwise} \end{cases} \quad (7)$$

where j corresponds to the dimension of the individual that is to be mutated, r is a random integer within one and the dimension of the problem space, $\gamma \in [0, 2]$ is a scaling factor that can be selected based on the application as a constant or that can vary randomly in some certain range, as w in PSO, φ is the uniform random function in the range of [0, 1], and p_r is the crossover rate that controls the occurrence of the mutation procedure for each dimension in the individual. p_r could be linearly decreased from its maximum value p_{rmax} (usually 1.0) to some minimum value p_{rmin}. Thus, in the beginning of the optimizing process, all dimensions of the parent vector are replaced in the offspring using the difference vector operator, while at later stages of the process, the offspring will likely inherit more elements of the parent.

The DE strategy described earlier is denoted as DE/rand/1/bin, based on the notation DE/x/y/z, where x represents the vector to be mutated (a random vector or the best vector), y indicates the number of difference vectors used, and z specifies the crossover scheme (binomial or exponential) [13]. Some of the more commonly used strategies include DE/rand/2/bin, DE/best/1/bin, and DE/best/2/bin.

C. Hybrid DE and PSO

We have previously achieved good results combining DE with PSO in [41]; see also [42]. Next, we briefly explain this hybrid approach.

In PSO, the search process is based on the social and cognitive components $\mathbf{p}_g$ and $\mathbf{p}_i$. The entire swarm tries to follow $\mathbf{p}_g$, thus improving its own position. However, (4) reveals that the new velocity of that particular $\mathbf{p}_g$ particle depends solely on the weighted old velocity. The goal, then, of the hybrid of DE and PSO, i.e., DEPSO, is to add diversity to the PSO and keep the particles from being stuck to local minima [40].

The DEPSO algorithm involves a two-step process. In the first step, the canonical PSO, as described in Section II-A, is implemented. In the second step, the DE mutation operator is applied to the particles. The crossover rate for this study is one [40]. Therefore, for every odd iteration, the canonical PSO algorithm is carried out, while for every even iteration, the DE operator is applied. The following steps summarize the procedure for implementing DEPSO.

1) Initialize a population of particles with random positions and velocities. Set the values of user-dependent parameters.
2) For every odd iteration, carry out the canonical PSO operation on each individual in the population.
   a) Calculate the fitness function $Fit(\mathbf{z}_i)$ for each particle $\mathbf{z}_i$.
   b) Compare the fitness value of each particle $Fit(\mathbf{z}_i)$ with $Fit(\mathbf{p}_i)$. If the current value is better, reset both $Fit(\mathbf{p}_i)$ and $\mathbf{p}_i$ to the current value and location.
   c) Compare the fitness value of each particle $Fit(\mathbf{z}_i)$ with $Fit(\mathbf{p}_g)$. If the current value is better, reset $Fit(\mathbf{p}_g)$ and $\mathbf{p}_g$ to the current value and location.
   d) Update the velocity and position of the particles based on (4) and (5).
3) For every even iteration, carry out the following steps.
   a) For every particle $\mathbf{z}_i$ with its personal best $\mathbf{p}_i$, randomly select four particles $\mathbf{z}_a$, $\mathbf{z}_b$, $\mathbf{z}_c$, and $\mathbf{z}_d$ that are different from $\mathbf{z}_i$, and calculate $\boldsymbol{\Delta}_1$ and $\boldsymbol{\Delta}_2$ as

$$\boldsymbol{\Delta}_1 = \mathbf{p}_a - \mathbf{p}_b, \qquad a \neq b \quad (8)$$

$$\boldsymbol{\Delta}_2 = \mathbf{p}_c - \mathbf{p}_d, \qquad c \neq d \quad (9)$$

   where $\mathbf{p}_a$, $\mathbf{p}_b$, $\mathbf{p}_c$, and $\mathbf{p}_d$ are the corresponding best solutions of the four selected particles.
   b) Calculate the mutation value $\boldsymbol{\delta}_i$ using (10), and create the offspring $\mathbf{o}_i$ using (11)

$$\boldsymbol{\delta}_i = (\boldsymbol{\Delta}_1 + \boldsymbol{\Delta}_2)/2 \quad (10)$$

$$o_{ij} = p_{ij} + \delta_{ij}, \qquad \text{if } \varphi \leq p_r \text{ or } j = r \quad (11)$$

   where φ, j, and r have the same definitions as in (7).
   c) Once the new population of offspring is created using steps a and b, their fitness is evaluated against that of the parent. The one with the best fitness is selected to participate in the next generation.
   d) Recalculate the $\mathbf{p}_g$ and $\mathbf{p}_i$ of the new population.
4) Repeat steps 2 and 3 until a stopping criterion is met, which usually occurs upon reaching the maximum number of iterations or discovering high-quality solutions.

D. DEPSO-Based Clustering

One of the major design decisions in implementing DEPSO for clustering is particle encoding. In the context of clustering N data objects into K clusters with cluster centroids $\mathbf{M} = \{\mathbf{m}_i, i = 1, \ldots, K\}$, the particle could be directly encoded to consist of the K cluster centroids, denoted as $\mathbf{z}_i = (\mathbf{m}_1, \ldots, \mathbf{m}_K)$, which is known as centroid-based representation [Fig. 1(a)] [24], [26], [27]. Other particle representations are also available, such as partition-based representation, which considers each individual in the swarm as a string of N data objects and in which the ith element of the string indicates the cluster number assigned to the ith data object [44]. For example, the representation of the clustering of nine data objects into three clusters $\{\mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_8\}$, $\{\mathbf{x}_3, \mathbf{x}_7\}$, and $\{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_6, \mathbf{x}_9\}$ is denoted as a string "132313213." However, in this case, the basic PSO, which is introduced in the following, must be modified in order to address the situation in which candidate solutions are integer vectors. Meanwhile, redundancy and context insensitivity are also problems requiring extra attention. These problems manifest as the replication of the same clustering but

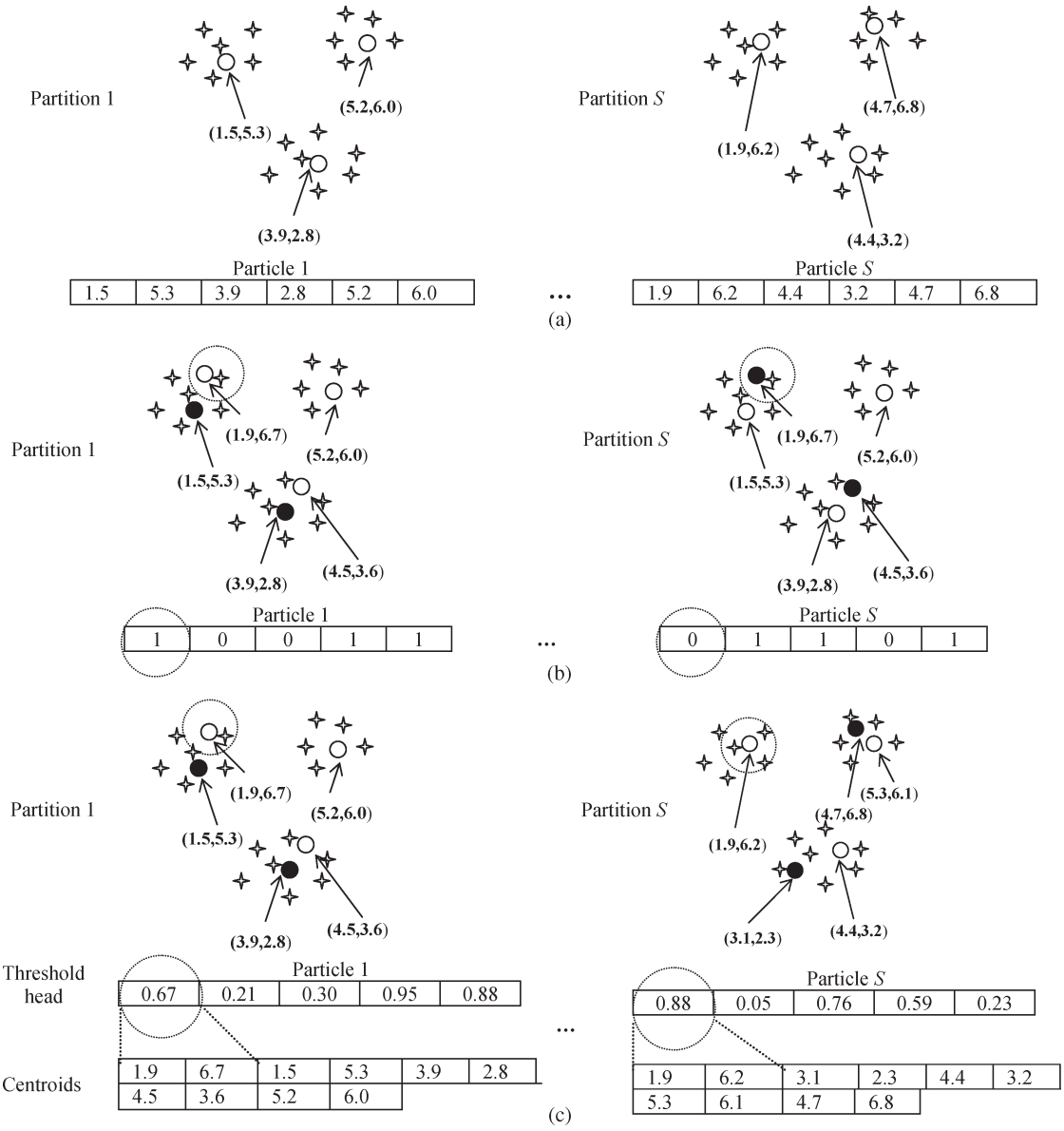

Fig. 1. Three types of particle representations in a 2-D space. Each particle could be encoded as (a) a sequence of cluster centroids, (b) a binary sequence indicating the selection of cluster centroids (1 for selected centroids represented in hollow circles and 0 for unselected centroids represented in solid circles; data samples are represented by a star-shaped symbol), or (c) an activation threshold head with a sequence of cluster centroids (if the threshold is greater than 0.5, cluster centroids are selected; otherwise, the centroids are not selected).

with different labels attached to the clusters. Thus, the string "321232132" also represents the same clustering solution as in the previous example. Additional solution parameters for cluster labels and group-oriented evolutionary operators that work with groups of genes are suggested to prevent such a cluster-label-associated cycle [45], [46].

The major disadvantage associated with centroid-based representation is the necessity of determining the number of clusters in advance, which is undesirable in some applications. Omran *et al.* addressed this problem by considering each particle as a binary vector, with a length equal to the maximum number of clusters $K_{\max}$, i.e., $\mathbf{z}_i = (z_{i1}, \ldots, z_{iK\max})$, with $z_{ij} \in \{0, 1\}$ for $j = 1, \ldots, K_{\max}$, as, generally, it is much easier to estimate the maximum number of clusters than the actual number of clusters [28]. Thus, if $z_{ij} = 1$, the corresponding cluster centroid is selected to be part of the clustering solution. On the other hand, if $z_{ij} = 0$, the corresponding cluster centroid is excluded from the current solution [Fig. 1(b)]. Note that, in this encoding strategy, only one set of cluster centroids is taken into account during each iteration.

Here, we adopt the particle representation proposed by Das *et al.* [29], which extends the aforementioned binary particle representation by considering $K_{\max}$ sets of cluster centroids at a time, rather than only one. Given the maximum number of clusters $K_{\max}$ and a set of d-dimensional cluster centroids $\{\mathbf{m}_{ij}, j = 1, \ldots, K_{\max}\}$, the particle i is encoded as a $K_{\max} + K_{\max} \times d$ vector $\mathbf{z}_i = (\tau_{i1}, \ldots, \tau_{iK\max}, \mathbf{m}_{i1}, \ldots, \mathbf{m}_{iK\max})$, where τ_{ij} $(j = 1, \ldots, K_{\max})$ is the activation threshold in the range of [0, 1]. The activation threshold functions as a control parameter that determines whether the corresponding cluster centroid is selected. If it is greater than 0.5, the cluster is chosen; otherwise, the cluster becomes inactive [Fig. 1(c)].

For example, assume that the vector (0.67 0.21 0.30 0.95 0.88 (1.9 6.7) (1.5 5.3) (3.9 2.8) (4.5 3.6) (5.2 6.0)) is an instance of a particle encoding a clustering partition in a 2-D data space with a maximum of five clusters. The activation thresholds 0.67, 0.95, and 0.88 then indicate that the first, fourth, and fifth clusters, with centroids (1.9 6.7), (4.5 3.6), and (5.2 6.0), become activated, and data objects are assigned to these three clusters. The two other clusters with corresponding activation thresholds less than 0.5 (0.21 and 0.3) will not be considered. If the activation thresholds become negative or greater than one, they are fixed to zero or one, respectively. If all activation thresholds for a particular particle are less than or equal to 0.5, which indicates that no clusters are activated, two thresholds will be selected randomly and reinitialized to a random value in the range of 0.5–1 to ensure that at least two clusters exist. Also, if an empty cluster exists, which may occur as a result of placing the cluster centroid too far from any boundaries of the distributions of the data set, we deactivate the cluster by randomly setting its corresponding activation threshold in the range of [0, 0.5]. We then check that the minimum number of clusters (2) is satisfied.

With this encoding strategy, the DEPSO-based clustering algorithm follows the steps summarized in the previous section while using some specific clustering validity index as the fitness function to evaluate the clustering quality. Eight of the most commonly used and well-known indices that are used in this study will be reviewed in the next section.

III. Clustering Validity Indices

Cluster validity is concerned with the objective and quantitative evaluation of the derived clusters or whether the disclosed clustering structure is meaningful [4], [47]. Given a data set, each clustering algorithm can always produce a partition, regardless of the actual existence of any particular structure in the data. Moreover, different clustering algorithms usually lead to different clusters of data; even for the same algorithm, the selection of different parameters or the presentation order of data objects may greatly affect the final clustering partitions. Thus, effective evaluation standards and criteria are critically important to equip users with a degree of confidence regarding the clustering results. At the same time, these assessments also provide some meaningful insights into questions about how many clusters are hidden in the data, which is called "the fundamental problem of cluster validity" [48] and includes questions regarding whether the clusters obtained are meaningful from a practical point of view or just artifacts of the algorithms, and why we prefer one algorithm over another for a specific problem.

As one important approach to assessing cluster quality, clustering validity indices combine information about intracluster compactness and intercluster isolation, as well as other factors, such as the defined squared error, the geometric or statistical properties of the data, the number of data objects, the dissimilarity or similarity measurement, and the number of clusters. Rich cluster validity indices with different properties and characteristics exist in the literature [49]–[51]. Bandyopadhyay and Maulik compared five validity indices for variable-string-length genetic-algorithm-based clustering [30], and Das *et al.* implemented two validity indices for DE-based clustering [32]. Sheng *et al.* also considered six different validity indices, together with the weighted sum of all of their validity functions [52]. Here, we focus on eight of these indices that are widely used as fitness functions in evolutionary algorithms or swarm-intelligence-based clustering.

A. CH Index

The CH index [33] achieved the best performance among 30 indices in Milligan and Cooper's comparison study [51]. Given a set of N data objects $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ assigned to K clusters $C = \{C_1, \ldots, C_K\}$ with centroids $\mathbf{m}_i$, with $i = 1, \ldots, K$, the CH index is defined as

$$CH(K) = \frac{Tr(\mathbf{S}_B)}{K-1} \Big/ \frac{Tr(\mathbf{S}_W)}{N-K} \tag{12}$$

where $Tr(\mathbf{S}_B) = \sum_{i=1}^{K} N_i \|\mathbf{m}_i - \mathbf{m}\|^2$ and $Tr(\mathbf{S}_W) = \sum_{i=1}^{K} \sum_{j=1}^{N_i} \|\mathbf{x}_j - \mathbf{m}_i\|^2$ are the traces of the between- and within-cluster scatter matrices, respectively. Here, N_i represents the number of data objects belonging to cluster C_i, and $\mathbf{m}$ is the total mean vector for the entire data set. A large value of $CH(K)$ suggests a clustering result with good quality.

B. DB Index

The DB index [35] attempts to maximize the between-cluster distance while minimizing the distance between the cluster centroid and the other data objects. By defining each individual cluster index R_i as the maximum comparison between cluster C_i and other clusters in the partition

$$R_i = \max_{j, j \neq i} \left(\frac{e_i + e_j}{D_{ij}} \right) \tag{13}$$

where $D_{ij} = \|\mathbf{m}_i - \mathbf{m}_j\|^2$ is the distance between the centroids of clusters C_i and C_j and e_i and e_j are the average errors for clusters C_i and C_j, respectively, given as $e_i = (1/N_i) \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{m}_i\|^2$, the DB index can be written as

$$DB(K) = \frac{1}{K} \sum_{i=1}^{K} R_i. \tag{14}$$

The minimum value of $DB(K)$ indicates the occurrence of the best clustering partition.

C. Dunn and Dunn-like Indices

The Dunn index [36] is designed to identify clusters that are compact and well separated. Defining the distance $D(C_i, C_j)$ between two clusters C_i and C_j as the minimum distance between a pair of data objects $\mathbf{x}$ and $\mathbf{y}$ belonging to C_i and C_j, respectively

$$D(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} D(\mathbf{x}, \mathbf{y}) \tag{15}$$

and the diameter $\delta(C_i)$ of cluster C_i as the maximum distance between two of its members

$$\delta(C_i) = \max_{\mathbf{x},\mathbf{y}\in C_i} D(\mathbf{x},\mathbf{y}) \tag{16}$$

the Dunn index is constructed as

$$Du(K) = \min_{i=1,\ldots,K}\left(\min_{\substack{j=1,\ldots,K\\ j\neq i}}\left(\frac{D(C_i,C_j)}{\max_{l=1,\ldots,K}\delta(C_l)}\right)\right). \tag{17}$$

A large value of $Du(K)$ suggests the presence of compact and well-separated clusters.

The Dunn index has the disadvantage of oversensitivity to noise, for which a family of 18 cluster validity indices is proposed based on the different definitions of cluster distance and cluster diameter [37]. Among them, we use two generalized Dunn indices GD_{33} and GD_{53}, hereby following the original subscript given to the distance function and cluster diameter by Bezdek and Pal [37], which achieve better performance in that empirical study. Both indices are based on the same definition of the cluster diameter $\delta_3(C_i)$, which is more robust to noise

$$\delta_3(C_i) = 2\frac{\sum_{\mathbf{x}\in C_i} D(\mathbf{x},\mathbf{m}_i)}{N_{i_i}} \tag{18}$$

but with different distance function definitions

$$D_3(C_i,C_j) = \frac{1}{N_iN_j}\sum_{\substack{\mathbf{x}\in C_i\\ \mathbf{y}\in C_j}} D(\mathbf{x},\mathbf{y}) \tag{19}$$

$$D_5(C_i,C_j) = \frac{1}{N_i+N_j}\left(\sum_{\mathbf{x}\in C_i} D(\mathbf{x},\mathbf{m}_j) + \sum_{\mathbf{y}\in C_j} D(\mathbf{y},\mathbf{m}_i)\right). \tag{20}$$

The former corresponds to the concept of average linkage used in hierarchical clustering, while the latter is regarded as a set distance integrating both the averaging concept and the cluster centroids. As in the Dunn index, we select the appropriate clustering partitions corresponding to large values of the two generalized Dunn indices.

D. CS Index

The CS index carries the same basic rationale as that introduced by the DB and Dunn indices in the previous sections, but it integrates and modifies the two indices with a focus on dealing with clusters of different densities and/or sizes [34]

$$CS(K) = \frac{\sum_{i=1}^{K}\left(\frac{1}{N_i}\sum_{\mathbf{x}_j\in C_i}\max_{\mathbf{x}_l\in C_i} D(\mathbf{x}_l,\mathbf{x}_j)\right)}{\sum_{i=1}^{K}\left(\max_{j\in K, j\neq i} D(\mathbf{m}_i,\mathbf{m}_j)\right)}. \tag{21}$$

Good clustering results correspond to small values of $CS(K)$.

E. I Index

The I index consists of three factors that compete with and balance each other, defined as in [30]

$$I(K) = \left(\frac{1}{K}\times\frac{E_1}{E_K}\times\max_{i,j=1,\ldots,K}\|\mathbf{m}_i-\mathbf{m}_j\|\right)^p \tag{22}$$

where $p\geq 1$ is any real number and

$$E_K = \sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i}\|\mathbf{x}-\mathbf{m}_i\| \tag{23}$$

which also shows that E_1 is a constant for a given data set. Upon further examination of (22), it becomes obvious that the first factor $1/K$ decreases linearly as K increases, which is complementary to the other two factors that tend to increase as K increases. Clustering solutions with large values of the I index are preferred.

F. SIL Index

The average distance of a data object $\mathbf{x}_j$ belonging to a cluster C_i from all other data objects in C_i is calculated as $a_j = (1/N_i - 1)\sum_{\substack{l=1,\ldots,N_i\\ l\neq j}}\|\mathbf{x}_j-\mathbf{x}_l\|$. For other clusters C_h, with $h=1,\ldots,K$ and $h\neq i$, we select the smallest average distance of $\mathbf{x}_j$ to all objects in C_h, denoted as $b_j = \min_{\substack{h=1,\ldots,K\\ h\neq i}}(1/N_h)\sum_{\mathbf{x}_l\in C_h}\|\mathbf{x}_j-\mathbf{x}_l\|$. We then obtain the silhouette width of $\mathbf{x}_j$, which measures how well a data object is clustered, as

$$s_j = \frac{b_j-a_j}{\max(a_j,b_j)}. \tag{24}$$

Thus, positive and negative large silhouette widths indicate that the corresponding object is well clustered and wrongly clustered, respectively. Any objects with silhouette widths around zero are considered not to be clearly discriminated between clusters. The SIL index is defined as the average of the silhouette width s_j over all data objects [38]

$$SIL = \frac{1}{N}\sum_{j=1}^{N} s_j. \tag{25}$$

Immediately following the previous analysis, large SIL values indicate good cluster partitions.

IV. Empirical Results

A. Data Sets

The DEPSO-based clustering algorithm, using different cluster validity indices as the fitness functions, is applied to the following six synthetic and four real data sets.

TABLE I
DESCRIPTION OF THE SYNTHETIC DATA SETS

Data Set	Number of Clusters	Dimensionality	Number of Data Points
2d4c	4	2	1078
2d10c	10	2	3073
2d20c	20	2	1517
10d20c	20	10	1316
50d4c	4	50	683
100d4c	4	100	629

The synthetic data used in this study are generated from two generators developed by Handl and Knowles [53] and are available at http://dbkgroup.org/handl/generators/. The first generator creates clusters based on a standard cluster model using multivariate Gaussian distributions, while the second generator provides more elongated cluster shapes in arbitrarily high dimensions. Among these data sets, the dimensions vary from 2, providing good visualization analysis, to medium values, such as 10 and 50, to a much higher value at 100, in order to examine the performance of the algorithm in high-dimension spaces. The number of clusters varies in the range of 4, 10, and 20, with different degrees of overlap, to reflect the different levels of clustering difficulty. We name these data sets based on their dimensions and number of clusters. For example, data set 100d4c indicates that the 100-dimension data set contains four clusters. The six data sets, four (2d4c, 2d10c, 2d20c, and 10d20c) from the first generator and two (50d4c and 100d4c) from the second generator, are summarized in Table I in terms of their number of clusters, dimensions, and number of data objects.

The four real data sets that we considered are the iris, wine, Wisconsin breast cancer, and small round blue cell tumor (SRBCT) data sets. The first three data sets can be downloaded from the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml/index.html [54]. These data sets are among the most popular for comparing and examining the performances of algorithms in the fields of pattern recognition and machine learning. Specifically, the iris data set consists of three categories, i.e., iris setosa, iris versicolor, and iris virginica, with each species having 50 samples represented in terms of four features (sepal length, sepal width, petal length, and petal width). Iris setosa can be linearly separated from iris versicolor and iris virginica, but iris versicolor and iris virginica overlap greatly. The wine data set was collected as a result of a chemical analysis of wines grown in the same region but derived from three different cultivars. The three categories consist of 59, 71, and 48 samples, respectively. Thirteen features are considered for each sample, including alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavonoids, nonflavonoid phenols, proanthocyanidins, color intensity, hue, OD280/OD315 of diluted wines, and proline. The Wisconsin breast cancer data set originally includes 699 samples, both benign and malignant, but we removed the 16 samples that have missing features, resulting in 683 total samples in our analysis, with 239 normal tissues and 444 cancer samples. Each sample contains nine features corresponding to clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. The SRBCT data set comes from diagnostic research of the SRBCTs of childhood, which consists of 83 samples from four categories, known as the Burkitt lymphomas, the Ewing family of tumors, neuroblastoma, and rhabdomyosarcoma [55]. Gene expression levels of 6567 genes were measured using a complementary DNA microarray for each sample. Of these samples, 2308 passed the filter that requires the red intensity of a gene to be greater than 20 and were kept for further analyses. The relative red intensity (RRI) of a gene is defined as the ratio between the mean intensity of that particular spot and the mean intensity of all filtered genes. The ultimate expression level measure is the natural logarithm of RRI.

B. Experimental Design

Because the real partitions of the data sets used here are already known, the performances of DEPSO-based clustering based on the different clustering validity indices can then be evaluated by comparing the resulting clusters with the real structures in terms of external criteria. Some commonly used criteria include the Rand index, the Jaccard coefficient, and the Fowlkes–Mallows index [2], [4].

Assuming that $\mathbf{P}$ is a prespecified partition of data set $\mathbf{X}$ with N data objects, which is also independent from a clustering structure $\mathbf{C}$ resulting from the use of the DEPSO-based clustering algorithm, a pair of data objects $\mathbf{x}_i$ and $\mathbf{x}_j$ will yield four different cases based on how $\mathbf{x}_i$ and $\mathbf{x}_j$ are placed in $\mathbf{C}$ and $\mathbf{P}$.

- Case 1) $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same clusters of $\mathbf{C}$ and the same category of $\mathbf{P}$.
- Case 2) $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same clusters of $\mathbf{C}$ but different categories of $\mathbf{P}$.
- Case 3) $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different clusters of $\mathbf{C}$ but the same category of $\mathbf{P}$.
- Case 4) $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different clusters of $\mathbf{C}$ and different categories of $\mathbf{P}$.

Correspondingly, the numbers of pairs of samples for the four cases are denoted as a, b, c, and d, respectively. Because the total number of pairs of samples is $N(N-1)/2$, denoted as M, we have $a+b+c+d=M$. The three external criteria used in our analysis can then be defined as follows, with larger values indicating a greater similarity of $\mathbf{C}$ and $\mathbf{P}$:

1) Rand index

$$R = (a+d)/M \tag{26}$$

2) Jaccard coefficient

$$J = a/(a+b+c) \tag{27}$$

3) Fowlkes-Mallows index

$$FM = \sqrt{\frac{a}{a+b}\frac{a}{a+c}}. \tag{28}$$

In order to correct an index for randomness, we want to normalize the index, denoted as Ind, so that its value is zero

TABLE II
COMPARISON OF PERFORMANCES OF THE DEPSO-BASED CLUSTERING ALGORITHM WITH DIFFERENT COGNITIVE (c_1) AND SOCIAL (c_2) COMPONENT SETTINGS BASED ON A ONE-TAILED t-TEST. THE ANALYSIS IS BASED ON 100 RUNS WITH THE ADJUSTED RAND INDEX USED FOR VALIDATION, AND SIMILAR RESULTS CAN BE OBTAINED WITH OTHER INDICES. THE p- VALUES ARE SHOWN FOR DIFFERENT INDICES, AND A p-VALUE LESS THAN 0.05 INDICATES THAT THE MEAN WHEN $c_1 = 2.5$ AND $c_2 = 1.5$ IS GREATER THAN THE MEAN OF THE OTHER SETTINGS AT THE 5% SIGNIFICANCE LEVEL

Data Set	c_1=2.5 and c_2=1.5 vs. c_1=0.5 and c_2=0.5								c_1=2.5 and c_2=1.5 vs. c_1=0.5 and c_2=0.5							
	CH	CS	DB	DUNN	GD_{33}	GD_{53}	I	SIL	CH	CS	DB	DUNN	GD_{33}	GD_{53}	I	SIL
2d4c	0.542	1	1	$<10^{-21}$	$<10^{-70}$	$<10^{-27}$	1	1	0.510	1	1	$<10^{-28}$	$<10^{-61}$	$<10^{-30}$	1	1
100d4c	0.003	$<10^{-9}$	0.001	0.019	$<10^{-33}$	0.430	0.014	1	0.022	0.036	0.003	0.099	$<10^{-41}$	0.352	0.043	1
Iris	0.002	$<10^{-16}$	0.5	0.5	0.865	0.005	1	0.5	0.008	$<10^{-7}$	0.5	0.5	0.237	0.715	1	0.5
Wine	0.058	1	0.928	0.019	0.884	0.999	1	0.995	0.129	1	0.962	0.698	0.659	0.940	0.997	0.932

TABLE III
PERFORMANCES OF THE DEPSO-BASED CLUSTERING ALGORITHM, USING THE CH, CS, DB, $Dunn$, GD_{33}, GD_{53}, I, AND SIL INDICES AS THE FITNESS FUNCTIONS, IN TERMS OF THE RAND AND ADJUSTED RAND INDICES. GIVEN ARE THE MEAN AND STANDARD DEVIATION BASED ON 100 RUNS. THE BEST PERFORMANCE IN TERMS OF MEAN VALUES FOR EACH DATA SET IS HIGHLIGHTED IN BOLD

Data Set	Fitness Function															
	CH		CS		DB		Dunn		GD_{33}		GD_{53}		I		SIL	
	R	AR	R	AR	R	AR	R	AR	R	AR	R	AR	R	AR	R	AR
2d4c	0.9540 ± 0.0274	0.8899 ± 0.0697	0.7494 ± 0.0120	0.4987 ± 0.0205	0.9547 ± 0.0061	0.8967 ± 0.0137	**0.9915 ± 0.0094**	**0.9802 ± 0.0213**	0.9548 ± 0.0027	0.8968 ± 0.0064	0.9622 ± 0.0001	0.9138 ± 0.0003	0.9252 ± 0.0296	0.8102 ± 0.0778	0.9898 ± 0.0011	0.9760 ± 0.0026
2d10c	0.8463 ± 0.1180	0.5823 ± 0.2638	0.6605 ± 0.0958	0.2635 ± 0.1045	0.8874 ± 0.0893	0.6434 ± 0.1714	0.6326 ± 0.2151	0.3261 ± 0.2440	0.7130 ± 0.0038	0.2938 ± 0.0040	0.7052 ± 0.0030	0.2935 ± 0.0051	**0.9718 ± 0.0131**	**0.8683 ± 0.0506**	0.9534 ± 0.0056	0.8012 ± 0.0225
2d20c	0.9429 ± 0.1011	0.7723 ± 0.2697	0.7536 ± 0.0187	0.2323 ± 0.0164	0.7745 ± 0.0313	0.2556 ± 0.0387	0.9259 ± 0.0310	0.5565 ± 0.0993	0.6633 ± 0.0111	0.1524 ± 0.0090	0.6557 ± 0.0069	0.1527 ± 0.0020	0.9718 ± 0.0156	0.7745 ± 0.1018	**0.9776 ± 0.0091**	**0.8113 ± 0.0651**
10d20c	0.6911 ± 0.0402	0.1711 ± 0.0261	0.4382 ± 0.1372	0.0640 ± 0.0299	0.3459 ± 0.1545	0.0528 ± 0.0355	0.9626 ± 0.0204	0.7355 ± 0.1179	0.5893 ± 0.0628	0.1231 ± 0.0310	0.1422 ± 0.0344	0.0077 ± 0.0069	0.5577 ± 0.1917	0.1022 ± 0.0740	**0.9906 ± 0.0055**	**0.9198 ± 0.0423**
50d4c	0.7333 ± 0.0685	0.4665 ± 0.1370	0.6877 ± 0.0548	0.3750 ± 0.1096	0.6641 ± 0.0894	0.3277 ± 0.1790	**0.7831 ± 0.0597**	**0.5660 ± 0.1195**	0.6369 ± 0.0552	0.2737 ± 0.1105	0.6320 ± 0.0054	0.2635 ± 0.0108	0.7613 ± 0.0742	0.5226 ± 0.1484	0.7376 ± 0.1570	0.4749 ± 0.3144
100d4c	0.7979 ± 0.0580	0.5895 ± 0.1114	0.6145 ± 0.0351	0.2486 ± 0.0623	0.5963 ± 0.1285	0.2946 ± 0.1688	0.5768 ± 0.1201	0.2845 ± 0.1708	0.8223 ± 0.0095	0.6445 ± 0.0189	0.5212 ± 0.1057	0.1991 ± 0.1336	0.6721 ± 0.1141	0.3642 ± 0.1637	**0.8411 ± 0.0100**	**0.6664 ± 0.0194**
Iris	**0.8700 ± 0.0346**	**0.7121 ± 0.0714**	0.8364 ± 0.0396	0.6456 ± 0.0648	0.7763 ± 0.0000	0.5681 ± 0.0000	0.7763 ± 0.0000	0.5681 ± 0.0000	0.7563 ± 0.0008	0.5233 ± 0.0018	0.7755 ± 0.0017	0.5664 ± 0.0038	0.8025 ± 0.0342	0.5717 ± 0.0656	0.7763 ± 0.0000	0.5681 ± 0.0000
Wine	0.6898 ± 0.0112	0.3811 ± 0.0232	0.3911 ± 0.0098	0.0063 ± 0.0079	0.7458 ± 0.1110	0.4846 ± 0.1959	0.6893 ± 0.0027	0.4085 ± 0.0069	0.7843 ± 0.1054	0.5580 ± 0.1858	0.7045 ± 0.0826	0.4224 ± 0.1324	0.7631 ± 0.1032	0.5061 ± 0.1855	**0.9166 ± 0.1036**	**0.8259 ± 0.1507**
Wisconsin Breast Cancer	0.8896 ± 0.0003	0.7763 ± 0.0005	0.7861 ± 0.0395	0.5605 ± 0.0830	0.8861 ± 0.0345	0.7685 ± 0.0776	0.7689 ± 0.1395	0.5122 ± 0.3122	0.7462 ± 0.0067	0.4759 ± 0.0145	0.6096 ± 0.0648	0.1639 ± 0.1437	0.8999 ± 0.0383	0.7979 ± 0.0794	**0.9185 ± 0.0003**	**0.8354 ± 0.0005**
SRBCT	**0.5521 ± 0.0479**	**0.0793 ± 0.0707**	0.3344 ± 0.0345	-0.0071 ± 0.0121	0.2785 ± 0.0026	-0.0017 ± 0.0036	0.3014 ± 0.0110	-0.0026 ± 0.0026	0.2973 ± 0.0393	-0.0003 ± 0.0166	0.2872 ± 0.0143	-0.0007 ± 0.0091	0.3387 ± 0.0863	0.0093 ± 0.0326	0.2936 ± 0.0075	-0.0016 ± 0.0016

when two partitions are randomly selected and one when two partitions are perfectly matched [4]

$$Adj\_Ind = \frac{Ind - E(Ind)}{\max(Ind) - E(Ind)} \tag{29}$$

where $E(Ind)$ is the expected value of Ind under the baseline distribution and $\max(Ind)$ is the maximum value of Ind. Specifically, the adjusted Rand index by Hubert and Arabie [56] assumes that the model of randomness takes the form of the generalized hypergeometric distribution, which is written as

$$Adj\_R = \frac{\binom{N}{2}(a+d) - ((a+b)(a+c)+(c+d)(b+d))}{\binom{N}{2}^2 - ((a+b)(a+c)+(c+d)(b+d))}. \tag{30}$$

An adjusted Rand index value of one indicates a perfect agreement between two partitions, while zero suggests an agreement due to chance. Its value can also be negative, indicating an agreement less than chance. The adjusted Rand index has the advantage of a relatively simple form compared with other corrected indices, and it has demonstrated consistently good performance in previous studies [57], [58].

C. Experimental Results

In our experiments, we let the inertia weight w vary in the range of [0.5, 1], as aforementioned, and the crossover rate change by sampling from a Gaussian distribution $p_r \sim N(0.5, 0.15)$. Thirty particles are chosen. The major parameters that must be tuned are the cognitive and social components c_1 and c_2 of the DEPSO algorithm. Based on our previous empirical results [41] and some common recommendations in the literature [17], [43], we select three groups of settings ($c_1 = 2.5$ and $c_2 = 1.5$; $c_1 = 2.5$ and $c_2 = 0.5$; and $c_1 = 0.5$ and $c_2 = 0.5$) and compare their performances. Table II illustrates the results of the one-tailed t-test (at the 0.05 significance level) on four data sets when c_1 and c_2 are set at 2.5 and 1.5 versus the other two settings ($c_1 = 2.5$ and $c_2 = 0.5$, and $c_1 = 0.5$ and $c_2 = 0.5$). Based on the results, however, no settings stand out for all the data sets. For our further experiments, we set c_1 and c_2 at 2.5 and 1.5, respectively, and we will show, as demonstrated in Fig. 6, that the other two settings do not affect our conclusions regarding the performances of the eight different indices.

TABLE IV

PERFORMANCES OF THE DEPSO-BASED CLUSTERING ALGORITHM, USING THE CH, CS, DB, $Dunn$, GD_{33}, GD_{53}, I, AND SIL INDICES AS THE FITNESS FUNCTIONS, IN TERMS OF THE JACCARD AND FOWLKES–MALLOWS INDICES. GIVEN ARE THE MEAN AND STANDARD DEVIATION BASED ON 100 RUNS. THE BEST PERFORMANCE IN TERMS OF MEAN VALUES FOR EACH DATA SET IS HIGHLIGHTED IN BOLD

Data Set	Fitness Function															
	CH		CS		DB		Dunn		GD_{33}		GD_{53}		I		SIL	
	J	FM	J	FM	J	FM	J	FM	J	FM	J	FM	J	FM	J	FM
2d4c	0.8599 ± 0.0842	0.9239 ± 0.0498	0.5254 ± 0.0142	0.7097 ± 0.0093	0.8693 ± 0.0160	0.9314 ± 0.0089	**0.9736 ± 0.0272**	**0.9866 ± 0.0140**	0.8693 ± 0.0077	0.9315 ± 0.0047	0.8896 ± 0.0003	0.9430 ± 0.0002	0.7589 ± 0.0955	0.8675 ± 0.0545	0.9674 ± 0.0035	0.9834 ± 0.0018
2d10c	0.5247 ± 0.2468	0.6916 ± 0.1782	0.2550 ± 0.0635	0.4816 ± 0.0581	0.5531 ± 0.1428	0.7248 ± 0.1099	0.3172 ± 0.1774	0.5410 ± 0.1437	0.2683 ± 0.0024	0.4931 ± 0.0024	0.2690 ± 0.0038	0.4990 ± 0.0082	**0.7950 ± 0.0653**	**0.8862 ± 0.0419**	0.7066 ± 0.0280	0.8326 ± 0.0190
2d20c	**0.7075 ± 0.2725**	0.8128 ± 0.2025	0.1786 ± 0.0097	0.4189 ± 0.0127	0.1934 ± 0.0248	0.4345 ± 0.0276	0.4238 ± 0.0883	0.6324 ± 0.0692	0.1321 ± 0.0051	0.3536 ± 0.0087	0.1324 ± 0.0012	0.3582 ± 0.0039	0.6612 ± 0.1277	0.7990 ± 0.0855	0.7037 ± 0.0895	**0.8296 ± 0.0556**
10d20c	0.1458 ± 0.0146	0.3630 ± 0.0175	0.0893 ± 0.0147	0.2844 ± 0.0234	0.0843 ± 0.0176	0.2856 ± 0.0272	0.6177 ± 0.1401	0.7697 ± 0.0987	0.1201 ± 0.0167	0.3361 ± 0.0209	0.0620 ± 0.0033	0.2452 ± 0.0086	0.1090 ± 0.0390	0.2935 ± 0.0471	**0.8625 ± 0.0651**	**0.9263 ± 0.0375**
50d4c	0.6233 ± 0.1154	0.7736 ± 0.0992	0.6001 ± 0.0548	0.7638 ± 0.0431	0.6024 ± 0.0707	0.7727 ± 0.0435	**0.7011 ± 0.0537**	**0.8362 ± 0.0326**	0.4702 ± 0.0573	0.6375 ± 0.0553	0.5730 ± 0.0036	0.7548 ± 0.0025	0.6172 ± 0.1371	0.7658 ± 0.1043	0.6782 ± 0.1243	0.8179 ± 0.0770
100d4c	0.6046 ± 0.0906	0.7626 ± 0.0664	0.3881 ± 0.0422	0.5753 ± 0.0513	0.4471 ± 0.0804	0.6525 ± 0.0545	0.4535 ± 0.0924	0.6681 ± 0.0638	0.6494 ± 0.0149	**0.8029 ± 0.0115**	0.4068 ± 0.0623	0.6263 ± 0.0410	0.4540 ± 0.0928	0.6421 ± 0.0751	**0.6556 ± 0.0151**	0.7976 ± 0.0105
Iris	**0.6840 ± 0.0635**	**0.8116 ± 0.0431**	0.6274 ± 0.0428	0.7749 ± 0.0276	0.5678 ± 0.0491	0.7271 ± 0.0391	0.5951 ± 0.0000	0.7715 ± 0.0000	0.5590 ± 0.0015	0.7381 ± 0.0014	0.5937 ± 0.0031	0.7701 ± 0.0028	0.5608 ± 0.0420	0.7220 ± 0.0339	0.5951 ± 0.0000	0.7715 ± 0.0000
Wine	0.4617 ± 0.0155	0.6442 ± 0.0150	0.3124 ± 0.0027	0.5241 ± 0.0047	0.5328 ± 0.1348	0.6954 ± 0.1015	0.4935 ± 0.0053	0.6848 ± 0.0056	0.5821 ± 0.1274	0.7361 ± 0.0950	0.4967 ± 0.0830	0.6763 ± 0.0628	0.5374 ± 0.1229	0.6982 ± 0.0982	**0.8049 ± 0.0928**	**0.8910 ± 0.0603**
Wisconsin Breast Cancer	0.8201 ± 0.0004	0.9014 ± 0.0002	0.6973 ± 0.0438	0.8240 ± 0.0281	0.8173 ± 0.0276	0.8998 ± 0.0164	0.7068 ± 0.1149	0.8317 ± 0.0696	0.6563 ± 0.0060	0.7973 ± 0.0041	0.5688 ± 0.0554	0.7430 ± 0.0345	0.8343 ± 0.0489	0.9093 ± 0.0298	**0.8624 ± 0.0004**	**0.9262 ± 0.0002**
SRBCT	0.2452 ± 0.0400	0.4088 ± 0.0554	0.2525 ± 0.0072	0.4717 ± 0.0184	0.2660 ± 0.0017	**0.5108 ± 0.0025**	**0.2617 ± 0.0030**	0.4967 ± 0.0081	0.2632 ± 0.0054	0.5008 ± 0.0161	0.2651 ± 0.0041	0.5068 ± 0.0076	0.2595 ± 0.0145	0.4841 ± 0.0383	0.2636 ± 0.0010	0.5022 ± 0.0034

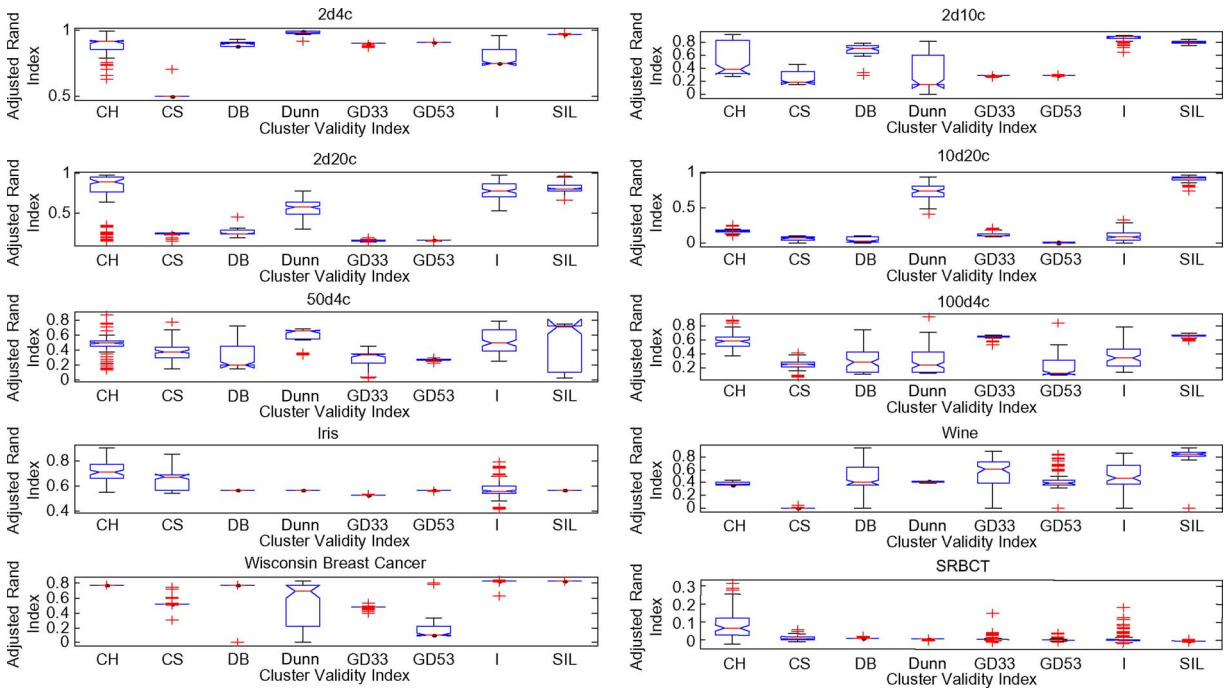

Fig. 2. Box-and-whisker plot that summarizes the clustering results of DEPSO-based clustering using the eight different cluster validity indices as the fitness functions on the ten synthetic and real data sets over 100 runs in terms of the adjusted Rand index.

The performances of DEPSO-based clustering with the eight cluster validity indices as the fitness functions are illustrated in Tables III and IV in terms of the Rand, adjusted Rand, Jaccard, and Fowlkes-Mallows indices. The corresponding box-and-whisker plot and the best clustering performance for the adjusted Rand index are shown in Figs. 2 and 3, respectively (because the results from the adjusted Rand index are similar to those from the other three indices, as observed from Tables III and IV, we only illustrate and discuss the results of the adjusted Rand index in our further analyses for the sake of clarity). All of the aforementioned results are based on 100 runs. As the tables and figures clearly indicate, the SIL index generally leads to the most stable and reliable performance on most data sets considered in this study among the eight indices examined,

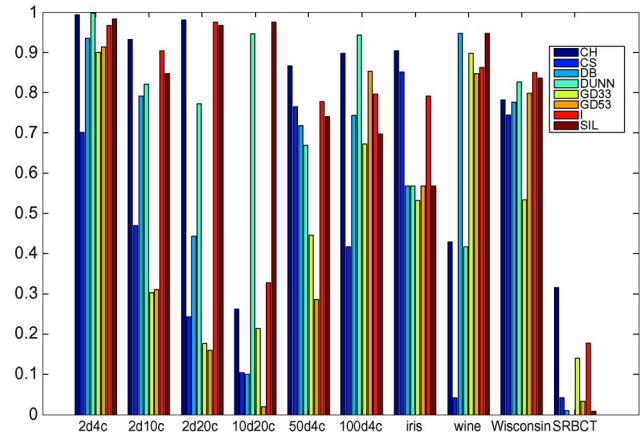

Fig. 3. Best clustering performance of DEPSO-based clustering, in terms of the adjusted Rand index, using the eight different cluster validity indices as the fitness functions over 100 runs. The bars are ordered from left to right as CH, CS, DB, $Dunn$, GD_{33}, GD_{53}, I, and SIL.

although some indices, such as the CH, $Dunn$, and I indices, can achieve nearly perfect performance in some runs. Such performance is also observed with the one-tailed t-test for comparing the differences between the SIL index results with respect to the seven other indices based on the adjusted Rand index, as summarized in Table V. A similar conclusion can also be drawn based on the Rand index, Jaccard coefficient, and Fowlkes-Mallows index. For example, the p-values indicate that the clustering performance of DEPSO using the SIL index is statistically better than the performance using any other index, at a 5% significance level, in four out of nine data sets. Moreover, in three out of the remaining five data sets, the SIL index performs extremely significantly or significantly better than all but one of the other indices in a statistical sense, at a 5% significance level. One data set on which the SIL index does not perform well is the iris data set; the SIL index treats the overlapping iris versicolor and iris virginica as one category. Similar results are also observed for the Dunn index and its two generalized versions. The only index that can correctly estimate the number of clusters in the iris data set is the CH index. The I index also performs well on our experiments, except for the 10d20c and 100d4c data sets, which achieved one best, three second best, and two third best average clustering performances based on the adjusted Rand index.

From Tables III and IV and Figs. 2 and 3, it can also be observed that the performance of DEPSO-based clustering deteriorates as the dimensionality of the data sets increases. For example, for the synthetic data sets, the best means of the adjusted Rand index are all above 0.81, with dimensionality no greater than ten. However, these values drop significantly when the dimensionalities of the data sets increase to 50 or 100. For the high-dimensional SRBCT data set, all indices fail to generate the four cancer clusters because of the inclusion of a large amount of genes that are irrelevant to the cancer types of interest. High dimensionality contributes significantly to the data complexity in clustering, and advances in feature extraction and biclustering provide an effective way to improve the performance of clustering algorithms and decrease computational burden [2], [59], [60].

Table VI summarizes the number of clusters estimated with the use of the eight cluster validity indices. Certain indices, such as CS, GD_{33}, and GD_{53}, underestimate the number of clusters in many cases, particularly when many clusters exist (e.g., 20 in this study). Meanwhile, the I index tends to divide some large clusters into smaller clusters, thus overestimating the number of clusters. Furthermore, when the number of clusters is 20, only the SIL index can expose the real clustering structures; however, when the dimensions are increased to 50 or 100 (the results are not shown here), no index can correctly identify the real number of clusters.

Fig. 4 shows the performance comparison in terms of running time for the DEPSO-based clustering algorithm to complete 5000 epochs on five synthetic data sets,[1] using the eight indices as the fitness functions. The DEPSO-based clustering algorithm is written in C++, and all experiments were run on a 2.4-GHz Intel Pentium 4 processor with 512 MB of DDR RAM. We do not specifically optimize the codes for this study. As shown in the figure, the CH, DB, GD_{53}, and I indices achieved fast performance in all cases, while the calculation burden could become an issue for the CS, $Dunn$, and SIL indices with large data sets. For example, the SIL index takes, on average, 4631.1 s to complete 5000 iterations for the 2d4c data set, which is around 29 times slower than that of the CH index (156.89 s on average) and 27 times slower than that of the DB index (170.43 s on average). This is because the silhouette width for every data object must be calculated. For large-scale data analyses with tens of thousands of data objects or more, a possible solution is parallel technology, specifically in terms of graphics processing units (GPUs). Although GPUs have gained familiarity mostly from computer games and 3-D graphics processors, they are increasingly becoming a complementary platform for general-purpose computation, as in clustering, which usually involves highly expensive computation. The population-based property of PSO makes it an ideal candidate for GPU implementation, and the SIL-index-based fitness function can also be calculated in a parallel way. However, it is necessary to understand and take into account the limitations of the graphics processing hardware when developing algorithms targeted at the GPU.

To investigate to what extent the results depend on the selected swarm-intelligence-based clustering algorithm, we also apply PSO- and DE-based clustering algorithms to the iris, wine, 2d4c, and 100d4c data sets. The box-and-whisker plots of the DE- and PSO-based clustering performances, evaluated by the adjusted Rand index, are shown in Fig. 5. Consistent with the results of DEPSO-based clustering, the SIL index usually generates good clustering partitions with little variance across different runs. Another observation is that, in many cases, DEPSO-based clustering performs better than or the same as DE- and PSO-based clustering in terms of the mean adjusted Rand index values. An exception is that PSO-based clustering performs better on the wine data set than DEPSO-based clustering for all indices except the DB index. In contrast, four indices fail to identify the wine categories when DE-based clustering is used.

[1]The results for other data sets, which perform similarly, are not shown here for the purpose of clarity.

TABLE V
COMPARISON OF PERFORMANCES OF SIL AGAINST THE CH, CS, DB, $Dunn$, GD_{33}, GD_{53}, AND I INDICES AS THE FITNESS FUNCTIONS FOR DEPSO-BASED CLUSTERING WITH A ONE-TAILED t-TEST. THE ANALYSIS IS BASED ON THE RESULTS IN TABLE II USING THE ADJUSTED RAND INDEX

Data Set	SIL vs.													
	CH		CS		DB		Dunn		GD_{33}		GD_{53}		I	
	p-value	Signi-ficance	*p*-value	Signi-ficance	*p*-value	Signi-ficance	*p*-value	Signi-ficance	*p*-value	Signi-ficance	*p*-value	Signi-ficance	*p*-value	Signi-ficance
2d4c	$<10^{-21}$	ES	$<10^{-140}$	ES	$<10^{-80}$	ES	0.9731	NS	$<10^{-133}$	ES	$<10^{-139}$	ES	$<10^{-38}$	ES
2d10c	$<10^{-12}$	ES	$<10^{-76}$	ES	$<10^{-14}$	ES	$<10^{-35}$	ES	$<10^{-142}$	ES	$<10^{-146}$	ES	1	NS
2d20c	0.0814	NS	$<10^{-103}$	ES	$<10^{-125}$	ES	$<10^{-49}$	ES	$<10^{-103}$	ES	$<10^{-101}$	ES	0.0014	S
10d20c	$<10^{-177}$	ES	$<10^{-196}$	ES	$<10^{-204}$	ES	$<10^{-28}$	ES	$<10^{-192}$	ES	$<10^{-139}$	ES	$<10^{-141}$	ES
50d4c	0.4031	NS	0.0016	S	$<10^{-4}$	S	0.9961	NS	$<10^{-8}$	ES	$<10^{-9}$	ES	0.9141	NS
100d4c	$<10^{-9}$	ES	$<10^{-93}$	ES	$<10^{-39}$	ES	$<10^{-40}$	ES	$<10^{-13}$	ES	$<10^{-58}$	ES	$<10^{-33}$	ES
Iris	1	NS	1	NS	0.5	NS	0.5	NS	$<10^{-139}$	ES	$<10^{-5}$	ES	0.7070	NS
Wine	$<10^{-51}$	ES	$<10^{-75}$	ES	$<10^{-29}$	ES	$<10^{-48}$	ES	$<10^{-22}$	ES	$<10^{-48}$	ES	$<10^{-28}$	ES
Wisconsin Breast Cancer	0	ES	$<10^{-54}$	ES	$<10^{-13}$	ES	$<10^{-17}$	ES	$<10^{-140}$		$<10^{-68}$	ES	$<10^{-5}$	S
SRBCT	1	NS	$<10^{-4}$	S	0.4693	NS	0.0016	S	0.7803	NS	0.8394	NS	0.9995	NS

NS: Not Significant; S: Significant; ES: Extremely Significant

TABLE VI
ESTIMATED NUMBER OF CLUSTERS OF THE DEPSO-BASED CLUSTERING ALGORITHM, USING THE CH, CS, DB, $Dunn$, GD_{33}, GD_{53}, I, AND SIL INDICES AS THE FITNESS FUNCTIONS. GIVEN ARE THE MEAN AND STANDARD DEVIATION BASED ON 100 RUNS

Data Set	Actual Number of Clusters	Fitness Function							
		CH	CS	DB	Dunn	GD_{33}	GD_{53}	I	SIL
2d4c	4	4.2000 ±1.0731	2.0200 ±0.2000	3.5600 ±0.4989	3.9300 ±0.3258	3.1600 ±0.3685	3.0000 ±0.0000	7.3200 ±0.9522	4.0000 ±0.0000
2d10c	10	6.1800 ±3.2610	2.7000 ±0.9374	6.7700 ±1.2048	3.7000 ±2.5086	3.0000 ±0.0000	3.0000 ±0.0000	12.6400 ±1.3523	8.8100 ±0.7875
2d20c	20	15.5100 ±5.8767	3.9800 ±0.1407	5.7700 ±0.8860	14.6300 ±3.2957	3.0000 ±0.0000	3.0000 ±0.0000	17.5600 ±2.7719	17.8300 ±1.8426
10d20c	20	3.1600 ±0.8729	2.0300 ±0.1714	2.0000 ±0.0000	13.5700 ±1.9810	2.3200 ±0.4899	2.0000 ±0.0000	6.5200 ±1.8882	17.3700 ±1.0314
50d4c	4	2.6300 ±0.7608	4.9900 ±0.7316	2.5700 ±0.8196	2.8000 ±0.4020	2.0000 ±0.0000	2.0000 ±0.0000	4.3000 ±0.7720	3.2600 ±0.9705
100d4c	4	3.2900 ±0.8077	4.8200 ±0.6257	2.8500 ±0.9252	2.3000 ±0.4820	2.0000 ±0.0000	2.4000 ±0.5685	4.0700 ±1.1033	4.0000 ±0.0000
Iris	3	3.0000 ±0.0000	6.8700 ±1.6370	4.2100 ±0.6079	2.0000 ±0.0000	2.0000 ±0.0000	2.0000 ±0.0000	4.1300 ±0.6913	2.0000 ±0.0000
Wine	3	2.0000 ±0.0000	2.0000 ±0.0000	2.7900 ±0.8324	2.0000 ±0.0000	2.9100 ±0.8299	2.2800 ±0.4940	3.1700 ±0.9955	3.7900 ±0.6403
Wisconsin Breast Cancer	2	2.0000 ±0.0000	2.0000 ±0.0000	1.9900 ±0.1000	2.4800 ±0.6432	2.0000 ±0.0000	2.0000 ±0.0000	2.8300 ±0.3775	2.0000 ±0.0000
SRBCT	4	2.3800 ±0.5081	6.7500 ±1.3286	2.0000 ±0.0000	2.2400 ±0.4522	2.1200 ±0.4090	2.0000 ±0.0000	2.4700 ±0.6269	2.0000 ±0.0000

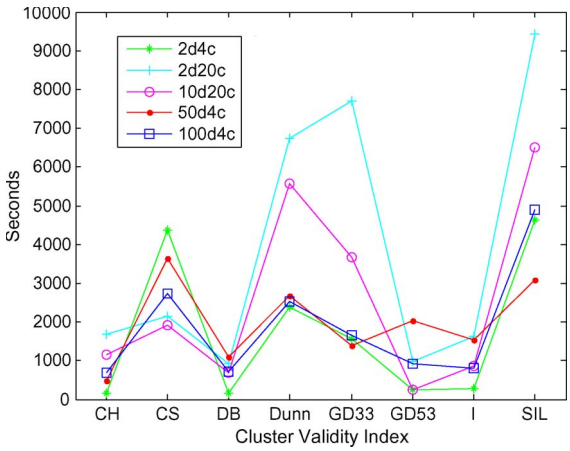

Fig. 4. Average time for the DEPSO-based clustering algorithm to run 5000 epochs with CH, CS, DB, $Dunn$, GD_{33}, GD_{53}, I, and SIL as the fitness functions.

The box-and-whisker plots of DEPSO-based clustering with the cognitive and social components set at $c_1 = 2.5$ and $c_2 = 0.5$ and at $c_1 = 0.5$ and $c_2 = 0.5$, respectively, are shown in Fig. 6. As shown in Table II, the clustering performance of certain indices on some data sets (e.g., the SIL index on the 2d4c data set) improves when we change the values of c_1 and c_2 from our assigned values (2.5 and 1.5). However, this change does not affect our observation that the SIL index usually leads to better clustering performance than the other indices considered here. We can also see that the CH index is a potential alternative, which is consistent with the previous results.

V. CONCLUSION

Swarm intelligence techniques, such as PSO, provide new search capabilities for exploring the embedded clustering structures of data of interest, a complicated optimization problem.

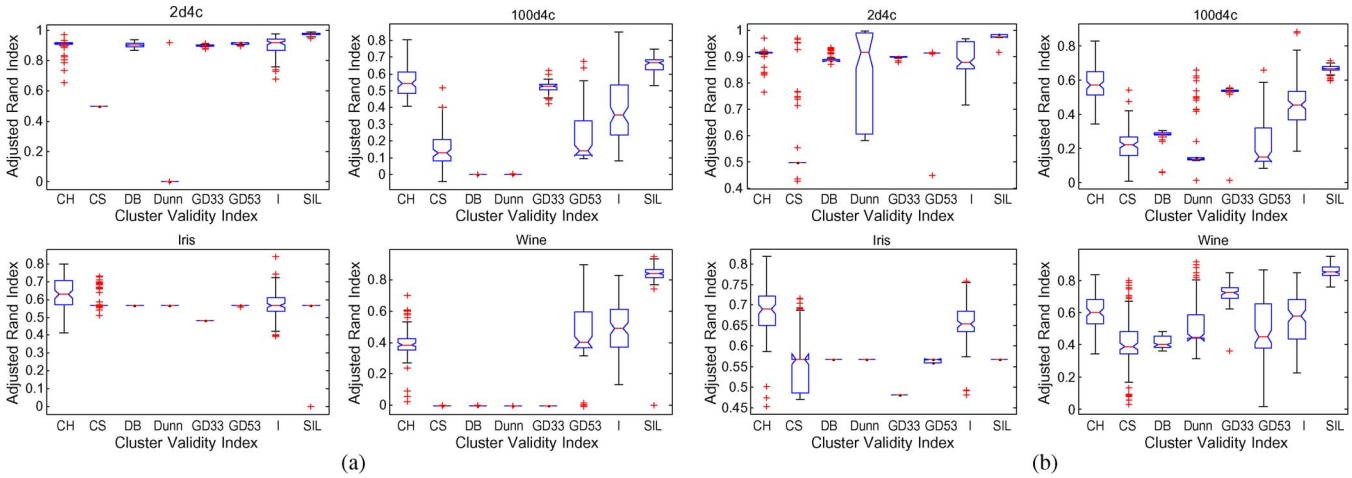

Fig. 5. Box-and-whisker plot summarizing the clustering results of (a) DE-based clustering and (b) PSO-based clustering, with the eight different cluster validity indices as the fitness functions on the iris, wine, 2d4c, and 100d4c data sets over 100 runs in terms of the adjusted Rand index.

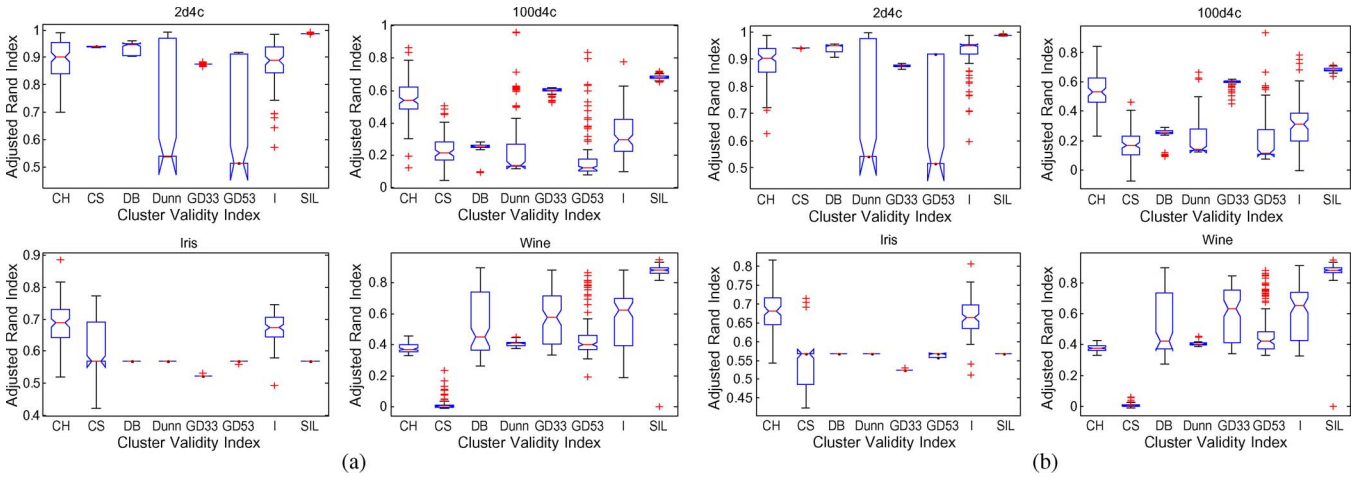

Fig. 6. Box-and-whisker plot summarizing the clustering results of DEPSO-based clustering when the cognitive and social components c_1 and c_2 are set (a) at 2.5 and 0.5 and (b) at 0.5 and 0.5, respectively. The results are evaluated in terms of the adjusted Rand index over 100 runs, using each of the eight cluster validity indices as the fitness functions on the iris, wine, 2d4c, and 100d4c data sets.

A major implementation problem associated with such applications is to determine an appropriate fitness function in order to evaluate the qualities of the obtained clustering solutions. Currently, a large number of clustering validity indices, each with its own limitations and developed based on different considerations and rationales, are used for this purpose. Thus, different indices selected as the fitness function may produce completely different clustering results, thus confusing or even misleading the users. In order to provide users with some useful insights into selecting clustering validity indices, here, we examined the performances of eight clustering validity indices, namely, the CH index, the CS index, the DB index, the Dunn index with two of its generalized versions, the I index, and the SIL index, which are widely used as fitness functions for swarm intelligence and other evolutionary-computational-approach-based clustering. The swarm intelligence method that we used is based on the combined concepts of DE and PSO for the purpose of enhancing the search capabilities. The encoding strategy used here can effectively estimate the number of clusters during the evolution process without requiring it to be determined *a priori*.

According to our experimental results on both synthetic and real data sets, the SIL index stands out among these indices, demonstrating performance comparable to or better than that of other indices. Such performance may be due to the fact that SIL calculates a quality measure for each data object, i.e., the data object's relative closeness to the cluster that it is assigned to, rather than depending on the overall variance. However, its high computational burden requires extra consideration when large-scale data analysis is involved. Some other indices, such as the CH index, the Dunn index, and the I index, can also effectively find the real clustering structures for some data sets, but their good performance is not consistent. These results reinforce the previous studies in [30], [32], [51], and [52]. For example, Bandyopadhyay and Maulik reported the superiority of the I index over the DB index, the Dunn index, and two generalized Dunn indices [30]. The CH index has long been considered an important index for cluster validation [51].

Overall, we recommend the SIL index as a good fitness function candidate for small- and moderate-sized data sets, although a specific speed-up strategy should be considered when using it for large-scale data clustering. Meanwhile, we

strongly discourage relying on clustering solutions using any single clustering validity index; thus, we suggest using indices such as the I index and the CH index to further verify the clustering solutions. Any conclusion about the structures in the data should be based on the results of several indices. Moreover, a possible way to define a fitness function that may contain conflicting objectives is to use weighted indices, as in [52]. Such a requirement may stem in part from the insufficiency of a single index in judging the clustering of high-dimensional data, as shown in the experimental results. However, our experiments that combined the CH, DB, $Dunn$, I, and SIL indices with equal assigned weights do not show a significant improvement of the weighted indices over the SIL index consistently. A future research direction is to investigate how to adapt the components of the weighted sum index and the appropriate weight for each considered index to different types of applications.

ACKNOWLEDGMENT

The authors would like to thank Dr. J. Handl and Dr. J. Knowles for making the data available. The authors would also like to thank the reviewers and the Associate Editor for the extensive comments.

REFERENCES

[1] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.
[2] R. Xu and D. Wunsch, II, *Clustering*. Hoboken, NJ: IEEE/Wiley Press, 2009.
[3] R. Xu and D. Wunsch, II, "Clustering algorithm in biomedical research: A review," *IEEE Rev. Biomed. Eng.*, vol. 3, pp. 120–154, 2010.
[4] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[5] B. Everitt, S. Landau, and M. Leese, *Cluster Analysis*, 4th ed. London, U.K.: Arnold, 2001.
[6] P. Hansen and B. Jaumard, "Cluster analysis and mathematical programming," *Math. Program.*, vol. 79, pp. 191–215, 1997.
[7] E. Forgy, "Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications," *Biometrics*, vol. 21, no. 3, pp. 768–780, 1965.
[8] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp.*, 1967, vol. 1, pp. 281–297.
[9] D. Steinley, " K-means clustering: A half-century synthesis," *Brit. J. Math. Stat. Psychol.*, vol. 59, pp. 1–34, May 2006.
[10] D. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 3rd ed. Piscataway, NJ: Wiley-IEEE Press, 2005.
[11] X. Yao, "Evolving artificial neural networks," *Proc. IEEE*, vol. 87, no. 9, pp. 1423–1447, Sep. 1999.
[12] E. Hruschka, R. Campello, A. Freitas, and A. 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.
[13] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.
[14] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.
[15] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. New York: Wiley, 1989.
[16] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proc. IEEE*, vol. 86, no. 11, pp. 2210–2239, Nov. 1998.
[17] R. Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications, and recourses," in *Proc. Congr. Evol. Comput.*, 2001, pp. 81–86.
[18] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. San Diego, CA: Academic, 2001.
[19] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.
[20] A. Abraham, S. Das, and S. Roy, "Swarm intelligence algorithms for data clustering," in *Soft Computing for Knowledge Discovery and Data Mining*, O. Maimon and L. Rokach, Eds. New York: Springer-Verlag, 2008, pp. 279–313.
[21] J. Handl, J. Knowles, and M. Dorigo, "Ant-based clustering and topographic mapping," *Artif. Life*, vol. 12, no. 1, pp. 35–61, 2006.
[22] P. Kanade and L. Hall, "Fuzzy ants and clustering," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 5, pp. 758–769, Sep. 2007.
[23] T. Runkler, "Ant colony optimization of clustering models," *Int. J. Intell. Syst.*, vol. 20, no. 12, pp. 1233–1251, Dec. 2005.
[24] D. Merwe and A. Engelbrecht, "Data clustering using particle swarm optimization," in *Proc. Congre. Evol. Comput.*, 2003, vol. 1, pp. 215–220.
[25] X. Cui, T. Potok, and P. Palathingal, "Document clustering using particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, 2005, pp. 185–191.
[26] F. Yang, T. Sun, and C. Zhang, "An efficient hybrid data clustering method based on K-harmonic means and particle swarm optimization," *Expert Syst. Appl.*, vol. 36, no. 6, pp. 9847–9852, Aug. 2009.
[27] T. Runkler and C. Katz, "Fuzzy clustering by particle swarm optimization," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2006, pp. 601–608.
[28] M. Omran, A. Salman, and A. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Anal. Appl.*, vol. 8, no. 4, pp. 332–344, Feb. 2006.
[29] S. Das, A. Abraham, and A. Konar, "Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm," *Pattern Recognit. Lett.*, vol. 29, no. 5, pp. 688–699, Apr. 2008.
[30] S. Bandyopadhyay and U. Maulik, "Nonparametric genetic clustering: Comparison of validity indices," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 1, pp. 120–125, Feb. 2001.
[31] S. Bandyopadhyay and S. Saha, "GAPS: A clustering method using a new point symmetry based distance measure," *Pattern Recognit.*, vol. 40, no. 12, pp. 3430–3451, Dec. 2007.
[32] 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.
[33] R. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat.*, vol. 3, no. 1, pp. 1–27, 1974.
[34] C. Chou, M. 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, Jul. 2004.
[35] D. Davies and D. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
[36] J. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1974.
[37] J. Bezdek and N. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 301–315, Jun. 1998.
[38] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley, 1990.
[39] R. Xu, J. Xu, and D. Wunsch, II, "Clustering with differential evolution particle swarm optimization," in *Proc. IEEE World Congr. Comput. Intell.*, Barcelona, Spain, 2010, pp. 1481–1488.
[40] W. Zhang and X. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2003, pp. 3816–3821.
[41] R. Xu, G. Venayagamoorthy, and D. Wunsch, II, "Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization," *Neural Netw.*, vol. 20, no. 8, pp. 917–927, Oct. 2007.
[42] P. Moore and G. Venayagamoorthy, "Evolving digital circuits using hybrid particle swarm optimization and differential evolution," *Int. J. Neural Syst.*, vol. 16, no. 3, pp. 163–177, Jun. 2006.
[43] Y. delValle, G. Venayagamoorthy, S. Mohagheghi, J. Hernandez, and R. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.
[44] B. Jarboui, M. Cheikh, P. Siarry, and A. Rebai, "Combinatorial particle swarm optimization (CPSO) for partitional clustering problem," *Appl. Math. Comput.*, vol. 192, no. 2, pp. 337–345, Sep. 2007.
[45] E. Hruschka and N. Ebecken, "A genetic algorithm for cluster analysis," *Intell. Data Anal.*, vol. 7, no. 1, pp. 15–25, Jan. 2003.
[46] E. Falkenauer, *Genetic Algorithms and Grouping Problems*. Chichester, U.K.: Wiley, 1998.
[47] A. Gordon, "Cluster validation," in *Data Science, Classification, and Related Methods*, C. Hayashi, N. Ohsumi, K. Yajima, Y. Tanaka, H. Bock, and Y. Bada, Eds. New York: Springer-Verlag, 1998, pp. 22–39.
[48] R. Dubes, "Cluster analysis and related issue," in *Handbook of Pattern Recognition and Computer Vision*, C. Chen, L. Pau, and P. Wang, Eds. River Edge, NJ: World Science, 1993, pp. 3–32.

[49] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods: Part I," *ACM SIGMOD Rec.*, vol. 31, no. 2, pp. 40–45, Jun. 2002.
[50] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods: Part II," *ACM SIGMOD Rec.*, vol. 31, no. 3, pp. 19–27, Sep. 2002.
[51] G. Milligan and M. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.
[52] 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.
[53] J. Handl and J. Knowles, "Improving the scalability of multiobjective clustering," in *Proc. Congr. Evol. Comput.*, 2005, vol. 3, pp. 2372–2379.
[54] A. Asuncion and J. Newman, *UCI Machine Learning Repository*, Irvine, CA, School Inf. Comput. Sci., Univ. California, 2007. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[55] J. Khan, J. Wei, M. Ringnér, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. Antonescu, C. Peterson, and P. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nat. Med.*, vol. 7, pp. 673–679, 2001.
[56] L. Hubert and P. Arabie, "Comparing partitions," *J. Classif.*, vol. 2, no. 1, pp. 193–218, Dec. 1985.
[57] G. Milligan and M. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivar. Behav. Res.*, vol. 21, no. 4, pp. 441–458, 1986.
[58] D. Steinley, "Properties of the Hubert–Arabie adjusted Rand index," *Psychol. Methods*, vol. 9, no. 3, pp. 386–396, Sep. 2004.
[59] R. Xu and D. Wunsch, II, "BARTMAP: A viable structure for biclustering," *Neural Netw.*, vol. 24, no. 7, pp. 709–716, Sep. 2011.
[60] S. Madeira and A. Oliveira, "Biclustering algorithms for biological data analysis: A survey," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 1, no. 1, pp. 24–45, Jan.–Mar. 2004.

Rui Xu (S'00–M'06) received the B.E. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1997, the M.E. degree in electrical engineering from Sichuan University, Chengdu, China, in 2000, and the Ph.D. degree in electrical engineering from Missouri University of Science & Technology, Rolla, in 2006.

He is currently an Information Technologist with the Machine Learning Laboratory, GE Global Research, Niskayuna, NY. His research interests include computational intelligence, machine learning, pattern clustering and classification, neural networks, evolutionary computation, and bioinformatics.

Jie Xu (S'01–M'11) received the B.S. degree in electrical engineering from Nanjing University, Nanjing, China, in 1999, the M.E. degree in electrical engineering from Shanghai Jiaotong University, Shanghai, China, in 2002, the M.S. degree in computer science from The State University of New York, Buffalo, in 2004, and the Ph.D. degree in industrial engineering and management sciences from Northwestern University, Evanston, IL, in 2009.

He is currently the Assistant Professor of Systems Engineering and Operations Research with George Mason University, Fairfax, VA. His principal research interests include stochastic simulation and optimization, evolutionary computation, and their applications in risk management, revenue management, and production planning.

Dr. Xu is a member of the IEEE Systems, Man, and Cybernetics Society, the IEEE Computational Intelligence Society, and the Institute for Operations Research and the Management Sciences and a senior member of the Institute of Industrial Engineers.

Donald C. Wunsch, II (M'85–SM'97–F'05) received the B.S. degree in applied mathematics from the University of New Mexico, Albuquerque, and the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from the University of Washington, Seattle.

Since 1999, he has been with Missouri University of Science & Technology, Rolla, where he is currently the Mary K. Finley Missouri Distinguished Professor of Computer Engineering. His prior positions were Associate Professor and Director of the Applied Computational Intelligence Laboratory, Texas Tech University, Lubbock; Senior Principal Scientist at Boeing; Consultant for Rockwell International; and Technician for International Laser Systems. His key research contributions are clustering, adaptive resonance and reinforcement learning architectures, hardware and applications, neurofuzzy regression, traveling salesman problem heuristics, robotic swarms, and bioinformatics. He has produced 16 Ph.D. recipients in computer engineering, electrical engineering, and computer science, has attracted over $8 million in sponsored research, and has over 300 publications including nine books. His research has been cited over 2500 times.

Dr. Wunsch was a fellow and the past President of the International Neural Network Society (INNS) and has been an INNS senior fellow since 2007. He served as General Chair of the International Joint Conference on Neural Networks and on several boards, including the St. Patrick's School Board, the IEEE Neural Networks Council, the INNS, and the University of Missouri Bioinformatics Consortium.