Swarm Intelligence-Based Clustering Algorithms: A Survey

Tülin İnkaya, Sinan Kayalıgil, and Nur Evin Özdemirel

Abstract Swarm intelligence (SI) is an artificial intelligence technique that depends on the collective properties emerging from multi-agents in a swarm. In this work, the SI-based algorithms for hard (crisp) clustering are reviewed. They are studied in five groups: particle swarm optimization, ant colony optimization, ant-based sorting, hybrid algorithms, and other SI-based algorithms. Agents are the key elements of the SI-based algorithms, as they determine how the solutions are generated and directly affect the exploration and exploitation capabilities of the search procedure. Hence, a new classification scheme is proposed for the SI-based clustering algorithms according to the agent representation. We elaborate on which representation schemes are used in different algorithm categories. We also examine how the SI-based algorithms, together with the representation schemes, address the challenging characteristics of the clustering problem such as multiple objectives, unknown number of clusters, arbitrary-shaped clusters, data types, constraints, and scalability. The pros and cons of each representation scheme are discussed. Finally, future research directions are suggested.

1 Introduction

Clustering aims at discovering the natural groups in a set of points. It forms groups of points based on a similarity measure so that the similarities between the points inside the same group are high, and the similarities between the points in different groups are low [34]. It can be used as an exploratory tool to discover the hidden and potentially useful groups in a data set. It also helps simplification and

T. İnkaya (✉)
Industrial Engineering Department, Uludağ University, Görükle, 16059, Bursa, Turkey
e-mail: tinkaya@uludag.edu.tr

S. Kayalıgil • N.E. Özdemirel
Industrial Engineering Department, Middle East Technical University,
Çankaya, 06800, Ankara, Turkey
e-mail: skayali@metu.edu.tr; nurevin@metu.edu.tr

M.E. Celebi, K. Aydin (eds.), *Unsupervised Learning Algorithms*,
DOI 10.1007/978-3-319-24211-8_12

data compression through generalizations. It has applications in many real-world problems such as image segmentation and clustering [8, 53], gene expression [30], geographic information systems [3], market segmentation [13], and so on.

Clustering is an unsupervised learning method, in which natural groupings are explored without any guidance and use of external information. For instance, the number of clusters in a data set is in general unknown a priori. The attributes may include various types of data, such as numerical, ordinal, and categorical. Hence, the similarity/dissimilarity function should be selected according to the type(s) of the attributes. Other complications with the clustering problem may be arbitrary-shaped clusters and density variations. The existence of such patterns complicates the evaluation of clustering quality. In addition to these, domain specific constraints and handling large data sets are some other challenging issues in the clustering problem.

Clustering problem has been studied in various domains including statistics, pattern recognition, management information systems, optimization, and artificial intelligence. Hence, a vast amount of literature has been accumulated. Basically, clustering approaches can be classified as partitional, hierarchical, density-based, graph-based, probabilistic, metaheuristic, and fuzzy approaches. Partitional approaches divide the data set into a number of groups according to an objective function [10]. Hierarchical approaches find the nested clusters recursively, and build a dendogram. A hierarchical algorithm works either in agglomerative (bottom-up) or divisive (top-down) mode. Density-based approaches form a cluster from a group of points in a dense region surrounded by less dense regions. Graph-based approaches are built upon the connectivity idea in graphs, and a cluster corresponds to a set of connected components. Probabilistic approaches assume that clusters originate from certain probability distributions and try to estimate the associative parameters. Metaheuristic approaches are stochastic search methods that consider clustering as an optimization problem. Typically, they are used to obtain the optimal clustering based on a criterion. Fuzzy clustering approaches determine the membership values when the cluster boundaries are not clear cut.

The flexibility of metaheuristic approaches makes them a promising tool for clustering. In particular, Swarm Intelligence (SI) has become an emerging solution approach for clustering. SI is an artificial intelligence technique that depends on the collective property emerging from multi-agents in a swarm. Agents in the swarm can collectively discover the attractive regions, i.e. core regions of clusters, through sharing density and connectivity information. Moreover, SI is a robust, scalable, and easily distributed tool. Motivated by these, SI has attracted attention of the clustering community, and it has become a widely used solution method. In recent years, several articles have been presented in the intersection of clustering and SI-based approaches. Some of them are ant colony optimization (ACO), ant-based sorting (ABS), particle swarm optimization (PSO), artificial bee colony (ABC), and so on.

There are several review articles about SI-based approaches for clustering [1, 2, 28, 50, 57]. Handl and Meyer [28] focus on the ant-based clustering algorithms, and study them in two categories: (1) methods that directly mimic the real ant colonies such as patch sorting and annular sorting and (2) ant-based optimization methods including ACO and its derivatives. They also discuss other types of swarm-based clustering methods such as PSO and information flocking. Abraham et al. [1] review the clustering algorithms based on ACO and PSO. Rana et al. [57] and Alam et al. [2] provide reviews of PSO-based clustering. Alam et al. [2] classify the PSO clustering algorithms into two categories: (1) algorithms in which PSO is directly used for clustering and (2) algorithms in which PSO is hybridized with another clustering method. Also, [50] study nature-inspired algorithms including evolutionary algorithms, simulated annealing (SA), genetic algorithms (GA), PSO, and ACO. However, their scope is limited to partitional clustering.

The studies in the intersection of data mining and SI are reviewed by Grosan et al. [24] and Martens et al. [48]. Grosan et al. [24] consider the early studies of PSO and ACO in data mining. Martens et al. [48] classify the SI-based data mining approaches into two categories: effective search and data organizing approaches. In effective search approaches, the agents move in the solution space and construct the solution. ACO and PSO are studied in this group. In data organizing approaches, the agents move the data points located in a low-dimensional space to accomplish the clustering task or to find a mapping solution. ABS is considered in this group.

In this paper, we study SI-based approaches for hard (crisp) clustering in five categories: ACO, ABS, PSO, hybrid approaches, and other SI-based approaches. ACO, ABS, and PSO are the well-known and widely used SI approaches that use the collective behavior of ants and birds. In the last decade, new SI-based approaches have emerged such as ABC, wasp swarm optimization (WSO), firefly algorithm (FA), and so on. We examine these algorithms as other SI-based algorithms. Moreover, SI-based approaches are combined with other metaheuristics and AI methods. Even two SI-based approaches can work together. These are considered as hybrid approaches.

Different from the previous review studies, we propose a new classification scheme for the SI-based clustering approaches according to agent representation. Agent representation is a key element, as it determines how solutions are generated. It also directly affects the exploration and exploitation capabilities of the search procedure. Taking into account this key role, we discuss the use of each agent representation scheme. The advantages and disadvantages of representation schemes are elaborated. Finally, we examine which challenging characteristics of the clustering problem are addressed by each representation scheme.

The rest of the chapter is organized as follows. Section 2 provides a definition of the clustering problem including its challenging issues. Section 3 briefly explains each SI-based approach. The new classification scheme based on agent representation is introduced in Sect. 4. The main properties of each representation scheme are explained. In addition, the studies using each representation scheme are discussed. Section 5 includes discussions about the representation schemes, SI-based algorithms, and challenging issues in the clustering problem. Finally, we conclude and give future research directions in Sect. 6.

2 The Clustering Problem

Given a data set $X = \{x_1, .., x_i, .., x_n\}$, which includes n data points each defined with d attributes, clustering forms k groups such that data points in the same cluster are similar to each other, and data points in different clusters are dissimilar.

The aim is to obtain compact, connected, and well-separated clusters. Compactness is related to intra-cluster similarity. It implies that the points in the same cluster should be similar/close to each other. Compactness measures are classified into two categories, namely representative point-based and edge-based measures. Representative point-based measures minimize the total dissimilarity/distance between each cluster member and a point that represents the cluster (centroid, medoid, and so on). When the number of clusters is given a priori, these measures usually lead to spherical clusters, and elongated or spiral clusters can hardly be obtained [19, 25]. The edge-based measures use the pairwise distances of the data points in the same cluster, for example, the sum of pairwise distances within each cluster, or the maximum edge length in the connected graph of a cluster. Edge-based approaches are more powerful than the other methods in handling arbitrary shapes.

Connectivity is the linkage between the data points in close proximity and having similarities. Data points in a certain vicinity of a given data point form the neighbors of that point, and a point and its neighbors should be assigned to the same cluster. For example, [29] calculate the connectivity objective for k-nearest neighbors as the degree of the neighboring data points placed in the same cluster.

Separation implies inter-cluster dissimilarity, that is, data points from different clusters should be dissimilar. Total inter-cluster distances between cluster representatives, single-link, average-link, and complete-link are some commonly used separation measures in the literature. For instance, single-link calculates the dissimilarity between a pair of clusters as the distance between the closest (most similar) points in the two clusters, whereas complete-link takes into account the most distant (most dissimilar) points. In average-link, the average distance between all point pairs in two clusters is used.

In this chapter, we consider the following challenging characteristics of the clustering problem.

(a) *Multiple objectives*: It is possible to conceptualize the aim of clustering, however it is difficult to combine and optimize the three objectives, namely compactness, connectivity, and separation, simultaneously. To the best of our knowledge, there is not a generally accepted clustering objective or measure that fits all data sets. The data set characteristics (types of attributes, shapes, and densities of clusters) and the application field determine the choice of the objective functions to be used in clustering.

(b) *Unknown number of clusters*: In real-world applications such as those in image segmentation, bioinformatics (gene clustering), and geographic information systems, the number of clusters is typically unknown, and there are no given class labels. Instead, one should extract it from the data set as a major piece of knowledge. Some approaches use a validity index to find the number of clusters

[5, 6]. After the execution of the clustering algorithm for different numbers of clusters, the one with the best validity index is selected. However, there is no validity index that works well for all data sets. There are also studies that assume the number of clusters is given. For instance, in a facility location problem, the number of facilities that will serve clusters of customers is typically predetermined. Then, the task is to determine the facility locations and to assign customers to these facilities.

(c) *Arbitrary-shaped clusters*: A data set may include clusters of any size and shape such as ellipsoids, elongated structures, and concentric shapes. Density variations within and between clusters make the clustering task even more difficult. Arbitrary-shaped clusters are in particular seen in geographical information systems, image segmentation, geology, and earth sciences [19, 35]. The success in extraction of arbitrary-shaped clusters with density variations depends on the clustering objective function and the similarity/distance function used. In addition to the compactness and separation objectives, the proximity and connectivity relations are also crucial in discovering these clusters.

(d) *Data type*: A data set may include two types of attributes: (1) qualitative or categorical (binary, nominal, or ordinal), (2) quantitative or numerical (continuous, discrete, or interval). The selection of the similarity/distance measure depends on the attribute type. When all the attributes are quantitative (e.g., height, weight), Manhattan, Euclidean, and Minkowski distances can be used [27]. In order to calculate the dissimilarities in qualitative data (e.g., color, rank), some well-known similarity measures are simple matching coefficient, Jaccard and Hamming distances [27]. If there exists a mixture of quantitative and qualitative attributes in a data set, the general similarity coefficient by Gower [23] and the generalized Minkowski distance by Ichino and Yaguchi [32] can be used. Note that, throughout the text, the terms similarity/dissimilarity measure and distance are used interchangeably.

(e) *Constraints*: Constrained clustering is a semi-supervised learning technique, in which additional conditions should be satisfied while grouping similar points into clusters. Constraints can be interpreted as a priori domain knowledge, and they improve the clustering accuracy and efficiency. However, additional mechanisms are needed in a clustering algorithm to handle such constraints. Constraints can be classified into three categories: cluster-level, attribute-level, and instance-level [7]. Cluster-level constraints include capacity, minimum cluster separation, maximum/minimum cluster diameter, and ϵ-distance neighbor within cluster. Attribute-level constraints are order preferences and constraints on the attribute values. Instance-level (relational) constraints are partial labels and pairwise relationships such as must-link (ML) and cannot-link (CL) constraints.

(f) *Scalability*: Improvements in the data storage and computing enable working with large amounts of data sets. However, as the number of data points and the number of attributes increase, data analysis requires more memory and computing time. For this reason, clustering algorithms that are capable of handling large data sets within a reasonable amount of time are needed.

3 Overview of the Swarm Intelligence-Based Approaches

A swarm is composed of simple agents. Even though there is a decentralized control mechanism in the swarm, it can perform complex tasks (such as finding food sources and protection from enemies) through sharing of information and experience among its members and interacting with its environment. Working principles of a swarm are dictated by the behavior of autonomous agents, the communication methods among the agents and with the environment, and moves of the agents. In the SI-based algorithms, these concepts correspond to agent representation, neighborhood definition, decision rule, exploration and exploitation mechanisms.

The starting point in the design of an SI-based algorithm is the agent representation; other characteristics of SI are built upon this ground. An agent may represent a set of variables in the solution, or it may work as a means to construct the solution. Hence, agent representation is directly related with the solutions obtained by the algorithm and the search procedure. The representation must allow all feasible solutions to be reachable during the search. Another important design element is the neighborhood definition. It is related with the locality of the agents, and defines their next possible moves during the solution construction. Next, a solution component is selected using a decision rule, which is often a stochastic choice mechanism. Together with the neighborhood properties, this rule determines the exploration and exploitation properties of the algorithm. Exploration ensures that the search space is examined efficiently, and the algorithm does not prematurely converge to a local optimal solution. This is usually provided by the randomness in the search process. Exploitation, however, is based on the experience of the swarm. It helps investigate the promising regions of the search space thoroughly. There should be a balance between exploration and exploitation for the success of the algorithm.

We study the SI-based algorithms in five categories: PSO, ACO, ABS, hybrid approaches, and other SI-based approaches. In the hybrid approaches, the aim is to strengthen an SI-based algorithm with the exploration and exploitation properties of other search methods such as other metaheuristics, AI approaches, or another SI-based algorithm. There are also studies that combine SI-based algorithms with local search algorithms such as k-means, k-harmonic mean, and so on. These local search algorithms are deterministic search procedures, and improve the exploitation property of the SI-based algorithm.

In this section, we briefly explain the basic characteristics of each SI-based algorithm. The hybrid approaches are derivatives of SI-based algorithms, therefore they are discussed in Sect. 4.

3.1 Particle Swarm Optimization

PSO was introduced by Kennedy and Eberhart [42]. It is inspired from the social behavior of a bird flock. Each particle represents a solution, and particles in the swarm fly in the search space with velocities updated according to the experiences they gained throughout the search. The aim is to discover better places in the search space. Let $x_i(t)$ and $v_i(t)$ denote the position and velocity of particle i in iteration t, respectively. The new velocity and new position of the particle are calculated according to the following equations:

$$v_i(t+1) = w \times v_i(t) + c1 \times rand1 \times (pbest_i(t) - x_i(t)) + c2 \times rand\,2 \times (gbest(t) - x_i(t)) \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

where w denotes the inertia weight, $c1$ and $c2$ are the acceleration coefficients (learning factors), $rand1$ and $rand2$ are random numbers between 0 and 1, $pbest_i(t)$ and $gbest(t)$ denote the personal best solution of particle i and the global best solution for the swarm, respectively. Random functions in the velocity update ensure exploration, whereas the use of personal and global best solutions directs the particles to the promising regions of the search space.

The outline of the PSO algorithm is presented in Table 1.

3.2 Ant Colony Optimization

ACO was introduced by Dorigo et al. [16] and Dorigo et al. [17]. It is inspired by the foraging behavior of ants. Ants search for food on the ground, and during their search, they deposit a substance called pheromone. This smelling substance is used by the other ants to direct the search. Promising paths have higher pheromone concentration, and more ants are directed towards these paths. In this manner, ants can find the food sources by indirect communication between them.

Each ant constructs a solution at each iteration. In each step of construction, the ant selects a solution component from its neighborhood according to a decision rule.

Table 1 The outline of the PSO algorithm

Set parameters and initialize a population of particles with random positions and velocities in the search space.
While (termination conditions are not met)
For each particle i do
Update the velocity of particle i according to Eq. (1).
Update the position of particle i according to Eq. (2).
Calculate the fitness value of particle i.
Update $pbest_i$ and $gbest$.
End for
End while

The decision rule depends on the pheromone concentration on the solution components as well as some problem specific heuristic information. For instance, the ant chooses solution component i according to the following probability.

$$p_i = \frac{(\tau_i)^{\alpha} (\eta_i)^{\beta}}{\sum\limits_{k \in N(i)} (\tau_k)^{\alpha} (\eta_k)^{\beta}} \tag{3}$$

where τ_i denotes the pheromone concentration of solution component i, η_i is the heuristic information indicating the quality of solution component i, α and β determine the relative effects of pheromone and quality, and $N(i)$ shows the neighborhood of component i. As seen in Eq. (4) below, all components used in a good solution receive the same pheromone deposit. However, not every component in a good solution has an equally high quality, hence η_i is used to reflect the quality differences among the solution components.

After ants construct their solutions, the pheromone concentration of each solution component is updated according to Eqs. (4) and (5).

$$\tau_i = (1 - \rho)\tau_i + \rho \Delta\tau_i \tag{4}$$

$$\Delta\tau_i = \sum_{k \in S | i \in k} w_k f_k \tag{5}$$

where $0 < \rho < 1$ is the rate of pheromone evaporation, S is the set of selected solutions such that they contain solution component i, f_k defines the quality of solution constructed by ant k, and w_k is the weight of solution constructed by ant k. There are different rules for selecting solutions to be used for the update, ranging from all solutions generated in the current iteration to the best solution found in the history of the search.

The pheromone deposit quantified in Eq. (5) helps the ant colony intensify the search towards the good solution components. On the other hand, the evaporation of pheromone in the first term of Eq. (4) and probabilistic selection of the solution components according to Eq. (3) increase the exploration capabilities of ACO.

The outline of the ACO algorithm is presented in Table 2.

Table 2 The outline of the ACO algorithm

Set parameters, and initialize pheromone trails.
While (termination conditions are not met)
For each ant k do
Construct a solution by selecting solution components using Eq. (4).
Calculate the fitness value.
Perform local search.
End for
Update the pheromone concentrations according to Eqs. (5) and (6).
End while

3.3 Ant-Based Sorting

ABS was introduced by Deneubourg et al. [15]. It is inspired from the gathering and sorting behaviors of ants such as corpse clustering, brood sorting, and nest building. Ants work as if they were forming a topographic map. An ant picks up a point in the search space and drops it off near the points similar to it. The pick-up and drop-off operations are performed based on the similarity within the surrounding neighborhood.

Lumer and Faieta [47] generalized this method for clustering. In their model, data points are scattered in a search space, which is split into grids. Ants walk through the grids and move the data points. The probabilities that an ant picks up and drops off point i are denoted as $p_{pick}(i)$ and $p_{drop}(i)$, respectively, and they are calculated as follows.

$$p_{pick}(i) = \left(\frac{k_1}{k_1 + f(i)}\right)^2 \tag{6}$$

$$p_{drop}(i) = \begin{cases} 2f(i), & \text{if } f(i) < k_2 \\ 1, & \text{if } f(i) \geq k_2 \end{cases} \tag{7}$$

where k_1 and k_2 are constants, and $f(i)$ is the density dependent function of point i defined as:

$$f(i) = \begin{cases} \frac{1}{s^2} \sum_{j \in N(i)} (1 - d_{ij}/\alpha), & \text{if } f(i) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Here s^2 is the size of the local neighborhood $N(i)$ (vision field), d_{ij} is the distance between points i and j, and α is the dissimilarity scaling parameter.

An ant has a higher pick-up (drop-off) probability for a dissimilar (similar) point compared to a similar (dissimilar) point. Hence, these probabilistic pick-up and drop-off operations support the exploration and exploitation properties of ABS.

The ABS algorithm is outlined in Table 3.

Table 3 The outline of the ABS algorithm

Randomly scatter the data points and ants on the toroidal grid.
While (termination conditions are not met)
For each ant k do
Calculate the similarity between each point and its neighbors.
Pick-up or drop-off a point according to Eqs. (7) and (8).
Move the ant to a randomly selected empty neighboring grid.
End for
End while

3.4 Other Swarm Intelligence-Based Metaheuristics

In recent years, many new nature-inspired algorithms have been proposed based on the collective behavior of bees, wasps, fish, termites, and so on. For the clustering problem, we examine the ABC, WSO, firefly (FA), and artificial fish (AF) algorithms as the other SI-based (OSIB) algorithms.

ABC was introduced by Karaboga [40], and it is based on the foraging behavior of the honeybees. In ABC, there are three groups of bees including employed bees, onlookers, and scouts. Each employed bee visits a potential food source. Onlooker bees choose promising food sources among them. When an employed bee abandons a food source, it becomes a scout, and starts to search for a new food source randomly. In WSO, wasps share limited resources according to their importance to the entire swarm and social status. In FA, fireflies have a luminescence quality, namely luciferin. They emit light proportional to this value, and each firefly is attracted by the brighter glow of other neighboring fireflies. AF is based on the shoal memberships. The fish shoals are open groups. That is, when shoal encounters are observed, individuals can choose neighboring fish with similar phenotypic properties such as color, size, and species.

Note that there are other population-based methods inspired by nature. However, they do not include the collective behavior of the swarm. For instance, genetic algorithms are based on evolving generations through crossover and mutation. These methods are out of our scope, and they are not included in this review.

4 Classification of the Swarm Intelligence-Based Algorithms for Clustering

Taking into account the importance of the agent representation, we classify the SI-based clustering algorithms into four groups: (1) data point-to-cluster assignment, (2) cluster representatives, (3) direct point-agent matching, and (4) search agent. In the data point-to-cluster assignment and cluster representatives schemes, the SI agent keeps information about the clusters. On the other hand, in the direct point-agent matching and search agent schemes, the agents are concerned about data points.

In this section, we provide the main properties of these representation schemes.

4.1 Data Point-to-Cluster Assignment

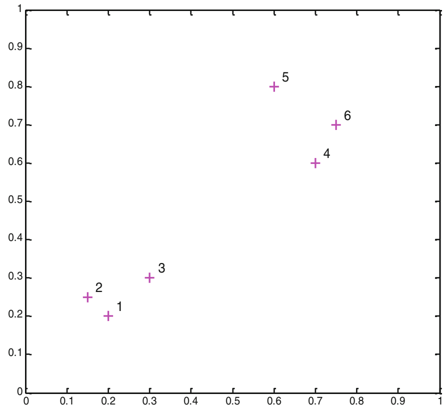
In this representation, an agent keeps track of the cluster assignments of the points in the data set. The number of clusters should be provided a priori. The size of the solution vector is defined by the number of points in the data set, and the number

Data set → $X = \{1, 2, 3, 4, 5, 6\}$

Clustering solution

Cluster 1 → $C_1 = \{1, 2, 3\}$

Cluster 2 → $C_2 = \{4, 5, 6\}$



Point #	1	2	3	4	5	6
Agent	1	1	1	2	2	2

Fig. 1 An example for the data point-to-cluster assignment representation

of attributes in the data set has no impact on the representation. For example, in a two-dimensional data set with six points, an agent is a six-dimensional vector. Each dimension shows the clustering assignment of the respective point. An example is provided in Fig. 1.

Jarboui et al. [36] adopt the data point-to-cluster assignment to PSO. PSO has originally been proposed for search in continuous space, so combinatorial PSO is used to switch between the combinatorial and continuous states. A dummy vector takes values from the set $\{-1, 1, 0\}$ according to the solution status, namely the local best cluster, the global best cluster, or none of these, respectively. After the update in each iteration, the new position of the particle is discretized using the dummy vector and a threshold value. The performance of the proposed approach is tested using two objective functions: minimization of within cluster variation and maximization of variance ratio criterion.

In ACO, there are several studies using this representation scheme [11, 58, 61]. The common point of these studies is that they minimize the within cluster variation/distance. Also, each point-to-cluster assignment is associated with a pheromone concentration. However, the pheromone update mechanisms in these algorithms are different. For instance, there are local and global pheromone update mechanisms in [11]. In the local update, each ant simultaneously updates the pheromone concentration on the particular point-to-cluster assignment. In the global update, after all ants build their solutions, the pheromone concentration is updated only for the global best assignments. Runkler [58] updates the pheromone concentration according to the difference between the fitness values of the incumbent solution and the current solution. Shelokar et al. [61] perform the pheromone update using best L solutions out of R solutions. An extended version of this representation proposed by Jiang et al. [38] can handle attribute selection and clustering tasks together. That is, an ant represents not only the cluster assignments, but also the selected attributes. They use two pheromone matrices, one for the attributes selected and another one for point-to-cluster assignments.

In the OSIB category, [59] uses data point-to-cluster assignment scheme together with WSO. The objective is to minimize within cluster distances. Data point-to-cluster assignments are performed using the stochastic tournament selection. Assignment probability is determined according to the distance between the cluster center and the point. An interesting point in this study is that there is no information exchange among the wasps. Hence, in contrast with the SI principles, collective behavior of the swarm is not observed directly.

ABS forms a topographic map, so the number of clusters is not known a priori. Hence, data point-to-cluster assignment scheme is not applied in ABS. This representation is not used in hybrid approaches either.

The main disadvantage of this representation is that the number of clusters should be given a priori. In addition, the studies that use this representation minimize the within cluster distance, and they typically generate spherical clusters.

4.2 Cluster Representatives

In this scheme, an agent shows the representatives of the clusters such as center, median, medoid, and so on. Typically, the number of clusters (representatives) is given, and the size of the solution vector depends on the number of attributes and the number of clusters. This representation scheme has an analogy with the partitional clustering approaches.

A stream of studies adopts this representation to determine the number of clusters. In one of the revised representations, the maximum number of clusters is used. An additional vector shows whether the corresponding cluster representative is selected or not. The length of this additional vector is equal to the maximum number of clusters. In order to determine the number of clusters, these studies use clustering validity indices such as Dunn's index and Davies Bouldin index for fitness calculation. In another revised representation, an additional variable is used to define the number of clusters for each particle.

In Fig. 2, an example agent is given for a data set with six points and two attributes. If the number of clusters is given as two, the agent corresponds to a four-dimensional vector.

In PSO, for the given number of clusters, there are several studies that use cluster representatives scheme [39, 45, 52, 53, 55, 63, 65, 73]. Van der Merwe et al. [65] propose two PSO algorithms for clustering. These are the standard *gbest* PSO algorithm and a combination of PSO and k-means. Both algorithms minimize the average quantization error, i.e. the distance between each point and its cluster

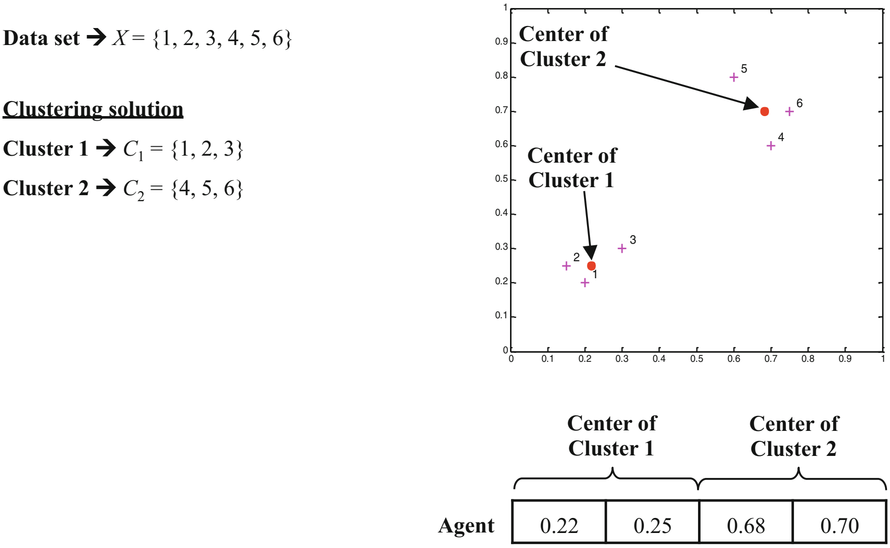


Fig. 2 An example for cluster representatives

representative. In the combined algorithm, k-means seeds the initial swarm. The velocity of a particle is updated according to the particle's previous velocity, its local best, and the swarm's global best. As the particles move, the coordinates of the representative points also change. Omran et al. [52] attempt to consider the multi-objective nature of the clustering algorithm, and use the weighted sum of the maximum average intra-cluster distance and the minimum inter-cluster distance. They test the proposed approach using MRI and satellite data sets. Omran et al. [53] revise the work in [52] by inserting the quantization error term to the objective function. They also use a different version of PSO in which the global best particle is updated by resetting the particle's position to the global best position and using the previous velocity and a random component for search direction. Paterlini and Krink [55] compare the performances of k-means, GA, differential evolution (DE), and PSO. The algorithms give similar results for simple data sets, however population-based heuristics are better for more complex data sets. The performance of DE is superior and more robust compared to the other algorithms, but it has poor performance for noisy data.

A group of studies use PSO together with the well-known local search algorithms to increase the exploitation capability. For example, [39] propose a hybrid algorithm combining Nelder-Mead simplex algorithm and PSO. Nelder-Mead algorithm is used for local search, whereas PSO is used as an exploration tool. The proposed algorithm converges to the global optimal solution with a relatively smaller error rate. Also, it provides an improvement in the number of function evaluations. Yang et al. [73] combine PSO with k-harmonic means (KHM). KHM is used as a local search method, which is applied to the particles in the swarm every eight generations. In a similar manner, [45] apply PSO together with k-means.

Different from the studies discussed so far, [63] consider document clustering. In order to increase the search capability of the algorithm, they introduce an environmental factor for each centroid, which is the mean of the data points belonging to the corresponding centroid. In the first stage of the proposed approach, instead of global best, the environmental factor is used to update the particle's velocity. In the second stage, the particle's velocity is updated according to not only the environmental factor, but also the global best.

A research stream in PSO assumes that the number of clusters is unknown [12, 14, 54]. They start with a set of potential cluster representatives for the maximum number of clusters. For instance, [54] use binary PSO in which a binary particle shows whether the cluster representative is used in the solution or not. The coordinates of the potential representatives are optimized using the k-means algorithm. Clustering solutions are evaluated using Turi's validity index. Instead of binary PSO, [14] use an activation function value for each potential cluster center. That is, a cluster center becomes active if its activation function value is larger than a given threshold. They also adopt a multi-elitist PSO for the clustering problem. In the proposed approach, the global best is replaced with the local best having the highest growth rate. In this way, early convergence is prevented. Moreover, a kernel-induced similarity measure is used instead of the conventional sum-of-squared distances. They use kernelized CS measure as the fitness function. This helps the algorithm extract complex and linearly non-separable clusters. Cura [12] uses an additional variable to define the number of clusters (K_p) for particle p. In each iteration, the particle first moves in dimension K_p. Next, the existing cluster centers of the particle are split or deleted according to the new K_p. Finally, the particle updates the new cluster centers. The proposed approach aims to obtain compact and well-separated clusters by minimizing the difference between total intra-cluster distances and minimum inter-cluster distance.

Lu et al. [46] extend the cluster representatives scheme for variable (attribute) weighting problem. In the proposed approach, there are three swarms: (1) position swarm of variable weights, (2) velocity swarm of variable weights, and (3) swarm of cluster centroids. Position and velocity particles denote the weight set for each cluster. Also, they use comprehensive learning mechanism by applying a crossover between the best position of the weight matrix and one of the other best personal positions in the swarm. Although the proposed approach runs slowly, its performance is superior to those of the other algorithms.

To the best of our knowledge, cluster representatives scheme is not applied in ACO and ABS. However, it is widely used in hybrid approaches [21, 31, 37, 51, 69, 70, 72]. Jiang and Wang [37] combine cooperative co-evolution (CC) framework with bare-bone PSO (BPSO). Different from classical PSO, BPSO updates the position of the particle according to the Gaussian distribution. CC decomposes the clustering problem into subproblems, and BPSO is applied to find the best solution for each subproblem, which represents a single cluster centroid. Each particle in the subproblem is evaluated using the best particles from the other subproblems.

Huang et al. [31] hybridize the continuous version of ACO (ACOR) and PSO. Both ACOR and PSO search for cluster centers. Pheromone-particle table is used to share the new particle solutions generated with PSO and ACOR. Four types of models are introduced: sequential, parallel, sequential with enlarged pheromone-particle table, and global best exchange. In the sequential model, ACOR and PSO share the same set of solutions using pheromone-particle table. For ACOR, pheromone-particle table shows the pheromone values. For PSO, it shows the current solution. In the parallel approach, new ants are generated directly from the pheromone-particle table, without considering the new particles. The sequential update model with the enlarged pheromone-particle table combines the new particles with the pheromone-particle table, and new ants are generated according to the enlarged pheromone-particle table. The global best exchange approach does not have a pheromone-particle table, and the two algorithms share the best solution only. Among the four models, the sequential update with the enlarged pheromone-particle table gives the best average performance.

Firouzi et al. [21] combine PSO, simulated annealing (SA), and *k*-means. In each iteration, SA searches around the global best, and strengthens the exploration property of PSO. At the end of PSO, *k*-means is applied. Niknam and Amiri [51] address the lack of search capability for the global best particle, and propose a combination of fuzzy adaptive PSO, ACO, and *k*-means. The inertia weight and learning factors of PSO are dynamically adjusted using fuzzy if-then rules. Also, for each particle, a different global best position is assigned according to the ACO best path selection method. Yan et al. [72] propose a hybrid version of ABC and GA. Arithmetic crossover is applied to the randomly selected cluster centers (food sources). Xiao et al. [69] combine self-organizing map (SOM) and PSO. SOM is used for clustering, whereas PSO optimizes the weights of SOM. Note that the weights of SOM correspond to the cluster representatives.

Similar to [14, 70] assume that the number of clusters is unknown, and they use an activation function value for each potential cluster center. They combine PSO and DE, and propose the DEPSO algorithm. DEPSO integrates the DE operator into the PSO procedure in order to diversify the PSO. Also, they compare the performances of various clustering validity indices including the silhouette index, Calinski-Harabasz index, CS index, and Davies-Bouldin index. They conclude that the silhouette index gives superior results in most of the data sets for DEPSO, PSO, and DE-based clustering algorithms.

In the OSIB algorithms, an agent usually denotes the cluster representatives. Fathian et al. [20] apply honeybee mating optimization to the clustering problem. New broods are created by exchanging the drone's genes with the queen's (crossover). Worker bees improve the broods by performing local search (royal jelly). Different from the other SI-based algorithms, the bees do not move in the search space; instead, they exchange the gene information. Hence, this algorithm fits more the principles of evolutionary algorithms than those of SI. Karaboga and Ozturk [41] and Zhang et al. [75] apply ABC algorithm to the clustering problem. Zhang et al. [75] study the unsupervised clustering problem, whereas [41] consider the supervised clustering problem, where each data set is partitioned into two subsets for training and testing. In both algorithms, employed and onlooker bees exploit the search space within the given neighborhood. Employed bees turn into scout bees so that they explore different regions of the search space. Zou et al. [78] propose a modified version of ABC, namely the cooperative ABC algorithm. There is a super best solution vector that keeps the best component in each dimension among all populations. The employed and onlooker bees perform their search using the super best solution.

Senthilnath et al. [60] propose a firefly algorithm for supervised clustering. The main idea behind the firefly algorithm is that particles move towards the attractive particles having higher light intensity. Wan et al. [68] apply chaotic ant swarm to clustering. The swarm performs two phases: chaotic phase and organization phase. An organization variable is introduced to achieve self-organization from the chaotic state. This controls the exploration and exploitation properties.

Unless the data set is projected to a transformed feature space, cluster representatives scheme is limited to generating spherical and ellipsoidal clusters. The revised version of this representation can determine the number of clusters, however most of the studies in the literature assume that the number of clusters is given. Moreover, the revised version should couple with an objective function that can optimize the number of clusters.

4.3 Direct Point-Agent Matching

There is a one-to-one correspondence between the agents and the points in this representation. That is, each agent is loaded with a data point, and the number of agents is equal to the number of data points. Agents are scattered in a two-dimensional search space. They move and carry with them the data points such that similar ones are positioned close to each other whereas dissimilar ones are far apart. Eventually, the ants' movements in the search space form a topographic map.

Direct point-agent matching is applied by Picarougne et al. [56] and Veenhuis and Köppen [66] in PSO. Both algorithms have two phases. In the first phase, similar points are gathered together whereas dissimilar ones are set apart using collective behavior of the swarm. In the second phase, clusters are retrieved using a hierarchical algorithm. The number of clusters is assumed to be unknown in both algorithms. The difference between the two algorithms is in the velocity update mechanism. Picarougne et al. [56] use the concept of flock of agents, whereas [66] is an application of PSO. Picarougne et al. [56] update the agent direction using the sum of all the influences in its neighborhood, and the agent speed using the minimum speed and the agent's isolation degree. Hence, agents in a group travel slowly compared to the agents traveling alone. In [66], a particle's new velocity is calculated using its previous velocity, influence factor for the points in its neighborhood, velocity matching factor to the similar points, and avoidance factor to the dissimilar ones.

Direct point-agent matching is widely used in ABS. Xu et al. [71] incorporate ML and CL constraints into the clustering problem. Points with ML constraints are added to the neighborhood with high weights, whereas points with CL constraints are excluded from the neighborhood by defining high negative weights. An ant sleeps or awakens according to a randomly generated number. It performs two types of heuristic walk, namely max-number direction moving and adaptive direction walking. In the max-number direction moving, the ant moves in the direction where most of its neighbors are located. In adaptive direction walking, direction selection is based on the weighted similarity values. The proposed approach gives better results compared to the previous constrained clustering methods.

Zhang et al. [77] introduce a novel ant movement model based on Renyi entropy. In this approach, the local similarity of each point is measured by entropy rather than distance. Also, kernel entropy component analysis (KECA) is used to create rough clusters. Ghosh et al. [22] incorporate aggregation pheromone to ABS in order to gather similar points into closer proximity. The amount of movement for an ant towards a point depends on the intensity of aggregation pheromone deposited by all other ants at that point. They model the pheromone intensity emitted by an ant using Gaussian distribution. Zhang and Cao [76] integrate ABS with kernel principle component analysis (KPCA). KPCA creates a projection for the rough clusters in order to find the non-spherical clusters. Then, the ABS algorithm is applied.

In the OSIB category, [4] introduce a hierarchical ant-based algorithm, which is inspired from the self-assembly behavior of real ants. Ants build a decision tree such that each ant moves in the tree according to the similarity. At the end of the algorithm, each sub-tree, which is directly connected to the support node, forms a cluster. Hence, a cluster retrieval operation is not needed. They apply the proposed approach to several real-world problems such as human skin analysis, web usage mining, and portal site data. Khereddine and Gzara [43] propose a hierarchical divisive clustering algorithm inspired from artificial fish shoals. The proposed algorithm forms a binary tree of clusters. Each time, the cluster with the maximum diameter is selected, and a bi-partitioning algorithm is applied. The direction of the fish is determined according to the similarity of the data point and its neighbors.

The direct agent-point matching is not used in ACO and hybrid approaches.

The common property of the algorithms in this category is that they do not use an explicit objective function or a fitness value. They implicitly maximize the total within-cluster similarity. The main advantage of this representation is that it visualizes the clusters with a topographic map or a decision tree. Moreover, most of the algorithms in this category determine the number of clusters. However, a separate cluster retrieval operation is needed in most of the studies. Another drawback of this representation is that, as the number of points increases, the number of agents increases leading to scalability problems.

4.4 Search Agent

In this representation, agent is a means to carry the similar points to the same neighborhood and the dissimilar points away from each other. In contrast to direct point-agent matching, agents do not have one-to-one correspondence with the data points, and there are fewer agents than points. Agents move around the search space picking up and dropping off the data points. The number of ants in this representation does not depend on the clustering problem parameters such as the number of points, the number of attributes, and the number of clusters.

İnkaya et al. [33] use the search agent representation in ACO. Ants choose the data points they will visit according to the pheromone concentration on the edges connecting pairs of points. A sequence of visits forms a subtour, and each subtour of an ant forms a cluster. The pheromone concentrations on the visited edges are updated in every iteration. In order to handle arbitrary-shaped clusters and unknown number of clusters, they propose two objective functions based on adjusted compactness and relative separation. The algorithm outputs a set of non-dominated solutions for these two criteria.

Lumer and Faieta [47] are the first authors that use search agent representation in ABS. Later, several variations of ABS are introduced. Martin et al. [49] is inspired from formation of ant cemetery activities, and revise the model in [15] with simpler assumptions about ants' behavior. For instance, when an unloaded ant has a body in one of the eight neighboring cells, it is loaded with probability 1.0. If there are several bodies, it chooses one of them at random. After the ant moves at least one step away, it drops its corpse if there are one or more corpses in its eight neighboring cells. The ant drops the corpse in a randomly chosen empty neighboring cell. The authors point out that the simple ants can do the corpse clustering as well. The only difference is that they are slower compared to the intelligent ants.

Vizine et al. [67] develop the Adaptive Ant Clustering Algorithm (A^2CA), which is an extension of the work by Lumer and Faieta [47]. There are two main modifications. Firstly, the vision field is changed adaptively according to a density dependent function. This is called progressive vision. Secondly, a pheromone function is used for each grid. Positions with high pheromone values are attractive for the ants to drop off the points. In contrast, the pick-up probability is inversely proportional to the amount of pheromone at that position. Yang and Kamel [74] propose an aggregated clustering approach. There are several parallel and independent ant colonies, and a queen ant agent. Firstly, clusters are formed by each colony. Secondly, the clustering results are combined by the queen ant using a hypergraph model. Finally, the new similarity matrix calculated by the queen ant is shared with the ant colonies.

Handl et al. [29] also modify the work in [47] with three features: (1) a short-term memory with look ahead capability, that is the next position of the ant is the grid cell with the best match (highest neighborhood function value), (2) increasing the size of the local neighborhood during the algorithm, and (3) a modified neighborhood function and threshold functions. An agglomerative clustering algorithm is used to retrieve the final clusters, and the size of the local neighborhood in the last phase of the ABS becomes the stopping condition for the cluster retrieval process. Hence, the proposed approach finds the number of clusters automatically.

Boryczka [9] propose two revised versions of the algorithm in [47], namely ACA and ACAM. The neighborhood function of ACA depends on the size of the local neighborhood and a dissimilarity scaling parameter (α). Parameter α determines the percentage of points on the grid that are classified as similar, and it is adaptively changed according to the rate of failure for drop-off. In ACAM a new neighborhood scaling parameter is defined using the relationship between the initial and the current size of the local neighborhood for an ant. Also, a cooling procedure is applied for parameter α, where it changes according to the average behavior of the density dependent function over the iterations. Gunes and Uyar [26] propose the parallelized version of ABS. Although clustering quality does not improve, the proposed approach works faster than the traditional ABS.

A recent study by Elkamel et al. [18] introduces a novel ABS algorithm, namely Communicating Ants for Clustering with Backtracking (CACB). CACB is a hierarchical clustering algorithm. It applies backtracking strategy so that the clustering errors in the previous iterations are corrected. Moreover, the use of dynamic aggregation threshold and dynamic data structure with doubly linked list improves the execution time of the algorithm.

In hybrid approaches, [62, 64] use the search agent representation. Tsai et al. [64] hybridize ACO with SA and tournament selection from GA. Ants generate subtours by inserting edges between data points. Pheromone concentration is updated for each edge connecting two points. The closer the distance between the two points is, the higher the pheromone deposit is. An ant selects the next point to visit using the tournament selection. SA ensures the reduction of number of points visited by each ant over the iterations. In the first phase of the algorithm, the edges between similar points become denser in terms of pheromone concentration. In the second phase, edges having higher pheromone density are set as dense, and they are merged using a hierarchical clustering algorithm to form the final clusters. The proposed approach in [62] is similar to the one in [64] except the use of tabu search. During the solution construction, an ant cannot visit the same point repeatedly. This restriction is implicitly imposed in [64] as well.

In the OSIB category, [44] introduce a clustering algorithm based on the API metaheuristic, which is inspired from the prey model of Pachycondyla apicalis ants. Ants generate random points around the nest, i.e. their initial hunting sites. Each ant visits its hunting site, and performs a local search. This is repeated until the ant does not find a prey. Then, the ant starts with another random point close to the nest.

In most of the algorithms, search agent representation helps find the number of clusters. Also, arbitrary-shaped clusters can be extracted. However, some of the algorithms require a cluster retrieval process that runs in isolation from the SI-based algorithm.

5 Discussion

The PSO, ACO, ABS, hybrid, and other SI-based algorithms reviewed in Sect. 4 are summarized in Appendix Tables 6, 7, 8, 9, and 10, respectively. The tables include the features of each study concerning the challenging issues in the clustering problem as discussed in Sect. 2, clustering objectives, agent representation, and application area if any.

In this section we provide a brief discussion of the review results from two perspectives: the agent representation and the challenging issues in clustering.

5.1 *Agent Representation Versus SI-Based Clustering Algorithms*

We summarize the frequency of studies discussed so far in Table 4 according to the agent representation and the type of the SI algorithm. Cluster representatives scheme is the most widely used agent representation in PSO. As PSO has originally been proposed for continuous optimization problems, it becomes a perfect fit for searching the cluster representatives.

An interesting observation is that, there is a single study that uses the data point-to-cluster assignment representation scheme. As this scheme works in the discrete domain, combinatorial PSO is adopted for the clustering problem. The PSO studies discussed so far do not use the search agent representation.

Data point-to-cluster assignment and search agent representations are applied in ACO. ACO is a construction metaheuristic where ants construct a solution from scratch at every iteration. However, cluster representatives and direct point-agent matching representation schemes tend to facilitate solution improvement rather than solution construction. For this reason, these representations have not been used in ACO so far.

In ABS, direct point-agent matching and search agent are the most popular representations. Clusters in ABS are formed using the collective behavior of ants,

Table 4 The frequency of articles by agent representation and SI algorithm

	Agent representation				Total
	Data point-to-cluster assignment	Cluster representatives	Direct point-agent matching	Search agent	
PSO	1	12	2	0	15
ACO	4	0	0	1	5
ABS	0	0	4	8	12
Hybrid	0	7	0	2	9
OSIB	1	6	2	1	10

each carrying or visiting individual data points. These two representation schemes are also based on the individual characteristics of the agents. Hence, they offer a good fit for the gathering and sorting activities of ABS. On the other hand, data point-to-cluster assignment and cluster representatives carry the information about the clusters rather than the points. Hence, they are not as suitable for the working principles of ABS.

Cluster representatives and search agent are the two representation schemes used in hybrid approaches. The cluster representatives scheme takes a partitional approach to the clustering problem, and it is often used together with minimization of the within cluster variance/distance or optimization of a validity index. In order to avoid premature convergence, the strong features of the other metaheuristics and AI techniques are combined with SI. In a similar manner, studies using search agent benefit from the exploration and exploitation capabilities of the other metaheuristics.

In the OSIB algorithms, all four types of representations are used. The most popular one is the cluster representatives scheme. In particular, cluster representatives are used in ABC and its variants, and the firefly algorithm.

5.2 Agent Representation Versus Challenging Issues in Clustering

In this section, we discuss which challenging characteristics of the clustering problem are addressed by each representation scheme. Use of the agent representation schemes in handling the challenging issues in clustering is overviewed in Table 5. The following observations can be made according to Table 5.

(a) *Multiple objectives*

- Most of the literature focuses on a single objective. In particular, the compactness objective, which minimizes the within cluster variance/distance, is widely used. Unless a transformation is applied to the data set, these

Table 5 Use of agent representations in handling the challenging issues in clustering

	Agent Representation			
Clustering characteristic	Data point-to-cluster assignment	Cluster representatives	Direct point-agent matching	Search agent
Multiple objectives		✓		✓
Unknown number of clusters		✓	✓	✓
Arbitrary-shaped clusters		✓	✓	✓
Data type		✓	✓	
Constraints			✓	
Scalability		✓	✓	✓

approaches generate spherical shapes regardless of the representation scheme.

- An early study considers the weighted sum of the two conflicting objectives, namely compactness and separation [53]. Cluster representatives are optimized using PSO. Although this study addresses multiple objectives, a multi-objective optimization method is not used.
- A recent study considers two objectives, namely adjusted compactness and relative separation, separately [33]. In ACO ants are search agents, and they insert edges between pairs of similar points. The proposed approach outputs a set of non-dominated solutions. However, it does not guarantee to generate the entire Pareto-efficient frontier.

(b) *Unknown number of clusters*

- The number of clusters can be extracted in the studies that use the direct point-agent matching and search agent representations. Some of the algorithms in this category form a tree or a map that illustrates the relative position of each data point. An additional cluster retrieval algorithm might be needed to determine the number of clusters. Hierarchical algorithms with thresholds of minimum similarity or maximum dissimilarity are often used for this purpose [29]. This limits the role of SI in finding the final clusters.
- Only a few studies using cluster representatives can extract the number of clusters. In these studies, the maximum number of clusters should be given, and an additional solution vector or variable is defined to control the active cluster representatives [12, 14, 54].

(c) *Arbitrary-shaped clusters*

- Direct point-agent matching and search agent representation schemes do not require any a priori information about the number of clusters or shapes of the clusters. They are flexible tools for extracting the relations among data points. Hence, they can find the arbitrary-shaped clusters, if they are coupled with proper objective(s) and a cluster retrieval process.
- When the agents denote the cluster representatives, and the objective is to minimize the total distance between cluster representatives and data points, the algorithm generates spherical clusters. This deficiency is resolved by the use of kernel-based similarity functions [14].
- In direct point-agent matching, kernel-based similarity measures are also used to extract the arbitrary-shaped clusters [76, 77]. In these algorithms, a cluster retrieval process is executed after the SI algorithm in order to determine the clusters.

(d) *Data type*

- Only a limited number of studies focus on the data sets with non-numerical attributes. For the binary and categorical attributes, Hamming distance is used together with direct agent-point matching [4].

- A couple of studies consider text and document clustering. They use cluster representatives, and apply extended Jaccard coefficient [46] and cosine distance functions [63] to measure the dissimilarities.
- SI-based algorithms are applied to the image and MRI data sets [53, 54]. However, these studies also use Euclidean distance, as the attributes are numerical.

(e) *Constraints*

- The constraint handling mechanisms in SI-based algorithms are limited. To the best of our knowledge, there is only one study that incorporates ML and CL constraints into ABS [71]. They use direct point-agent matching.

(f) *Scalability*

- The most significant work in this category is the parallelization of the ABS algorithm [26].
- Kernel-based approaches produce the rough clusters, so they reduce the runtime of the algorithm [14, 76, 77]. This approach is used in PSO and ABS so far.
- Another approach to handle the scalability is the execution of a preprocessing step for the subclusters [33]. As the subclusters include connected points, only the points on the boundaries of the subclusters are considered for merging and outlier detection operations. A boundary formation algorithm is used to exclude the interior points in the subclusters. The proposed clustering algorithm is executed for the boundary points of the subclusters only.

6 Conclusion

In this study we provide a systematic review for the SI-based hard (crisp) clustering algorithms. Our review includes PSO, ACO, ABS, hybrid, and other SI-based algorithms. Agent representation is the key element for the design of an SI-based algorithm. Motivated by this, we classify the SI-based clustering algorithms into four categories according to agent representation. We examine the capabilities of each representation scheme in handling the challenging characteristics of the clustering problem including multiple objectives, unknown number of clusters, arbitrary-shaped clusters, data types, constraints, and scalability.

As agent representation, search agent and direct point-agent matching schemes can be used when there is no a priori information about the clustering problem. When the data set includes spherical clusters and the number of clusters is known a priori, data point-to-cluster assignment and cluster representatives schemes are candidates for agent representation.

Not all representation schemes are applied to all categories of SI-based algorithms. For instance, ACO works in discrete search space, whereas direct point-agent matching works in continuous space. ABS is based on the moves of ants in the search space, so representation schemes that carry cluster information are not used in ABS. The search agent constructs a solution, whereas PSO is an improvement metaheuristic. Still, the use of different representation schemes in different types of SI-based algorithms is a promising future study. Such adoptions may lead to more effective and efficient search mechanisms for the SI-based algorithms.

Furthermore, SI researchers could pay more attention to the challenging characteristics of the clustering problem. Future research directions can be as follows:

- Current SI-based algorithms use different similarity measures for mixed data types. The development of SI-based mechanisms to handle various data types can be studied further.
- SI-based algorithms for semi-supervised clustering can be a future research direction.
- The development of multi-objective SI-based algorithms for the clustering problem can be a future research direction. In particular, the extraction of Pareto-efficient frontier needs to be explored.
- Another important issue with the real-life clustering problems is the scalability. Hence, scalability should be taken into account during the SI-algorithm design.

Appendix

See Tables 6, 7, 8, 9, and 10.

Table 6 Characteristics of the PSO algorithms for clustering

Article	Problem characteristics					Clustering objective	Particle representation	Application area
	Multi-objective	# of clusters	Arbitrary shapes	Data type	Constraints			
[36]	No	Given	No	Numerical	No	– Minimize within cluster variation (sum of squared Euclidean distances from cluster centers) – Minimize variance ratio	Data point-to-cluster assignment: A particle shows cluster assignments of all points	–
[12]	Yes	Not given	Yes	Numerical	No	Minimize the difference between total intra-cluster distances and minimum inter-cluster distance	Cluster representatives: A particle represents the number of clusters and cluster centers	–
[14]	No	Not given	Yes	Numerical	No	Maximize kernelized CS measure	Cluster representatives: A particle represents the activation function values and cluster centers	–

(continued)

Table 6 (continued)

Article	Problem characteristics					Clustering objective	Particle representation	Application area
	Multi-objective	# of clusters	Arbitrary shapes	Data type	Constraints			
[39]	No	Given	No	Numerical	No	Minimize within cluster distances (sum of Euclidean distances from cluster centers)	Cluster representatives: Cluster centroids are represented by a particle	–
[45]	No	Given	No	Numerical	No	Minimize within cluster variation (sum of squared Euclidean distances from cluster centers)	Cluster representatives: Cluster centroids are represented by a particle	Clustering production orders of a company
[46]	No	Given	No	Numerical and text	No	Minimize within cluster distance	Cluster representatives: Position and velocity particles denote cluster weight sets. Third particle represents cluster centroids	Text clustering
[52]	Yes	Given	Yes	Numerical	No	– Minimize maximum average within cluster distance – Maximize minimum inter-cluster distance	Cluster representatives: Cluster centroids are represented by a particle	MRI and satellite

[53]	Yes	Given	Yes	Numerical	No	– Minimize maximum average within cluster distance – Maximize minimum inter-cluster distance – Minimize quantization error	Cluster representatives: Cluster centroids are represented by a particle	MRI and satellite
[54]	No	Not given	No	Numerical	No	Improve validity index	Cluster representatives: Binary particle denotes whether a cluster centroid is used or not	MRI and satellite
[55]	No	Given	No	Numerical	No	– Minimize trace within criterion – Minimize variance ratio – Minimize Marriott's criterion	Cluster representatives: Cluster centroids are represented by a particle	–
[63]	No	Given	No	Document	No	Minimize average distance between the documents and their closest cluster centroids	Cluster representatives: Cluster centroids are represented by a particle	Document

(continued)

Table 6 (continued)

Article	Problem characteristics					Clustering objective	Particle representation	Application area
	Multi-objective	# of clusters	Arbitrary shapes	Data type	Constraints			
[65]	No	Given	No	Numerical	No	Minimize quantization error	Cluster representatives: Cluster centroids are represented by a particle	–
[73]	No	Given	No	Numerical	No	Minimize harmonic mean of distances from data points to all centers	Cluster representatives: Cluster centroids are represented by a particle	–
[56]	No	Not given	Yes	Numerical	No	Maximize within cluster similarity	Direct point-agent matching: Each point is represented by a particle. Position of particle in two-dimensional space denotes position of point	Facial skin database for women
[66]	No	Not given	Yes	Numerical	No	Maximize within cluster similarity	Direct point-agent matching: Each point is represented by a particle (datoid). Position of particle in two-dimensional space denotes position of datoid	–

Table 7 Characteristics of the ACO algorithms for clustering

Article	Problem characteristics					Clustering objective	Ant representation	Application area
	Multi-objective	# of clusters	Arbitrary shapes	Data type	Constraints			
[11]	No	Given	No	Numerical	No	Minimize within cluster variation (sum of squared Euclidean distances from cluster centers)	Data point-to-cluster assignment: An ant shows cluster assignments of all points	Electronic library
[38]	No	Given	No	Numerical	No	Minimize average quantization error	Data point-to-cluster assignment: An ant shows cluster assignments and selected attributes	–
[58]	No	Given	No	Numerical	No	Minimize within cluster variation (sum of squared Euclidean distances from cluster centers)	Data point-to-cluster assignment: An ant represents cluster assignment of a point	–
[61]	No	Given	No	Numerical	No	Minimize the sum of squared Euclidean distances between each object and the center	Data point-to-cluster assignment: An ant represents cluster assignments of points	–
[33]	Yes	Not given	Yes	Numerical	No	– Adjusted compactness – Relative separation	Search agent: An ant connects pairs of points as it moves	–

Table 8 Characteristics of the ABS algorithms for clustering

Article	Problem characteristics					Clustering objective	Particle representation	Application area
	Multi-objective	# of clusters	Arbitrary shapes	Data type	Constraints			
[22]	No	Given	Yes	Numerical	No	Maximize within cluster similarity and average linkage (implicit)	Direct point-agent matching: Each point is represented by an ant	–
[71]	No	Given	Yes	Numerical	ML and CL	Maximize within cluster similarity (implicit)	Direct point-agent matching: Each point is represented by an ant	–
[76]	No	Not given	Yes	Numerical	No	Minimize within cluster distances (implicit)	Direct point-agent matching: Each point is represented by an ant	–
[77]	No	Not given	Yes	Numerical	No	Maximize within cluster similarity	Direct point-agent matching: Each point is represented by an ant	–
[9]	No	Not given	Yes	Numerical	No	Maximize within cluster similarity (implicit)	Search agent: An ant carries data points	–
[18]	No	Not given	Yes	Numerical	No	Maximize within cluster similarity (implicit)	Search agent: An ant carries data points	Image indexing
[26]	No	Not given	Yes	Numerical	No	Maximize within cluster similarity (Algorithm is adopted from Handl et al. 2006)	Search agent: An ant searches through data points	–

[29]	No	Not given	Yes	Mixed	No	Maximize within cluster similarity (implicit)	Search agent: An ant searches through solution components (points)	–
[47]	No	Not given	Yes	Numerical	No	Minimize within cluster distances (implicit)	Search agent: An ant carries data points	–
[49]	No	Not given	Yes	Numerical	No	Cemetery formation (corpse clustering)	Search agent: An ant carries corpses (points)	–
[67]	No	Not given	Yes	Numerical	No	Maximize within cluster similarity (implicit)	Search agent: An ant carries data points	–
[74]	No	Given	Yes	Numerical	No	Maximize within cluster similarity	Search agent: An ant searches through solution components (points)	–

Table 9 Characteristics of the hybrid algorithms for clustering

Article	Problem characteristics					Clustering objective	Particle representation	Application area
	Multi-objective	# of clusters	Arbitrary shapes	Data type	Constraints			
[21]	No	Given	No	Numerical	No	Minimize within cluster variation (sum of squared Euclidean distances from cluster centers)	Cluster representatives: Each agent (ant/particle) represents the cluster centers	Market segmentation
[31]	No	Given	No	Numerical	No	Minimize total intra-cluster distance to total inter-cluster distance ratio	Cluster representatives: Each agent (ant/particle) represents the cluster centers	–
[37]	No	Given	No	Numerical	No	Minimize within cluster distance (sum of Euclidean distances from cluster centroids)	Cluster representatives: Cluster centroids are represented by a particle	–
[51]	No	Given	No	Numerical	No	Minimize total intra-cluster variance (total mean squared quantization error)	Cluster representatives: Cluster centers are represented by a particle	Market segmentation
[69]	No	Not given	Yes	Mixed	No	Maximize within cluster similarity	Cluster representatives: A particle denotes the complete weight set of SOM	–

[70]	No	Not given	Yes	Numerical	No	Improve cluster validity indices including CH, DB, Dunn and its variants, CS, I, and Silhouette	Cluster representatives: A particle represents the activation function values and cluster centers	–
[72]	No	Given	No	Numerical	No	Minimize total intra-cluster variance (total mean squared quantization error)	Cluster representatives: Cluster centers are represented by a particle	–
[62]	No	Not given	Yes	Numerical	No	Minimize within cluster distances	Search agent: An ant searches through solution components (points)	–
[64]	No	Not given	Yes	Numerical	No	Minimize within cluster distances	Search agent: An ant searches through solution components (points)	Credit card data

Table 10 Characteristics of the other SI-based algorithms for clustering

Article	Problem characteristics					Clustering objective	Particle representation	Application area
	Multi-objective	# of clusters	Arbitrary shapes	Data type	Constraints			
[59]	No	Given	No	Numerical	No	Minimize within cluster variation (sum of squared Euclidean distances from cluster centers)	Data point-to-cluster assignment: A particle denotes cluster assignment of a point	–
[20]	No	Given	No	Numerical	No	Minimize within cluster variation (sum of squared Euclidean distances from cluster centers)	Cluster representatives: Cluster centroids are represented by queen and drones	–
[41]	No	Given	No	Numerical	No	Minimize within cluster distance	Cluster representatives: A bee (employed and onlooker) represents cluster centers (food source)	–
[60]	No	Given	No	Numerical	No	Minimize within cluster variance (SSE)	Cluster representatives: Particles represent cluster centers	–
[68]	No	Given	No	Numerical	No	Minimize within cluster variance (SSE)	Cluster representatives: Particles represent cluster centers	–
[75]	No	Given	No	Numerical	No	Minimize sum of intra-cluster distances	Cluster representatives: A bee (employed and onlooker) represents cluster centers (food source)	–

[78]	No	Given	No	Numerical	No	Minimize sum of intra-cluster variance	Cluster representatives: A bee (employed and onlooker) represents cluster centers (food source)	–
[4]	No	Not given	Yes	Mixed	No	Build a tree with maximum similar roots and sufficiently dissimilar branches	Direct point-agent matching: An ant represents a point	Human skin analysis, web usage mining, portal site
[43]	No	Given	Yes	Numerical	No	Maximize within cluster similarity	Direct point-agent matching: Each point is represented by a particle. Position of particle in two-dimensional space denotes position of point	–
[44]	No	Given	Yes	Numerical	No	Maximize within cluster similarity	Search agent: An ant searches through data points	–

References

1. Abraham, A., Das, S., Roy, S.: Swarm Intelligence Algorithms for Data Clustering. In: Maimon, O., Rokach, L. (eds.) Soft Computing for Knowledge Discovery and Data Mining, pp. 279–313. Springer, New York (2008)
2. Alam, S., Dobbie, G., Koh, Y.S., Riddle, P., Rehman, S.U.: Research on particle swarm optimization based clustering: a systematic review of literature and techniques. Swarm Evol. Comput. **17**, 1–13 (2014)
3. Alani, H., Jones, C.B., Tudhope, D.: Voronoi-based region approximation for geographical information retrieval with gazetteers. Int. J. Geogr. Inf. Sci. **15**(4), 287–306 (2001)
4. Azzag, H., Venturini, G., Oliver, A., Guinot, C.: A hierarchical ant-based clustering algorithm and its use in three real-world applications. Eur. J. Oper. Res. **179**(3), 906–922 (2007)
5. Bandyopadhyay, S., Saha, S.: GAPS: A clustering method using a new point symmetry-based distance measure. Pattern Recogn. **40**, 3430–3451 (2007)
6. Bandyopadhyay, S., Saha, S.: A point symmetry-based clustering technique for automatic evolution of clusters. IEEE Trans. Knowl. Data Eng. **20**(11), 1441–1457 (2008)
7. Basu, S., Davidson, I.: KDD 2006 Tutorial Clustering with Constraints: Theory and Practice (2006). Available via http://www.ai.sri.com/~basu/kdd-tutorial-2006.Cited2March2015
8. Bong, C.-W., Rajeswari, M.: Multi-objective nature-inspired clustering and classification techniques for image segmentation. Appl. Soft Comput. **11**(4), 3271–3282 (2011)
9. Boryczka, U.: Finding groups in data: Cluster analysis with ants. Appl. Soft Comput. **9**(1), 61–70 (2009)
10. Celebi, M.E.: Partitional Clustering Algorithms. Springer, Switzerland (2015)
11. Chen, A., Chen, C.: A new efficient approach for data clustering in electronic library using ant colony clustering algorithm. Electron. Libr. **24**(4), 548–559 (2006)
12. Cura, T.: A particle swarm optimization approach to clustering. Expert Syst. Appl. **39**(1), 1582–1588 (2012)
13. D'Urso, P., De Giovanni, L., Disegna, M., Massari, R.: Bagged clustering and its application to tourism market segmentation. Expert Syst. Appl. **40**(12), 4944–4956 (2013)
14. Das, S., Abraham, A., Konar, A.: Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm. Pattern Recogn. Lett. **29**(5), 688–699 (2008)
15. Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chržtien, L.: The dynamics of collective sorting: robot-like ants and ant-like robots. In: Meyer, J.A., Wilson, S. (eds.) From Animals to Animats: Proceedings of the 1st International Conference on Simulation of Adaptive Behavior, pp. 356–365. Cambridge, MIT Press (2008)
16. Dorigo, M., Maniezzo, V., Colorni, A.: Positive feedback as a search strategy. Technical Report 91016 Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italy (1991)
17. Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. B Cybern. **26**(1), 29–41 (1996)
18. Elkamel, A., Gzara, M., Ben-Abdallah, H.: A bio-inspired hierarchical clustering algorithm with backtracking strategy. Appl. Intell. **42**, 174–194 (2015)
19. Ester, M., Kriegel, K.P., Sander J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
20. Fathian, M., Amiri, B., Maroosi, A.: Application of honey-bee mating optimization algorithm on clustering. Appl. Math. Comput. **190**(2), 1502–1513 (2007)
21. Firouzi, B.B., Sadeghi, M.S., Niknam, T.: A new hybrid algorithm based on PSO, SA and *k*-means for cluster analysis. Int. J. Innovative Comput. Inf. Control **6**(7), 3177–3192 (2010)
22. Ghosh, A., Halder, A., Kothari, M., Ghosh, S.: Aggregation pheromone density based data clustering. Inf. Sci. **178**(13), 2816–2831 (2008)
23. Gower, J.: Coefficients of association and similarity, based on binary (presence-absence) data: an evaluation. Biometrics **27**, 857–871 (1971)

24. Grosan, C., Abraham, A., Chis, M.: Swarm Intelligence in Data Mining. In: Abraham, A., Grosan, C., Ramos, V. (eds.) Swarm Intelligence in Data Mining, pp. 1–20. Springer, Berlin (2006)
25. Guha, S., Rastogi, R., Shim, K.: CURE: An efficient clustering algorithm for large databases. In: Proceedings of the ACM-SIGMOD International Conference on Management of Data, pp. 73–84 (1998)
26. Gunes, O.G., Uyar, A.S.: Parallelization of an ant-based clustering approach. Kybernetes **39**(4), 656–677 (2010)
27. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufman, Massachusetts (2011)
28. Handl, J., Meyer, B.: Ant-based and swarm-based clustering. Swarm Intell. **1**, 95–113 (2007)
29. Handl, J., Knowles, J., Dorigo, M.: Ant-based clustering and topographic mapping. Artif. Life **12**, 35–61 (2006)
30. Hruschka, E.R., Campello, R.J.G.B., de Castro, L.N.: Evolving clusters in gene-expression data. Inf. Sci. **176**(13), 1898–1927 (2006)
31. Huang, C.L., Huang, W.C., Chang, H.Y., Yeh, Y.C., Tsai, C.Y.: Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering. Appl. Soft Comput. **13**(9), 3864–3872 (2013)
32. Ichino, M., Yaguchi, H.: Generalized Minkowski metrics for mixed feature-type data analysis. IEEE Trans. Syst. Man Cybern. **24**(4), 698–708 (1994)
33. İnkaya, T., Kayalıgil, S., Özdemirel, N.E.: Ant colony optimization based clustering methodology. Appl. Soft Comput. **28**, 301–311 (2015)
34. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. **31**, 651–666 (2010)
35. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**(3), 264–323 (1999)
36. Jarboui, B., Cheikh, M., Siarry, P., Rebai, A.: Combinatorial particle swarm optimization for partitional clustering problem. Appl. Math. Comput. **192**, 337–345 (2007)
37. Jiang, B., Wang, N.: Cooperative bare-bone particle swarm optimization for data clustering. Soft Comput. **18**, 1079–1091 (2014)
38. Jiang, L., Ding, L., Peng, Y.: An efficient clustering approach using ant colony algorithm in multidimensional search space. In: Proceedings of the 8th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 1085–1089 (2011)
39. Kao, Y.T., Zahara, E., Kao, I.W.: A hybridized approach to data clustering. Expert Syst. Appl. **34**(3), 1754–1762 (2008)
40. Karaboga, D.: An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
41. Karaboga, D., Ozturk, C.: A novel clustering approach: Artificial Bee Colony (ABC) algorithm. Appl. Soft Comput. **11**(1), 652–657 (2011)
42. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Network, vol. 4, pp. 1942–1948 (1995)
43. Khereddine, B., Gzara, M.: FDClust: a new bio-inspired divisive clustering algorithm. Advances in Swarm Intelligence. Lecture Notes in Computer Science, vol. 6729, pp. 136–145. Springer, Berlin (2011)
44. Kountche, D.A., Monmarche, N., Slimane, M.: The Pachycondyla Apicalis ants search strategy for data clustering problems. Swarm and Evolutionary Computation. Lecture Notes in Computer Science, vol. 7269, pp. 3–11. Springer, Berlin (2012)
45. Kuo, R.J., Wang, M.J., Huang, T.W.: An application of particle swarm optimization algorithm to clustering analysis. Soft Comput. **15**, 533–542 (2011)
46. Lu, Y., Wang, S., Li, S., Zhou, C.: Particle swarm optimizer for variable weighting in clustering high-dimensional data. Mach. Learn. **82**, 43–70 (2011)
47. Lumer, E.D., Faieta, B.: Diversity and adaptation in populations of clustering ants. In: Cliff, D., Husbands, P., Meyer, J.A., Wilson, S.W. (eds.) Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior: From Animals to Animats, pp. 501–508. MIT Press/Bradford Books, Cambridge (1994)

48. Martens, D., Baesens, B., Fawcett, T.: Editorial survey: swarm intelligence for data mining. Mach. Learn. **82**, 1–42 (2011)
49. Martin, M., Chopard, B., Albquerque, P.: Formation of an ant cemetry: swarm intelligence or statistical accident? Futur. Gener. Comput. Syst. **18**(7), 951–959 (2002)
50. Nanda, S. J., Panda, G.: A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol. Comput. **16**, 1–18 (2014)
51. Niknam, T., Amiri, B.: An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. Appl. Soft Comput. **10**(1), 183–197 (2010)
52. Omran, M., Salman, A., Engelbrecht, A.P.: Image classification using particle swarm optimization. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, pp. 18–22 (2002)
53. Omran, M., Engelbrecht, A.P., Salman, A.: Particle swarm optimization method for image clustering. Int. J. Pattern Recognit. Artif. Intell. **19**(3), 297–321 (2005)
54. Omran, M.G.H., Salman, A., Engelbrecht, A.P.: Dynamic clustering using particle swarm optimization with application in image segmentation. Pattern. Anal. Appl. **8**, 332–344 (2006)
55. Paterlini, S., Krink, T.: Differential evolution and particle swarm optimisation in partitional clustering. Comput. Stat. Data Anal. **50**, 1220–1247 (2006)
56. Picarougne, F., Azzag, H., Venturini, G., Guinot, C.: A new approach of data clustering using a flock of agents. Evol. Comput. **15**(3), 345–367 (2007)
57. Rana, S., Jasola, S., Kumar, R.: A review on particle swarm optimization algorithms and their applications to data clustering. Artif. Intell. Rev. **35**, 211–222 (2011)
58. Runkler, T.A.: Ant colony optimization in clustering models. Int. J. Intell. Syst. **20**, 1233–1251 (2005)
59. Runkler, T.A.: Wasp swarm optimization of the c-means clustering model. Int. J. Intell. Syst. **23**, 269–285 (2008)
60. Senthilnath, J., Omkar, S.N., Mani, V.: Clustering using firefly algorithm: performance study. Swarm Evol. Comput. **1**, 164–171 (2011)
61. Shelokar, P.S., Jayaraman, V.K., Kulkarni, B.D.: An ant colony approach for clustering. Anal. Chim. Acta **509**, 187–195 (2004)
62. Sinha, A.N., Das, N., Sahoo, G.: Ant colony based hybrid optimization for data clustering. Kybernetes **36**(2), 175–191 (2007)
63. Song, W., Ma, W., Qiao, Y.: Particle swarm optimization algorithm with environmental factors for clustering analysis. Soft Comput. (2014). doi: 10.1007/s00500-014-1458-7
64. Tsai, C., Tsai, C., Wu, H., Yang, T.: ACODF: a novel data clustering approach for data mining in large databases. J. Syst. Softw. **73**, 133–145 (2004)
65. Van der Merwe, D.W., Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: Proceedings of the 2003 Congress on Evolutionary Computation, pp. 215–220 (2003)
66. Veenhuis, C., Köppen, M.: Data swarm clustering. In: Abraham, A., Grosan, C., Ramos, V. (eds.) Swarm Intelligence in Data Mining, pp. 221–241. Springer, Berlin (2006)
67. Vizine, A.L., De Castro, L.N., Hruschka, E.R., Gudwin, R.R.: Towards improving clustering ants: an adaptive ant clustering algorithm. Informatica **29**, 143–154 (2005)
68. Wan, M., Wang, C., Li, L., Yang, Y.: Chaotic ant swarm approach for data clustering. Appl. Soft Comput. **12**(8), 2387–2393 (2012)
69. Xiao, X., Dow, E.R., Eberhart, R., Miled, Z.B., Oppelt, R.J.: A hybrid self-organizing maps and particle swarm optimization approach. Concurrency Comput. Pract. Exp. **16**, 895–915 (2004)
70. Xu, R., Xu, J., Wunsch, D.C.: A comparison study of validity indices on swarm-intelligence-based clustering. IEEE Trans. Syst. Man Cybern. B Cybern. **12**(4), 1243–1256 (2012)
71. Xu, X., Lu, L., He, P., Pan, Z., Chen, L.: Improving constrained clustering via swarm intelligence. Neurocomputing **116**, 317–325 (2013)
72. Yan, X., Zhu, Y., Zou, W., Wang, L.: A new approach for data clustering using hybrid artificial bee colony. Neurocomputing **97**, 241–25 (2012)
73. Yang, F., Sun, T., Zhang, C.: An efficient hybrid data clustering method based on k-harmonic means and particle swarm optimization. Expert Syst. Appl. **36**(6), 9847–9852 (2009)

74. Yang, Y., Kamel, M.S.: An aggregated clustering approach using multi-ant colonies algorithms. Pattern Recogn. **39**(7), 1278–1289 (2006)
75. Zhang, C., Ouyang, D., Ning, J.: An artificial bee colony approach for clustering. Expert Syst. Appl. **37**(7), 4761–4767 (2010)
76. Zhang, L., Cao, Q.: A novel ant-based clustering algorithm using the kernel method. Inf. Sci. **181**(20), 4658–4672 (2011)
77. Zhang, L., Cao, Q., Lee, J.: A novel ant-based clustering algorithm using Renyi entropy. Appl. Soft Comput. **13**(5), 2643–2657 (2013)
78. Zou, W., Zhu, Y., Chen, H., Sui, X.: A clustering approach using cooperative artificial bee colony algorithm. Discret. Dyn. Nat. Soc. **459796**, 1–16 (2010)