Swarm Intell (2007) 1: 95–113
DOI 10.1007/s11721-007-0008-7

# Ant-based and swarm-based clustering

**Julia Handl · Bernd Meyer**

Received: 20 December 2006 / Accepted: 5 October 2007 / Published online: 13 November 2007


**Abstract** Clustering with swarm-based algorithms is emerging as an alternative to more conventional clustering methods, such as hierarchical clustering and $k$-means. Ant-based clustering stands out as the most widely used group of swarm-based clustering algorithms. Broadly speaking, there are two main types of ant-based clustering: the first group of methods directly mimics the clustering behavior observed in real ant colonies. The second group is less directly inspired by nature: the clustering task is reformulated as an optimization task and general purpose ant-based optimization heuristics are utilized to find good or near-optimal clusterings. This papers reviews both approaches and places these methods in the wider context of general swarm-based clustering approaches.

**Keywords** Ant-based clustering · Swarm-based clustering · Ant colony optimization · Particle swarm optimization · Clustering · Data-mining

## 1 Introduction

Swarm intelligence (SI) is an artificial intelligence paradigm based on the study of emergent behavior in decentralized, self-organized systems. SI methods aim to imitate such behavior and apply it to the solution of hard computational problems. The hope behind such algorithmic approaches is for them to inherit the simple design, scalability and robustness of their naturally occurring counterparts.

Arguably, the best known example of self-organizing decision making in nature are the mechanisms that govern foraging in ant colonies. These have served as the inspiration for one particular type of stochastic metaheuristic for combinatorial optimization, ant colony optimization (Dorigo and Stützle 2004). Other prominent examples of collective emergent

J. Handl (✉)
Manchester Interdisciplinary Biocentre, University of Manchester, Manchester, UK
e-mail: J.Handl@postgrad.manchester.ac.uk

B. Meyer
Clayton School of IT, Monash University, Melbourne, Australia

behavior in nature that have served as an inspiration to solve complex computational problems include mechanisms for foraging and division of labor in honey bee swarms and ant colonies, which have been used in algorithms for production facility control (Campos et al. 2001), and the control of collective movement in flocks of birds and schools of fish (Reynolds 1987), which is imitated in collaborative robotics (Bonabeau et al. 1999) and particle swarm optimization (Kennedy and Eberhart 1995).

This review paper focuses on data clustering and primarily on ant-inspired approaches to this challenging problem. Clustering, that is, the "identification of homogeneous groups of objects" (Arabie et al. 1996), is one of the core processes in data mining and central to many other applications in the sciences, as well as in commercial and economic arenas: Grouping objects based on common properties is a natural precursor of deriving a full taxonomy and is thus often used as a first step towards building hypotheses about the underlying structure of some data collection or the processes that have generated particular data (Jain et al. 1999b; Jain and Dubes 1988; Everitt et al. 2001; Theoridis and Koutrumbas 2006).

Clustering is usually performed as an *unsupervised* classification task. This means that the types and characteristics of clusters are unknown and have to be discovered in the clustering process. Semi-supervised clustering is different from unsupervised clustering in that it uses explicit partial information on cluster membership as input to the clustering algorithm. While there is increasing interest in semi-supervised clustering in the context of classical methods (Bilenko et al. 2004), this problem has to the best of the authors' knowledge not been investigated with ant-based methods. This survey thus focuses on unsupervised clustering.

While some unsupervised algorithms require at least the number of different clusters to be specified, others also attempt to automatically determine this. Classical clustering methods include deterministic algorithms, such as hierarchical clustering (Vorhees 1985) and spectral methods (Ng et al. 2001; Kannan et al. 2004), as well as randomized methods, such as $k$-means (MacQueen 1967) and Kohonen networks (Kohonen 1997). For a comprehensive overview of different types of clustering algorithms see (Jain et al. 1999b).

The term ant-based clustering has been used in many different contexts in the literature, and it is difficult to gain a clear sense of what the common characteristics of the proposed methods are. This is mainly due to the fact that this term has also been used for algorithms that are based on other types of natural "swarm" behavior and for which the more general term "swarm-based" would have been more appropriate. In this survey we interpret ant-based clustering quite literally as only comprising such methods that are directly modeled on the social behavior of ant colonies. We will clarify the relation to more general "swarm-based" clustering in Sect. 4.

Ant-inspired clustering approaches in this narrower sense can be roughly classified into two different main types. The first group mimics nature very directly: The gathering (or clustering) of items and, occasionally, their sorting by certain properties are activities that have been observed in nest and brood care (Deneubourg et al. 1991). This has inspired the development of ant-based clustering algorithms that directly imitate this behavior to cluster abstract data. A feature of these algorithms is that the clustering objective is implicitly defined. Implicit here means that neither the overall objective of the process (i.e., clustering) nor the types of clusters sought are defined explicitly at any point during the clustering process (some may prefer to label the clustering objective "emergent"). An assessment of the suitability of such an algorithm for the clustering of a particular type of data can therefore be very challenging.

The second group of ant-based clustering methods is less directly inspired by nature: It recasts the clustering task as an optimization task[1] and utilizes general-purpose ant-based optimization methods to compute good or near optimal clusterings. An advantage of such methods is the explicit objective function, which allows one to better understand and predict the performance of the clustering algorithm on particular types of data and to customize the clustering objective for particular tasks or types of data.

A number of ant-based clustering algorithms, however, fall in between these two categories: These are derived from ant-based general purpose optimization algorithms, but modified in such a way that an explicit global objective does not exist anymore.

The remainder of this paper is structured as follows. Section 2 briefly reviews some biological clustering and sorting processes observed in ant colonies and discusses the main algorithm derived from these with its applications. Section 3 introduces the idea of using an ant-inspired general purpose optimization heuristic, ant colony optimization, to perform clustering. This section also reviews clustering heuristics that are derivatives of ant colony optimization but cannot directly be interpreted as global optimization methods themselves anymore.

While this survey primarily focuses on algorithms that have been modeled on the principles of social behavior in ant colonies, this distinction is only based on the natural inspiration and does not always represent the ideal demarcation from a theoretical algorithmic perspective. Some other swarm-based approaches are on a fundamental level closely related to the ant-based approaches discussed here. Section 4 therefore gives a brief overview of other swarm-based clustering approaches and puts ant-based methods in context. Section 5 discusses applications and advantages of ant-based clustering, as well as important directions for future research, and Section 6 presents conclusions.

## 2 Clustering-as-emergent-phenomenon algorithms

### 2.1 Biological inspiration

Sorting and clustering behavior has been observed in real ant colonies and other social insects in two different forms: patch sorting and annular sorting (Bonabeau et al. 1999). As the name suggests, patch sorting simply arranges objects in patches (spatial neighborhoods), while annular sorting arranges items in complete or partial (concentric) rings, with characteristic spacing of items in each individual ring.

#### *2.1.1 Patch sorting*

One of the best known examples of clustering in patches is cemetery formation in *Pheidole pallidula*. It has been observed that ants of this species cluster the corpses of dead nest mates when removing them from the nest. Algorithmic models for this type of behavior have been introduced by Deneubourg et al. (1991). Their models are based on decentralized control and exclusively use memoryless local interaction between individual ants and the environment. Communication between the agents simulating ants in this model is limited to an indirect form known as stigmergy: the modification of the environment by one agent and subsequent sensing of the environment by another agent. Deneubourg et al. use a discrete Monte Carlo

[1] Note that, even for a fixed number of clusters, the resulting optimization task is NP-hard for most clustering objectives, for example, the popular sum-of-squares criterion (Garey and Johnson 1979).

model in which ants perform a random walk on a grid with periodic boundary conditions. Each ant may pick up items that it encounters during its random walk and drop them at some new location. Pick-up and drop-off are probabilistic actions, and the probability to take one of these actions is determined by the distribution of items in the agent's immediate current neighborhood, as well as the agent's state (whether it is or is not carrying an item). The more similar the items in an agent's immediate neighborhood are to the one that the agent currently carries, the more likely this item will be dropped. Conversely, if the agent does not hold any item, its probability of picking up an item will increase if this item is dissimilar to the items in its immediate neighborhood.

Deneubourg et al. validated their models experimentally and demonstrated that patch sorting can indeed be obtained using this simple model.

#### 2.1.2 *Annular sorting*

Annular sorting has been observed in *Leptothorax unifasciatus* which arrange their larvae as a function of their size (Melhuish et al. 2006; Sendova-Franks et al. 2004; Scholes and Wilson 2004). The conjectured biological utility of such sorting behavior is increased efficiency in nest care, as the brood is spatially arranged according to its nutritional and metabolic needs. Recently, some fundamental aspects of this sorting have been reproduced in computational simulations as well as in simulations using groups of simple robots (Melhuish et al. 2006; Scholes and Wilson 2004). These simulations indicate that annular sorting may rely on a two stage process: a first clustering phase, which sees the gathering of all items in one central heap, and a second spacing phase, which sees the gradual arrangement of items according to type.

### 2.2 Application to data clustering

The previous section has discussed computational models introduced by biologists for the clustering and sorting behavior observed in real ant colonies. Collective robotics suggests itself as the obvious technical application of such methods (see Hartmann 2005; Vik 2005 for some recent work), but the approach can also be used for data clustering (Lumer and Faieta 1994). It seems particularly suitable for applications such as network clustering where centralized clustering is not feasible (Ekola et al. 2004; Merkle et al. 2004).

Lumer and Faieta (1994) proposed a basic ant-based data clustering algorithm (shown in Algorithm 1), which closely mimics the ant behavior described in (Deneubourg et al. 1991). Their method associates a position on a toroidal grid with each of the #*items* data items to be clustered. The positions of these items, as well as those of the agents moving them around, are initialized randomly (Lines 2 and 3). Subsequently, the clustering and sorting phase starts: this is a simple loop, in which one agent is randomly selected (Line 5) and performs a step of a given *stepsize* on the grid (Line 6). If it is currently carrying a data item and is placed at a free grid position, it decides probabilistically whether to drop the item at the current position (Lines 7–15). If the agent is not carrying a data item, but it has arrived at a grid position occupied by a data item, it decides probabilistically whether to pick up this item (Lines 16–22). Both picking and dropping decisions require the evaluation of $f(i)$, which provides information on the similarity and density of the data items in the ant's local neighborhood. After a pre-defined number of iterations the algorithm terminates and the output is a spatial *embedding* of the data on the toroidal grid. The picking and dropping probabilities, given a grid position and a particular data item $i$, are computed using the density functions (Deneubourg et al. 1991;

**Algorithm 1** Lumer & Faieta's basic algorithm for ant-based clustering

```
procedure LUMER & FAIETA
  Randomly scatter data items on the toroidal grid
  Randomly place agents on the toroidal grid
  for t := 1 to max_iterations do
    j := random agent
    move agent j randomly by stepsize grid cells
    l := does agent j carry a data item?
    e := is agent j's grid position occupied by a data item?
    if (l = TRUE) ∧ (e = FALSE) then
      i := data item carried by agent j
      drop := (random() ≤ p_drop(i)) // see equations 2 and 3
      if drop = TRUE then
        let agent j drop data item i at its current position
      end if
    end if
    if (l = FALSE) ∧ (e = TRUE) then
      i := data item at agent j's grid position
      pick := (random() ≤ p_pick(i)) // see equations 1 and 3
      if pick = TRUE then
        let agent j pick up data item i
      end if
    end if
  end for
end procedure
```

Lumer and Faieta 1994):

$$p_{\text{pick}}(i) = \left(\frac{k^+}{k^+ + f(i)}\right)^2, \tag{1}$$

and

$$p_{\text{drop}}(i) = \begin{cases} 2f(i) & \text{if } f(i) < k^-, \\ 1 & \text{otherwise,} \end{cases} \tag{2}$$

where $k^+$ and $k^-$ are constants, and $f(i)$ is a neighborhood function introduced in (Lumer and Faieta 1994):

$$f(i) = \max\left(0, \frac{1}{\sigma^2}\sum_{j\in L}\left(1 - \frac{d(i,j)}{\alpha}\right)\right), \tag{3}$$

where, $d(i,j) \in [0,1]$ is a measure of the dissimilarity between data points $i$ and $j$, $\alpha \in [0,1]$ is a data-dependent scaling parameter, and $\sigma^2$ is the size of the local neighborhood $L$.

Figure 1 shows the positions of the data points in different phases of a modified Lumer & Faieta type algorithm (Handl and Meyer 2002). The data consists of four two-dimensional Gaussian clusters with means $(0,0)$, $(0,8)$, $(8,0)$, $(8,8)$, standard deviations $(2,2)$, $(2,2)$, $(2,2)$, $(2,2)$, and 200 elements each. These data points are initially randomly scattered on the grid. The data points in Fig. 1 are colored according to cluster membership. It is clearly visible how the original four clusters are increasingly recovered. The fundamental principle that makes this possible is a positive feedback loop mediated through

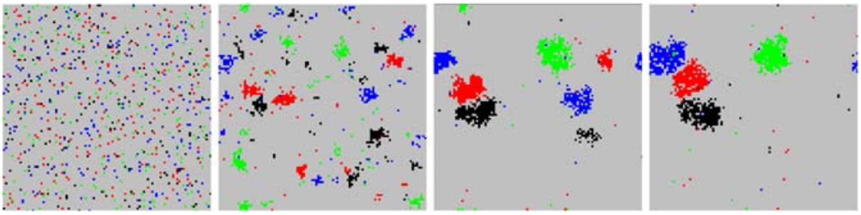

**Fig. 1** Arrangement of four-cluster data on the grid at (from left to right) 0, 10000, 50000, and 130000 iterations of the ant-based clustering algorithm described in (Handl and Meyer 2002)

modification of the environment. Any area that has a concentration of a certain type of data attracts similar data elements as these are more likely to be dropped in this area. These new elements in turn increase the bias of the area towards elements of the same type. Thus, any small initial random fluctuation of the spatial distribution will be amplified and cause the environment to shape into designated zones that attract different types of elements.

Several authors have later introduced extensions and modifications of this algorithm to improve its performance. These include the adaptive setting of some of the algorithm's crucial parameters (Handl and Meyer 2002; Handl et al. 2006; Vizine et al. 2005a, 2005b) the hybridization of ant-based clustering with fuzzy $c$-means and $k$-means algorithms (Gu and Hall 2006; Kanade and Hall 2003, 2004; Monmarché 2000; Monmarché et al. 1999), the simultaneous transportation of entire stacks of data items (Li et al. 2005; Monmarché 2000), the use of short-term memory (Lumer and Faieta 1994; Handl and Meyer 2002; Handl et al. 2006) or pheromone traces (Ramos and Merelo 2002; Vizine et al. 2005b) to direct ant movement towards promising grid positions, information exchange between agents (Montes de Oca et al. 2005), and the replacement of picking and dropping probabilities by fuzzy rules (Schockaert et al. 2004a, 2004b).

An objective comparison of Lumer & Faieta type algorithms to conventional clustering requires the development of mechanisms to automatically (i.e., without any user intervention such as providing the correct number of clusters) convert the spatial embedding generated by the algorithm to an explicit partitioning. In Handl et al. (2006), this was realized using a hierarchical clustering method to group items located at neighboring grid positions. Ant-based clustering was then compared to a number of traditional clustering techniques, including $k$-means, average link hierarchical clustering and self-organizing maps. This comparison showed that across a range of different benchmark data sets, the algorithm performed competitively to these standard algorithms. Ant-based clustering has the advantage of automatically determining the number of clusters, and it was shown to outperform a statistical method (the Gap statistic, Tibshirani et al. 2001) in this respect (Handl et al. 2006).

Lumer and Faieta also proposed to use their algorithm as a "heuristic mapping of a possibly high-dimensional and sparse data set on a plane, in a way which preserves neighborhood relationships as much as possible", that is, as an approximate topographic mapping or multi-dimensional scaling. This motivated a number of real-world applications of ant-based clustering and sorting to topographic mapping, in particular for the visualization of document collections in the form of topic maps (Handl and Meyer 2002; Hoe et al. 2002; Ramos and Merelo 2002) and for graph partitioning (Kuntz and Snyers 1994, 1999; Kuntz et al. 1998). Recent work (Handl et al. 2006), however, demonstrates that the topology preservation provided by the algorithm is in fact quite poor and that ant-based clustering provides a very limited intra-cluster sorting only.

### 2.3 Related methods

Recently, the term ant-based clustering has also been used to refer to a number of conceptually different algorithms in which agents ("ants") are directly associated with individual data items (Azzag et al. 2004; Chen et al. 2004, 2005; Labroche et al. 2004; Moere and Clayden 2005). In these algorithms, agents that carry similar data items attract each other, which results in the formation of groups or hierarchies of agents that are then interpreted as flat or hierarchical partitionings of the data. These methods are thus closely related to flocking-based methods. The general concept of flocking-based clustering and some representative approaches are discussed in Sect. 4.

## 3 Clustering-as-optimization algorithms

An alternative to using specialized clustering methods is to recast clustering as an optimization task, with clustering quality as the objective, and to use a suitable general purpose optimization method to find a good clustering. The principal advantage of this approach is that the objective of the clustering is explicit, which enables us to better understand the performance of the clustering algorithm on particular types of data and to use task-specific clustering objectives. It is even possible to consider several objectives simultaneously, an approach explored recently in (Handl and Knowles 2007).

### 3.1 Ant colony optimization

In principle, any general purpose optimization method can serve as the basis of this approach. Given that the focus of our paper is on ant-based and swarm-based clustering, we will discuss the use of ant colony optimization (ACO) (Dorigo and Stützle 2004; Dorigo et al. 1996). ACO is a stochastic metaheuristic for combinatorial optimization that is inspired by the foraging behavior of mass recruiting ants, which use pheromones to mark promising foraging areas and paths to potential food sources. In ACO a number of agents ("ants") independently construct solutions in parallel by iteratively augmenting partial solutions. The sequence of components that are used in the construction of a solution corresponds to the choices made along a path to a food source. The paradigmatic (and probably most easily understood) problem solved with ACO is the traveling salesman problem (TSP), where a partial solution corresponds to a partial path. It is worthwhile introducing ACO with this classical example, as it is necessary to understand the original metaheuristic to be able to appreciate the modifications made for clustering applications. For the TSP problem, every construction step extends a partial path to a new city. The ants make the choice of the next city to be visited based on a so-called "pheromone value", which models the preference for a particular choice and is cooperatively learned by the ants during the search. In a TSP the pheromone value is commonly attached to a pair of cities $c_i$, $c_j$ and models the preference to go from city $c_i$ to $c_j$. The pheromone values are learned through a reinforcement strategy in which each ant, after the complete solutions have been constructed, reinforces the choices it has made during the solution construction with an amount of pheromone reinforcement proportional to the solution quality it obtained.

### 3.2 Application to data clustering

When a general purpose optimization metaheuristic is used for clustering, a number of important design choices have to be made. Predominantly, these are the problem representation and the objective function. Both of these can have an important effect on optimization performance and thus on clustering quality. Many different solutions have been proposed for clustering in the context of other metaheuristics, specifically genetic algorithms (GAs) (Cole 1998; Jain et al. 1999a; Rayward-Smith 2005). Possible choices for the representation of clustering solutions include permutation-based (Cole 1998), linkage-based (Handl and Knowles 2007), and matrix-based encodings (Cole 1998). Similarly, a wide range of different objective functions can be derived from a large variety of different clustering quality measures (Handl et al. 2005).

In comparison with the variety of approaches investigated for clustering with GAs, to date only a very limited range of design choices have been explored with ACO. Initial studies of ACO-based algorithms for clustering have been presented in (Runkler 2005; Saatchi and Hung 2005). In Runkler (2005) a representation based on the cluster membership matrix is used, resulting in a discrete assignment problem. Based on a similar representation, Saatchi and Hung (2005) propose a hybridization of ACO with $k$-means. The approach in (Runkler 2005) was tested for the standard $k$-means criterion as well as for fuzzy $c$-means, and the author reports good quality results on a number of reasonably small data sets, including the well-known lung cancer test data in the UCI Machine Learning Repository. However, the author also cautions that his approach is computationally more expensive than standard conventional state-of-the-art methods.

### 3.3 Related methods

Next to these direct applications of ACO to clustering, the literature contains a number of ACO-derivatives that cannot directly be understood as clustering-as-optimization approaches (Chu et al. 2004; Tsai et al. 2002). These algorithms modify ACO in such a way that a single explicit global objective can no longer be identified. For this reason, they do not truly belong into the "clustering-as-optimization" category and are better viewed as specialized clustering heuristics. However, as they are still closely related to ACO we will briefly discuss them here.

The basic idea of these ACO-derived heuristics is simple: As in ACO for the TSP, each ant tries to find a cost-minimizing path, where the nodes of the path are the data points to be clustered. Like in the TSP, the cost of moving from data point $x_i$ to $x_j$ is the distance $d(i, j)$ between these points, measured by some appropriate dissimilarity metric. Thus, the next point to be added to the path tends to be similar to the last point on the path.

An important way in which these algorithms deviate from ACO algorithms is that the ants do not necessarily visit all data points: the fraction of data points visited may be a constant fraction of all data points (Chu et al. 2004) or it may be decreasing over time (Tsai et al. 2002). While Tsai et al. (2002) have to be credited with proposing to use reduced path lengths in ACO for clustering, the authors neither describe the algorithm in sufficient detail nor present a mature evaluation. Some first steps towards an evaluation are later reported in (Chu et al. 2004) and the authors come to the conclusion that their adaption of this principle compares favorably with the well-known DBSCAN algorithm (Ester et al. 1996).

An important consequence of the reduced path length is that, unlike in ACO, there is no direct correspondence between an individual path and a full solution (clustering) or a solution component (cluster). Thus, a post-process that extracts explicit clusters from the pheromone matrix is required. The basic idea used for this in (Chu et al. 2004; Tsai et al. 2002) is to break connections between data points that carry less than a threshold level of pheromone and to interpret the remaining connected components as clusters.

## 4 Other swarm-based clustering methods

Apart from ant-inspired techniques, a number of other swarm-based methods have been used for clustering.[2] Unfortunately, it is not an easy task to review this literature. While it is ripe with inspired and promising ideas, there is very little mature work. Thorough evaluations are rare and details of the proposed algorithms are often missing. We will therefore discuss the conceptual ideas behind individual methods, but cannot give full details in many cases.

### 4.1 Information flocking

#### *4.1.1 Biological inspiration*

Flocking is an important type of animal behavior that has been used as the basis of swarm algorithms. The idea goes back to Reynolds' discovery that the flocking behavior of birds can be simulated by groups of "bird-like" agents, so-called *boids*, whose behavior is governed by only three simple activity principles, without any central control of the swarm (Reynolds 1987).

The three principles are:

- Alignment: Each boid aims to match its heading and speed to that of other boids in its immediate neighborhood of radius $\delta_1$.
- Cohesion: Each boid adjusts its steering towards the center position of the group of boids in its immediate neighborhood of radius $\delta_2$.
- Avoidance: A boid that comes too close to another boid steers away from it within a neighborhood of radius $\delta_3$.

Alignment and cohesion are the forces that allow the flock to form groups, while avoidance counteracts crowding and balances the other two forces. These principles are, in fact, biologically realistic (Couzin et al. 2002) even if much simplified. Using a force-directed model, they can be implemented through a simple unified update formula. Let the position of the $i$-th boid $b_i$ at time $t$ be $\vec{x}_i(t)$ and its velocity vector $\vec{v}_i(t)$. The initial values for these are random. At each time step $t$ each boid $b_i$ updates its position and velocity according to

$$\begin{aligned}
\vec{x}_i(t+1) &= \vec{x}_i(t) + \vec{v}_i(t), \\
\vec{v}_i(t+1) &= \alpha_0 \vec{v}_i(t) \\
&\qquad + \alpha_1 \frac{1}{|N_1(i,t)|} \sum_{j \in N_1(i,t)} \vec{v}_j(t) \quad //\text{alignment} \\
&\qquad + \alpha_2 \left( \frac{1}{|N_2(i,t)|} \sum_{j \in N_2(i,t)} \vec{x}_j(t) - \vec{x}_i(t) \right) \quad //\text{cohesion} \\
&\qquad - \alpha_3 \sum_{j \in N_3(i,t)} \big( \vec{x}_j(t) - \vec{x}_i(t) \big) \quad //\text{avoidance.}
\end{aligned}$$

Hence, the new velocity vector of the boid is computed as the weighted sum between its old velocity vector and the forces resulting from the three activity principles. The $\alpha_k$ are the

[2] We take "swarm" to mean a collection of spatially embedded motile agents that interact using predominantly local knowledge and simple local interactions.

positive weighting factors for the individual components and $N_k(i,t)$ is the neighborhood of $b_i$ at time $t$ taking into account the different sensory radii for the $k$-th activity principle

$$N_k(i,t) = \{ j \mid \delta_k \geq \| (\vec{x}_i - \vec{x}_j) \| \}.$$

Even though these principles are completely localized, they enable the group to exhibit complex and natural flocking behavior (Reynolds 1987). Additional environmental influences, such as a preferred flock direction for homing or avoidance of external obstacles can easily be taken into account by extending the force-directed model.

#### *4.1.2 Application to data clustering*

The idea of using boid-like flocking for clustering in information visualization is due to Proctor and Winter. They presented a flocking implementation to cluster people into social groups according to their different interests in (Proctor and Winter 1998) where the term "information flocking" was coined.

The boid idea is adapted by representing each person as one boid. $\vec{x}_i$ is taken as the boid's on-screen 2d-position for visualization. A flocking behavior that encourages clustering according to interests is achieved by weighting the alignment component in the velocity update with an additional factor $s(i,j)$ that is proportional to the similarity of interests of person $i$ and person $j$. Furthermore, the cohesion component is replaced by a homing force that steers the boids towards the center of the screen so that they do not leave the visualization field.

It is worth noting that this method was not intended for static clustering but instead for dynamic information visualization, where more similar documents stay more closely together in a flock, while dissimilar documents drift apart.[3]

Information flocking has later been extended to visualize time-varying data in (Vande Moere 2004). Here stock market movements are visualized such that each boid represents a single share. While the high-level group behavior of the flock defies a simple and direct interpretation, it is still interesting to observe that stocks with an unusual behavior (i.e., not well aligned with average market movement) are typically found at the periphery of the flock, often in peripheral "bulges", while stocks whose behavior is close to average, such as the S&P500, are found at the center of the flock.

The idea of information flocking later resurfaces in (Cui et al. 2006) where it is used for text document classification.

A related application of flocking algorithms is to cluster discovery in spatial data. This idea goes back to (Macgill 2000; Openshaw and Macgill 1998) and was later extended in (Folino and Spezzano 2002; Folino et al. 2003a, 2003b). The task is to cluster a set of spatially distributed data points according to their attributes and spatial distribution into spatial clusters of "interesting" data. The algorithms use a set of boids that move through the space into which these data points are embedded and that have the ability to sense the data in their immediate neighborhood. The boids are divided into four groups, marked by colors.

[3]Using force-based approaches to generate *static* embeddings is a standard technique in automatic graph layout (Di Battista et al. 1999). Indeed, we can use a dataset to generate a complete graph in which each vertex corresponds to one data point and in which the desired edge lengths are proportional to the corresponding entries of the data's dissimilarity matrix. The layout problem for this graph is isomorphic to the standard multi-dimensional scaling problem (MDS) (Borg and Groenen 1997) for the same data set. Its solution can be understood as a soft geometric clustering. Advanced MDS techniques, such as stress majorization (Gansner et al. 2004), can be used to solve this problem efficiently.

A boid that senses a very high density of interesting data in its immediate neighborhood turns red, one that sees no interesting data at all turns white, and boids that sense an intermediate density turn green (higher density) or yellow (lower density). A boid that turns red or white stops altogether, in order to mark such regions. Green and yellow boids are controlled by the same behavioral rules as in Reynold's original method, but the forces are modified such that alignment and cohesion ignore yellow boids. White boids become the origin of repulsive forces. In (Folino and Spezzano 2002) the performance of their SPARROW algorithm is compared to DBSCAN (Ester et al. 1996), which is a conventional standard algorithm for this problem. The paper comes to the conclusion that SPARROW performs comparably to DBSCAN but uses slightly fewer spatial queries.

## 4.2 Particle swarm optimization for clustering

### *4.2.1 Particle swarm optimization*

As ant foraging has inspired ACO, flocking behavior has inspired PSO (Particle Swarm Optimization (Kennedy and Eberhart 1995)), another general purpose optimization metaheuristic. PSO is used to optimize functions of continuous variables and its basic algorithmic form is of extreme simplicity. To optimize an objective function $f(\vec{x}) : \mathbf{R}^n \to \mathbf{N}$, a swarm of "particles" moves cooperatively in $\mathbf{R}^n$. Each particle represents a complete solution and moves under the influence of forces that attract it to some degree to good positions in the search space that have previously been explored by itself or by other swarm members. Initially, a number of particles are placed at random positions in the $n$-dimensional search space and are given random velocities. As above, let the position of particle $i$ be $\vec{x}_i$ and its velocity $\vec{v}_i$. At every time step each particle evaluates the objective at its position. It then updates its position and velocity as well as a memory for its individually best position $\vec{p}_i^{ib}$ throughout the process and the memory for the best position $\vec{p}^{gb}$ of all particles throughout the process. The particles have some momentum and are attracted to both of these memory positions:

$$
\begin{aligned}
&\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}(t),\\
&\vec{v}_i(t+1) = \alpha_1 \vec{v}_i(t) + \alpha_2 \mu_1 \big(\vec{p}_i^{ib} - \vec{x}_i(t)\big) + \alpha_3 \mu_2 \big(\vec{p}^{gb} - \vec{x}_i(t)\big),\\
&\vec{p}_i^{ib} \leftarrow \begin{cases} \vec{x}_i(t) & \text{if } f(\vec{x}_i(t)) < f(\vec{p}_i^{ib}),\\ \vec{p}_i^{ib} & \text{otherwise,} \end{cases}\\
&\vec{p}^{gb} \leftarrow \begin{cases} \vec{x}_i(t) & \text{if } f(\vec{x}_i(t)) < f(\vec{p}^{gb}),\\ \vec{p}^{gb} & \text{otherwise} \end{cases}
\end{aligned}
$$

where $\mu_i$ are random variables $U(0, 1)$. Modern PSO implementations deviate significantly from this simple schema and often integrate ideas from other stochastic search methods (Poli et al. 2007).

### *4.2.2 Application to data clustering*

A straightforward implementation of PSO for clustering was introduced in (Omran et al. 2002). The algorithm uses a fixed number of clusters $n_c$ and uses PSO to search for the optimal centroids of these clusters. As in vector quantization, each data point automatically belongs to the cluster with the closest centroid. Each particle represents the position of all the $n_c$ centroids for the clusters. For a $d$-dimensional data space, $\vec{x}_i$ is thus $(d \cdot n_c)$-dimensional. These particles search the space using the normal PSO algorithm and use the vector quantization error as the objective function:

$$f(\vec{x}) = \frac{1}{n_c} \sum_{j=1}^{n_c} \sum_{z \in C_j} \|\vec{z} - \vec{m}_j\|$$

where $C_j$ is the set of all data points belonging to the $j$th cluster and $\vec{m}_j$ is its centroid. The idea for this clustering method has been introduced in (Omran et al. 2002) for use with image data and has been extended in (van der Merwe and Engelbrecht 2003) through a hybridization with $k$-means[4] where a preliminary performance evaluation is given for the standard IRIS and WINE clustering test data sets. Other authors have used the same idea of a PSO + $k$-means hybrid for clustering of text documents (Cui et al. 2005; Cui and Potok 2005). In contrast to (van der Merwe and Engelbrecht 2003), they use $k$-means also as a post-process. This makes sense, as $k$-means is known to be a local optimizer (Kanungo et al. 2000) and therefore can only improve the results. However, it is not clear from these studies whether the improved performance of the hybrids justifies the computational effort of using swarms. Its results are, however, not directly comparable to (van der Merwe and Engelbrecht 2003) as the nature of the test data is quite different: (Cui and Potok 2005) cluster high-dimensional data for text classification of the TREC databases.

### 4.3 Swarm-based clustering in context

A variety of other approaches that are not directly based on swarming metaphors can nevertheless be considered as related to swarm-based algorithms. From an extreme angle, it could be argued that all population-based stochastic search methods are somewhat related to swarm intelligence. The surveys (Jain et al. 1999b; Rayward-Smith 2005) give a discussion of some of these for clustering, specifically of genetic algorithms. Estimation of distribution algorithms (EDAs) (Larrañaga and Lozano 2002) are a close relative of genetic algorithms. Like these, they are based on a sampling and elimination process. In contrast to these, they do not manipulate individual population members directly, but instead represent a whole population implicitly via joint probability distributions. Specifically, the univariate marginal distribution algorithm (UMDA) (Mühlenbein 1998) and population-based incremental learning (PBIL) (Baluja and Caruana 1995) are closely related to ACO (Cordón et al. 2002; Monmarché et al. 2000; Zlochin et al. 2004). EDAs for clustering have been studied in (Roure et al. 2002), where a comparison with GAs and $k$-means is given. Another nature-inspired algorithm framework, often cited in the context of swarm intelligence, is artificial immune systems (AIS) (Dasgupta 1999). These, too, are closely related to genetic algorithms. To the best of the authors' knowledge the idea to use AIS for clustering goes back to (Timmis et al. 1999), with more complete descriptions given in (Timmis et al. 2000; Timmis and Neal 2001). A closely related approach is presented in (de Castro and von Zuben 2000). Finally, Kohonen networks or self-organizing maps (SOMs) (Kohonen 1997) deserve a brief mention. While this popular method for clustering high-dimensional data (as well as to project high-dimensional data into low-dimensional spaces) is usually interpreted as a neural-net structure, it could also be interpreted as a swarm of motile elements in a topological neighborhood. This renders it closely related to flocking mechanisms for clustering. While some readers may feel that this broadens the concept of a swarm-based approach too much, SOMs certainly have a special place as the first self-organizing approach to clustering.

[4] $k$-means is used as a pre-process to initialize the PSO positions.

## 5 Perspectives

### 5.1 Applications

Ant-based clustering algorithms have recently been applied in a range of different problem domains, underlining the increasing interest in these new types of clustering algorithms from practitioners. For the interested reader Table 1 gives selected examples of the different areas in which ant-based clustering has found application. Note that the table does not include applications of ant-based methods to data-mining tasks different from clustering, which can also be found in the literature (see Parpinelli et al. 2002, 2005; Langham 2005). Swarm-based clustering methods, different from ant-based clustering, have also been applied for a number of different clustering tasks, most prominently spatial data mining (Macgill 2000; Openshaw and Macgill 1998; Folino and Spezzano 2002) and text mining (Cui and Potok 2005; Cui et al. 2005, 2006), as discussed in Sect. 4.

### 5.2 Challenges for the future

Research on swarm-based clustering is still very much in its infancy and driven by the desire to understand how self-organized behavior can achieve such a task. This, in itself, is a valuable goal. However, from an application perspective new clustering algorithms can be of use only if they are thoroughly tested and compared with established methods. New clustering algorithms are proposed quite frequently in the data mining and pattern analysis literature, either as improvements over existing techniques, or as bespoke solutions to a particular (perhaps novel) class of clustering problems. Whether such methods make an impact or just contribute to the battery of techniques available depends very much on a detailed understanding of the particular properties of the method proposed, most of all on an understanding of the types of data it is suitable for. In practice, the first requirement for the application of a clustering algorithm is a precise definition of the specific type of clustering problems to be solved. Different algorithms make very different assumptions about the type of cluster structures sought, and given any prior knowledge available about the clustering

**Table 1** Examples of applications of ant-based clustering

| Application area | Reference |
|---|---|
| Intrusion detection | Ramos and Abraham (2005) |
| | Tsang and Kwong (2005) |
| Web usage mining and recommendation systems | Abraham and Ramos (2003) |
| | Labroche et al. (2003) |
| Graph partitioning | Kuntz et al. (1998) |
| Bioinformatics | Vizine et al. (2005b) |
| Text mining | Handl and Meyer (2002) |
| | Azzag et al. (2004) |
| | Vizine et al. (2005a) |
| Packet clustering in router-based networks | Merkle et al. (2004) |
| Texture segmentation | Channa et al. (2006) |

problem at hand it is of fundamental importance to select an algorithm consistent with this. Each algorithm has a bias towards particular types of cluster structures, and many algorithms will even generate a clustering in the absence of a natural class structure. Cluster validation (Handl et al. 2005) is thus crucial both (i) in application studies for the purpose of result verification, and (ii) in algorithm development in order to provide an objective assessment of the strengths and weaknesses of a given method.

Ideally, empirical analysis of the performance of the algorithm should be complemented by theoretical results on the working mechanisms of the algorithms and by predictions regarding the types of cluster problems that can be solved. This endeavor would be of particular importance for algorithms like those reviewed in Sect. 2, where no explicit clustering objective is given, making it particularly challenging to obtain a thorough understanding of the biases and limitations of the method.

### 5.3 Potential of ant-based and swarm-based clustering

Aspects other than clustering accuracy may also factor in the practical utility of a clustering algorithm. Ant-based clustering and, more generally, swarm-based clustering may have a number of potentially attractive features based on their self-organizing nature.

1. The absence of central control may make this type of algorithm suitable for the application in distributed clustering problems (Chandrasekar and Srinivasan 2007), that is, clustering problems where the data cannot be accessed in its entirety, for example, due to geographical, time or space constraints.
2. Their use of multiple independent agents renders some swarm-based algorithms naturally suitable for parallelization.
3. Swarm-based clustering algorithms are usually thought to promise robustness and adaptiveness, and may thus be able to deal with noise and incremental data.

Evidently, all of the above features would make ant-based and swarm-based clustering highly attractive in many important clustering applications. Unfortunately, there is currently a lack of rigorous investigations of the degree to which existing algorithms really fulfill those promises. For example, despite contrary claims (Albuquerque and Dupuis 2002), it is not immediately obvious how Lumer & Faieta type algorithms could be successfully parallelized. It is crucial that in the future such claims be precisely defined for the particular application at hand, measured objectively, and demonstrated rigorously.

## 6 Conclusion

This survey has been concerned with clustering algorithms that draw their inspiration from ant behavior observed in nature. Two distinct classes of ant-based clustering algorithms have been identified: those that directly imitate clustering behaviors observed in nature and those that are based on ant colony optimization. Other types of swarm-based clustering methods and their relation to these two categories have also been discussed, applications and advantages of ant-based clustering methods have been summarized and challenges for future research have been highlighted. The initial research on swarm-based clustering has opened a fascinating world of novel clustering methods. For these to graduate from being merely ‘interesting’ to being practically useful it will be vital that future research evaluates such approaches thoroughly and critically, both theoretically and in the context of real world applications.

## References


Abraham, A., & Ramos, V. (2003). Web usage mining using artificial ant colony clustering and linear genetic programming. In R. Sarker (Ed.), *The 2003 congress on evolutionary computation, CEC 2003* (Vol. 2, pp. 1384–1391). Piscataway: IEEE Press.

Albuquerque, P., & Dupuis, A. (2002). A parallel cellular ant colony algorithm for clustering and sorting. In S. Bandini, B. Chopard, & M. Tomassini (Eds.), *Lecture notes in computer science: Vol. 2493*. *Cellular automata: 5th international conference on cellular automata for research and industry, ACRI 2002* (pp. 220–230). Berlin: Springer.

Arabie, P., Lawrence, J. H., & Soete, G. D. (1996). *Clustering and classification*. River Edge: World Scientific.

Azzag, H., Guinot, C., & Venturini, G. (2004). How to use ants for hierarchical clustering. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, & T. Stützle (Eds.), *Lecture notes in computer science: Vol. 3172*. *Ant colony optimization and swarm intelligence: 4th international workshop, ANTS 2004* (pp. 350–357). Berlin: Springer.

Baluja, S., & Caruana, R. (1995). Removing the genetics from the standard genetic algorithm. In A. Prieditis & S. Russel (Eds.), *Machine learning, proceedings of the twelfth international conference on machine learning, ICML 1998* (pp. 38–46). San Francisco: Kaufmann.

Bilenko, M., Basu, S., & Mooney, R. (2004). Integrating constraints and metric learning in semi-supervised clustering. In R. Greiner & D. Schuurmans (Eds.), *Proceedings of the 21st international conference on machine learning, ICML 2004* (pp. 119–126). New York: ACM.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence—from natural to artificial systems*. Oxford: Oxford University Press.

Borg, I., & Groenen, P. (1997). *Modern multidimensional scaling: theory and applications*. New York: Springer.

Campos, M., Bonabeau, E., Theraulaz, G., & Deneubourg, J.-L. (2001). Dynamic scheduling and division of labor in social insects. *Adaptive Behaviour*, *8*(2), 83–92.

Chandrasekar, R., & Srinivasan, T. (2007). An improved probabilistic ant based clustering for distributed databases. In M. M. Veloso (Ed.), *IJCAI 2007, proceedings of 20th international joint conference on artificial intelligence* (pp. 2701–2706). Online proceedings available at http://www.ijcai.org/papers07/contents.php.

Channa, A. H., Rajpoot, N. M., & Rajpoot, K. M. (2006). Texture segmentation using ant tree clustering. In *2006 IEEE international conference on engineering of intelligent systems, ICEIS 2006* (pp. 1–6). Piscataway: IEEE Press.

Chen, L., Xu, X.-H., & Chen, Y.-X. (2004). An adaptive ant colony clustering algorithm. In *Proceedings of the international conference on machine learning and cybernetics, ICMLC 04* (Vol. 3, pp. 1387–1392). Piscataway: IEEE Press.

Chen, L., Tu, L., & Chen, H. (2005). A novel ant clustering algorithm with digraph. In L. Wang, K. Chen, & Y. S. Ong (Eds.), *Lecture notes in computer science: Vol. 3611*. *Advances in natural computation first international conference, ICNC 2005* (pp. 1218–1228). Berlin: Springer.

Chu, S.-C., Roddick, J. F., Su, C.-J., & Pan, J.-S. (2004). Constrained ant colony optimization for data clustering. In C. Zhang, H. W. Guesgen, & W. K. Yeap (Eds.), *Lecture notes in artificial intelligence: Vol. 2417*. *PRICAI 2004: trends in artificial intelligence: 8th pacific rim international conference on artificial intelligence* (pp. 534–543). Berlin: Springer.

Cole, R. M. (1998). *Clustering with genetic algorithms*. Master's thesis, Department of Computer Science, University of Western Australia, Australia.

Cordón, O., Herrera, F., & Stützle, T. (2002). A review on the ant colony optimization metaheuristic: basis, models and new trends. *Mathware and Soft Computing*, *9*(2–3), 141–175.

Couzin, I., Krause, J., James, R., Ruxton, G., & Franks, N. (2002). Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, *218*, 1–11.

Cui, X., & Potok, T. E. (2005). Document clustering analysis based on hybrid PSO+K-means algorithm. *Journal of Computer Sciences*, *5*, 27–33.

Cui, X., Potok, T. E., & Palathingal, P. (2005). Document clustering using particle swarm optimization. In *Proceedings of the IEEE swarm intelligence symposium, SIS 2005* (pp. 185–191). Piscataway: IEEE Press.

Cui, X., Gao, J., & Potok, T. E. (2006). A flocking based algorithm for document clustering analysis. *Journal of Systems Architecture*, *52*, 505–515.

Dasgupta, D. (Ed.). (1999). *Artificial immune systems and their applications*. Berlin: Springer.

de Castro, L. N., & von Zuben, F. J. (2000). An evolutionary immune network for data clustering. In *Proceedings of the Brazilian symposium on neural networks, SBRN 00* (pp. 84–89). Piscataway: IEEE Press.

Deneubourg, J.-L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., & Chrétien, L. (1991). The dynamics of collective sorting: robot-like ants and ant-like robots. In J.-A. Meyer & S. Wilson (Eds.), *From animals to animats: proceedings of the first international conference on simulation of adaptive behavior* (pp. 356–365). Cambridge: MIT Press.

Di Battista, G., Eades, P., Tamassia, R., & Tollis, I. (1999). *Graph drawing*. Upper Saddle River: Prentice Hall.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. Cambridge: MIT Press.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, *26*(1), 29–41.

Ekola, T., Laurikkala, M., Lehto, T., & Koivisto, H. (2004). Network traffic analysis using clustering ants. In *Proceedings of the world automation congress, WAC 2004* (Vol. 17, pp. 275–280). Piscataway: IEEE Press.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, & U. Fayyad (Eds.), *Proceedings of the second international conference on knowledge discovery and data mining* (pp. 226–231). Menlo Park: AAAI Press.

Everitt, B., Laindau, S., & Leese, M. (2001). *Cluster analysis* (4th ed.). London: Edward Arnold.

Folino, G., & Spezzano, G. (2002). An adaptive flocking algorithm for spatial clustering. In J. J. M. Guervós, P. Adamidis, H.-G. Beyer, & J. L. Fernández-Villacañas (Eds.), *Lecture notes in computer science: Vol. 2439. Parallel problem solving from nature—PPSN VII* (pp. 924–933). Berlin: Springer.

Folino, G., Forestiero, A., & Spezzano, G. (2003a). Discovering clusters in spatial data using swarm intelligence. In W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, & J. Ziegler (Eds.), *Lecture notes in artificial intelligence: Vol. 2801. Advances in artificial life: 7th European conference, ECAL 2003* (pp. 598–605). Berlin: Springer.

Folino, G., Forestiero, A., & Spezzano, G. (2003b). Swarming agents for discovering clusters in spatial data. In *Proceedings of the international symposium on parallel and distributed computing, ISPDC 03* (pp. 72–79). Piscataway: IEEE Press.

Gansner, E., Koren, Y., & North, S. (2004). Graph drawing by stress majorization. In J. Pach (Ed.), *Lecture notes in computer science: Vol. 3383. Graph drawing: 12th international symposium, GD 2004* (pp. 239–250). Berlin: Springer.

Garey, M., & Johnson, D. (1979). *Computers and intractability; a guide to the theory of NP-completeness*. New York: Freeman.

Gu, Y., & Hall, L. O. (2006). Kernel based fuzzy ant clustering with partition validity. In *Proceedings of the IEEE international conference on fuzzy systems* (pp. 263–267). Piscataway: IEEE Press.

Handl, J., & Knowles, J. (2007). An evolutionary approach to multiobjective clustering. *IEEE Transactions on Evolutionary Computation*, *11*(1), 56–76.

Handl, J., & Meyer, B. (2002). Improved ant-based clustering and sorting in a document retrieval interface. In J. J. Merelo, P. Adamidis, & H.-G. Beyer (Eds.), *Lecture notes in computer science: Vol. 2439. Parallel problem solving from nature—PPSN VII* (pp. 913–923). Berlin: Springer.

Handl, J., Knowles, J., & Kell, D. B. (2005). Computational cluster validation in post-genomic data analysis. *Bioinformatics*, *21*(15), 3201–3212.

Handl, J., Knowles, J., & Dorigo, M. (2006). Ant-based clustering and topographic mapping. *Artificial Life*, *12*(1), 35–61.

Hartmann, V. (2005). Evolving agent swarms for clustering and sorting. In H.-G. Beyer (Ed.), *Proceedings of the genetic and evolutionary computation conference* (pp. 217–224). New York: ACM.

Hoe, K., Lai, W., & Tai, T. (2002). Homogeneous ants for web document similarity modeling and categorization. In M. Dorigo & G. Di (Eds.), *Lecture notes in computer science: Vol. 2463. Ant algorithms: third international workshop, ANTS 2002* (pp. 256–261). Berlin: Springer.

Jain, A., & Dubes, R. (1988). *Algorithms for clustering data*. Englewood Cliffs: Prentice Hall.

Jain, A., Murty, M., & Flynn, P. (1999a). Data clustering: a review. *ACM Computing Surveys*, *31*(3), 264–323.

Jain, A., Murty, M., & Flynn, P. J. (1999b). Data clustering: a review. *ACM Computing Surveys*, *31*, 264–323.

Kanade, P. M., & Hall, L. O. (2003). Fuzzy ants as a clustering concept. In *NAFIPS 2003: 22nd international conference of the North American fuzzy information processing society* (pp. 227–232). Piscataway: IEEE Press.

Kanade, P. M., & Hall, L. O. (2004). Fuzzy ant clustering by centroid positioning. In *Proceedings of the IEEE international conference on fuzzy systems* (Vol. 1, pp. 371–376). Piscataway: IEEE Press.

Kannan, R., Vempala, S., & Vetta, A. (2004). On clusterings: good, bad and spectral. *Journal of the ACM*, *51*(3), 497–515.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2000). The analysis of a simple k-means clustering algorithm. In *Symposium on computational geometry* (pp. 100–109). New York: ACM.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE international joint conference on neural networks, IJCNN 95* (pp. 1942–1948). Piscataway: IEEE Press.

Kohonen, T. (1997). *Self-organizing maps*. Berlin: Springer.

Kuntz, P., & Snyers, D. (1994). Emergent colonization and graph partitioning. In D. Cliff, P. Husbands, J.-A. Meyer, & S.W. Wilson (Eds.), *From animals to animats 3: proceedings of the third international conference on simulation of adaptive behavior* (pp. 494–500). Cambridge: MIT Press.

Kuntz, P., & Snyers, D. (1999). New results on an ant-based heuristic for highlighting the organization of large graphs. In *Proceedings of the 1999 congress on evolutionary computation, CEC 99* (pp. 1451–1458). Piscataway: IEEE Press.

Kuntz, P., Snyers, D., & Layzell, P. (1998). A stochastic heuristic for visualising graph clusters in a bi-dimensional space prior to partitioning. *Journal of Heuristics*, *5*(3), 327–351.

Labroche, N., Monmarché, N., & Venturini, G. (2003). Antclust: ant clustering and web usage mining. In E. Cantu-Paz (Ed.), *Lecture notes in computer science: Vol. 2723. Genetic and evolutionary computation, GECOO 2003* (pp. 25–36). Berlin: Springer.

Labroche, N., Guinot, C., & Venturini, G. (2004). Fast unsupervised clustering with artificial ants. In X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, et al. (Eds.), *Lecture notes in computer science: Vol. 3242. Parallel problem solving from nature, PPSN VIII* (pp. 1143–1152). Berlin: Springer.

Langham, E. (2005). Emergent restructuring of resources in ant colonies: a swarm-based approach to partitioning. In M. Ali & F. Esposito (Eds.), *Lecture notes in artificial intelligence: Vol. 3533. Innovations in applied artificial intelligence: 18th international conference on industrial and engineering applications of artificial intelligence and expert systems* (pp. 638–647). Berlin: Springer.

Larrañaga, P., & Lozano, J. (2002). *Genetic algorithms and evolutionary computation: Vol. 2. Estimation of distribution algorithms: a new tool for evolutionary computation*. Boston: Kluwer Academic.

Li, Q., Shi, Z., Shi, J., & Shi, Z. (2005). Swarm intelligence clustering algorithm based on attractor. In L. Wang, K. Chen, & Y.-S. Ong (Eds.), *Lecture notes in computer science: Vol. 3612. Advances in natural computation, first international conference, ICNC 2005* (pp. 496–504). Berlin: Springer.

Lumer, E., & Faieta, B. (1994). Diversity and adaptation in populations of clustering ants. In D. Cliff, P. Husbands, J.-A. Meyer, & S. W. Wilson (Eds.), *From animals to animats 3: proceedings of the third international conference on simulation of adaptive behavior* (pp. 501–508). Cambridge: MIT Press.

Macgill, J. (2000). Using flocks to drive a geographic analysis machine. In M. Bedau, J. S. McCaskill, N. H. Packard, S. Rasmussen, J. McCaskill, & N. Packard (Eds.), *Artificial life VII: proceedings of the seventh international conference on artificial life* (pp. 446–453). Cambridge: MIT Press.

MacQueen, L. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (pp. 281–297). Berkeley: University of California Press.

Melhuish, C., Sendova-Franks, A. B., & Scholes, S. (2006). Ant inspired sorting by robots: the importance of initial clustering. *Journal of the Royal Society Interface*, *3*(7), 235–242.

Merkle, D., Middendorf, M., & Scheidler, A. (2004). Decentralized packet clustering in networks. In *Proceedings of the 18th international parallel and distributed processing symposium, IPDPS 2004* (pp. 163–170). Piscataway: IEEE Press.

Moere, A. V., & Clayden, J. J. (2005). Cellular ants: combining ant-based clustering with cellular automata. In *17th IEEE international conference on tools with artificial intelligence, ICTAI 2005* (pp. 177–184). Piscataway: IEEE Press.

Monmarché, N. (2000). *Algorithmes de fourmis artificielles: applications à la classification et à l'optimisation*. PhD thesis, Laboratoire d'Informatique, Université de Tours, France.

Monmarché, N., Slimane, M., & Venturini, G. (1999). On improving clustering in numerical databases with artificial ants. In D. Floreano, J.-D. Nicoud, & F. Mondada (Eds.), *Lecture notes in artificial intelligence: Vol. 1674. Advances in artificial life: 5th European conference, ECAL 99* (pp. 626–635). Berlin: Springer.

Monmarché, N., Ramat, E., Desbarats, L., & Venturini, G. (2000). Probabilistic search with genetic algorithms and ant colonies. In A. S. Wu (Ed.), *Workshop on optimization by building and using probabilistic models, GECCO 2000* (pp. 209–211).

Montes de Oca, M. A., Garrido, L., & Aguirre, J. L. (2005). Effects of inter-agent communication in ant-based clustering algorithms: a case study on communication policies in swarm systems. In A. Gelbukh & H. Terashima (Eds.), *Lecture notes in artificial intelligence: Vol. 3789. MICAI 2005: advances in artificial intelligence: 4th Mexican international conference on artificial intelligence* (pp. 254–263). Berlin: Springer.

Mühlenbein, H. (1998). The equation for response to selection and its use for prediction. *Evolutionary Computation*, *5*, 303–346.

Ng, A. Y., Jordan, M., & Weiss, Y. (2001). On spectral clustering: analysis and an algorithm. In T. Diettrich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems 14: proceedings of the 2001 neural information processing systems (NIPS) conference* (pp. 849–856). Cambridge: MIT Press.

Omran, M., Salman, A., & Engelbrecht, A. (2002). Image classification using particle swarm optimization. In L. Wang, K. C. Tan, T. Furuhashi, J.-H. Kim, & X. Yao (Eds.), *Proceedings of the fourth Asia-pacific conference on simulated evolution and learning, SEAL'02* (pp. 370–374). Piscataway: IEEE Press.

Openshaw, S., & Macgill, J. (1998). The use of flocks to drive a geographical analysis machine. In R. J. Abrahart (Ed.), *GeoComputation 98, proceedings of the 3rd international conference on geocomputation*. GeoComputation CD-ROM.

Parpinelli, S., Lopes, S., & Freitas, A. (2002). Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation*, *6*, 321–332.

Parpinelli, S., Lopes, S., & Freitas, A. (2005). Classification-rule discovery with an ant colony algorithm. In *Proceedings of the IEEE/WIC international conference on intelligent agent technology* (pp. 420–424). Piscataway: IEEE Press.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization—an overview. *Swarm Intelligence*, *1*(1), 33–57.

Proctor, G., & Winter, C. (1998). Information flocking: data visualization in virtual worlds using emergent behaviours. In J.-C. Heudin (Ed.), *Lecture notes in computer science: Vol. 1434. Proceedings of the international conference on virtual worlds* (pp. 168–176). Berlin: Springer.

Ramos, V., & Abraham, A. (2005). Antids: self-organized ant-based clustering model for intrusion detection system. In A. Abraham, Y. Dote, T. Furuhashi, M. Köppen, A. Ohuchi, & Y. Ohsawa (Eds.), *Springer engineering series: Vol. 29. Soft computing as transdisciplinary science and technology: proceedings of the fourth IEEE international workshop WSTST 05* (pp. 977–986). Berlin: Springer.

Ramos, V., & Merelo, J. (2002). Self-organized stigmergic document maps: environments as a mechanism for context learning. In *Proceedings of the first Spanish conference on evolutionary and bio-inspired algorithms* (pp. 284–293). Mérida: Centro Univ. Mérida.

Rayward-Smith, V. J. (2005). Metaheuristics for clustering in KDD. In *Proceedings of the 2005 IEEE congress on evolutionary computation* (Vol. 3, pp. 2380–2387). Piscataway: IEEE Press.

Reynolds, C. W. (1987). Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics*, *21*(4), 25–34.

Roure, J., Larrañaga, P., & Sangüesa, R. (2002). An empirical comparison between $k$-means, GAs and EDAs in partitional clustering. In P. Larrañaga & J. A. Lozan (Eds.), *Estimation of distribution algorithms*, Chap. 17. Norwell: Kluwer Academic.

Runkler, T. A. (2005). Ant colony optimization of clustering models. *International Journal of Intelligent Systems*, *20*(12), 1233–1251.

Saatchi, S., & Hung, C. C. (2005). Hybridization of the ant colony optimization with the $k$-means algorithm for clustering. In *Lecture notes in computer science: Vol. 3540. Image analysis*. Berlin: Springer.

Schockaert, S., Cock, M. D., Cornelis, C., & Kerre, E. E. (2004a). Efficient clustering with fuzzy ants. In D. Ruan, P. D'hondt, M. D. Cock, M. Nachtegael, & E. E. Kerre (Eds.), *Applied computational intelligence, proceedings of the 6th international FLINS conference* (pp. 195–200). River Edge: World Scientific.

Schockaert, S., Cock, M. D., Cornelis, C., & Kerre, E. E. (2004b). Fuzzy ant based clustering. In M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, & T. Stützle (Eds.), *Lecture notes in computer science: Vol. 3172. Ant colony optimization and swarm intelligence, 4th international workshop, ANTS 2004* (pp. 342–349). Berlin: Springer.

Scholes, S., & Wilson, M. (2004). Comparisons in evolution and engineering: the collective intelligence of sorting. *Adaptive Behavior*, *12*(3–4), 147–159.

Sendova-Franks, A. B., Scholes, S. R., Franks, N. R., & Melhuish, C. (2004). Brood sorting by ants: two phases and differential diffusion. *Animal Behaviour*, *68*, 1095–1106.

Theoridis, S., & Koutrumbas, K. (2006). *Pattern recognition* (3rd ed.). New York: Academic Press.

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*, 411–423.

Timmis, J., & Neal, M. (2001). A resource limited artificial immune system for data analysis. *Knowledge-based Systems*, *14*, 121–130.

Timmis, J., Neal, M., & Hunt, J. (1999). Data analysis using artificial immune systems, cluster analysis and Kohonen networks: some comparisons. In *Proceedings of the IEEE international conference on systems, man, and cybernetics* (Vol. 3, pp. 922–927). Piscataway: IEEE Press.

Timmis, J., Neal, M., & Hunt, J. (2000). An artificial immune systems for data analysis. *BioSystems*, *55*, 143–150.

Tsai, C.-F., Wu, H.-C., & Tsai, C.-W. (2002). A new clustering approach for data mining in large databases. In *Proceedings of the international symposium on parallel architectures, algorithms and networks* (pp. 278–283). Piscataway: IEEE Press.

Tsang, W., & Kwong, S. (2005). Unsupervised anomaly intrusion detection using ant colony clustering model. In A. Abraham, Y. Dote, T. Furuhashi, M. Köppen, A. Ohuchi, & Y. Ohsawa (Eds.), *Springer engineering series: Vol. 29. Soft computing as transdisciplinary science and technology: proceedings of the fourth IEEE international workshop, WSTST 05* (pp. 223–232). Berlin: Springer.

van der Merwe, D., & Engelbrecht, A. (2003). Data clustering using particle swarm optimization. In *Proceedings of the 2003 congress on evolutionary computation* (pp. 215–220). Piscataway: IEEE Press.

Vande Moere, A. (2004). Information flocking: time-varying data visualization using boid behaviors. In *Proceedings of the eighth international conference on information visualization* (pp. 409–414). Piscataway: IEEE Press.

Vik, A. H. (2005). Evolving annular sorting in ant-like agents. In M. Capcarrere, A. A. Freitas, P. J. Bentley, C. G. Johnson, & J. Timmis (Eds.), *Lecture notes in artificial intelligence: Vol. 3630. Advances in artificial life: 8th European conference, ECAL 2005* (pp. 594–603). Berlin: Springer.

Vizine, A. L., de Castro, L. N., & Gudwin, R. R. (2005a). Text document classification using swarm intelligence. In *International conference on the integration of knowledge intensive multi-agent systems* (pp. 134–139). Piscataway: IEEE Press.

Vizine, A. L., de Castro, L. N., Hruschka, E. R., & Gudwin, R. R. (2005b). Towards improving clustering ants: an adaptive ant clustering algorithm. *Informatica*, *29*, 143–154.

Vorhees, E. (1985). *The effectiveness and efficiency of agglomerative hierarchical clustering in document retrieval*. PhD thesis, Department of Computer Science, Cornell University, Ithaca, NY.

Zlochin, M., Birattari, M., Meuleau, N., & Dorigo, M. (2004). Model-based search for combinatorial optimization: a critical survey. *Annals of Operations Research*, *131*, 373–395.