An Adaptive Flocking Algorithm for Spatial Clustering

Gianluigi Folino and Giandomenico Spezzano

CNR-ISI

Via Pietro Bucci cubo 41C
c/o DEIS, UNICAL, 87036 Rende (CS), Italy
Phone: +39 984 831722, Fax: +39 984 839054
E-mail: spezzano@si.deis.unical.it

Abstract. This paper presents a parallel spatial clustering algorithm based on the use of new Swarm Intelligence (SI) techniques. SI is an emerging new area of research into Artificial Life, where a problem can be solved using a set of biologically inspired (*unintelligent*) agents exhibiting a collective intelligent behaviour. The algorithm, called SPARROW, combines a smart exploratory strategy based on a flock of birds with a density-based cluster algorithm to discover clusters of arbitrary shape and size in spatial data. Agents use modified rules of the standard flock algorithm to transform an agent (*boid*) into a hunter foraging for clusters in spatial data. We have applied this algorithm to two synthetic data sets and we have measured, through computer simulation, the impact of the flocking search strategy on performance. Moreover, we have evaluated the accuracy of SPARROW compared to the DBSCAN algorithm.

1 Introduction

Clustering spatial data is the process of grouping similar objects according to their distance, connectivity, or their relative density in space [1]. Spatial clustering has been an active area of research into data mining, with many effective and scalable clustering methods developed. These methods can be classified into partitioning methods [2], hierarchical methods [3], density-based methods [4], and grid-based methods [5]. Han, Kamber, and Tung's paper [6] is a good introduction to this subject.

Recently, other algorithms based on biological models have been proposed to solve the clustering problem. These algorithms are characterized by the interaction of a large number of simple agents sensing and changing their environment locally. They exhibit complex, emergent behaviour that is robust compared to the failure of individual agents. Ants colonies, flocks of birds, termites, swarms of bees etc. are agent-based insect models that exhibit a collective intelligent behaviour (swarm intelligence) [7] and may be used to define new algorithms of clustering.

In one of the first studies related to this domain, due to Deneubourg et al. [8], a population of ant-like agents randomly moving onto a 2D grid are allowed to move basic objects so as to classify them. This method was further developed by Lumer and Faietta [9] with simple objects that represent records in a numerical data set, and by

Kuntz and Snyers [10] who analyzed a real clustering problem in order to efficiently resolve an optimization problem. Monmarchè et al. [11] exploit this existing work from the knowledge discovery point of view with the aim of solving real world problems. They introduce a more robust heuristics based on stochastic principles of an ant colony in conjunction with the deterministic principles of the Kmeans algorithm. A flocking algorithm has been proposed by Macgill and S. Openshaw [12,13] as a form of effective search strategy to perform an exploratory geographical analysis. The method takes advantage of the parallel search mechanism a flock implies, by which if a member of a flock finds an area of interest the mechanics of the flock will drive other members to scan that area in more detail.

In this paper, we present a parallel spatial clustering algorithm SPARROW (***SPA****tial Cluste****R****ing Algo****R****ithm thr****O****ugh S****W****arm Intelligence*), which is based on an adaptive flocking algorithm combined with a density-based cluster algorithm, to discover clusters of arbitrary shape and size in spatial data. SPARROW uses the stochastic and exploratory principles of a flock of birds for detecting clusters in parallel according to the density-based principles of the DBSCAN algorithm, and a parallel iterative procedure to merge the clusters discovered.

SPARROW is a multi-agent algorithm where agents use modified rules of Reynolds' standard flock algorithm [14] to transform a *boid* into a hunter foraging for clusters in spatial data. Each agent searches the clusters in parallel and, by changing colour, signals the presence or the lack of significant patterns in the data to other flock members. The entire flock then moves towards the agents (*attractors*) that have discovered interesting regions, in order to help them, avoiding the uninteresting areas that are instead marked as obstacles. Moreover, each agent has a variable speed, though sharing a common minimum and maximum with the others. An agent will speed up in order to leave an empty or uninteresting region, whereas it will slow down in order to investigate an interesting region more carefully. The variable speed introduces an adaptive behaviour in the algorithm. In fact, the agents adapt their movement by changing their behaviour (speed) according to their previous experience represented by the agents which have stopped to signal an interesting region or an empty one.

We have built a Starlogo [15] simulation of SPARROW to investigate the interaction of the parameters that characterize the algorithm. The first experiments showed encouraging results and a better performance of SPARROW in comparison with the standard flock search and the linear randomised search.

The remainder of this paper is organized as follows: section 2 briefly presents the heuristics of the DBSCAN algorithm used for discovering clusters in spatial data, section 3 introduces the classical flocking algorithm and presents the SPARROW algorithm; section 4 discusses the obtained results while section 5 draws some conclusions and refers to future work.

2 The DBSCAN algorithm

One of the most popular spatial clustering algorithms is DBSCAN, which is a density-based spatial clustering algorithm. A complete description of the algorithm

and its theoretical basis is presented in the paper by Ester et al. [16]. In the following we briefly present the main principles of DBSCAN. The algorithm is based on the idea that all points of a data set can be regrouped into two classes: *clusters* and *noise*. Clusters are defined as a set of dense connected regions with a given radius (*Eps*) and containing at least a minimum number (*MinPts*) of points. Data are regarded as noise if the number of points contained in a region falls below a specified threshold. The two parameters, Eps and MinPts, must be specified by the user and allow to control the density of the cluster that must be retrieved. The algorithm defines two different kinds of points in a clustering: *core point*s and *non-core points*. A core point is a point with at least *MinPts* number of points in an Eps-neighborhood of the point. The non-core points in turn are either *border points* if are not core points but are density-reachable from another core point or *noise points* if they are not core points and are not density-reachable from other points. To find the clusters in a data set, DBSCAN starts from an arbitrary point and retrieves all points with the same density reachable from that point using Eps and MinPts as controlling parameters. A point *p* is density reachable from a point *q* if the two points are connected by a chain of points such that each point has a minimal number of data points, including the next point in the chain, within a fixed radius. If the point is a core point, then the procedure yields a cluster. If the point is on the border, then DBSCAN goes on to the next point in the database and the point is assigned to the noise. DBSCAN builds clusters in sequence (that is, one at a time), in the order in which they are encountered during space traversal. The retrieval of the density of a cluster is performed by successive spatial queries. Such queries are supported efficiently by spatial access methods such as R*-trees.

3 A multi-agent spatial clustering algorithm

In this section, we will present the SPARROW algorithm which combines the stochastic search of an adaptive flocking with the DBSCAN heuristics for discovering clusters in parallel. SPARROW replaces the DBSCAN serial procedure for clusters identification with a multi-agent stochastic search that has the advantage of being easily implementable on parallel computers and is robust compared to the failure of individual agents.

We will first introduce Reynolds' flock of birds model to describe the movement rules of the agents from which SPARROW takes inspiration. Then we will illustrate the details of the behavioral rules of the agents that move through the spatial data looking for clusters and communicating their findings to each other.

3.1 The flock algorithm

The flock algorithm was originally devised as a method for mimicking the flocking behavior of birds on a computer both for animation and as a way to study emergent behavior. Flocking is an example of emergent collective behavior: there is no leader, i.e., no global control. Flocking behavior emerges from the local interactions. In the flock algorithm each agent has direct access to the geometric description of the whole

scene, but reacts only to flock mates within a certain small radius. The basic flocking model consists of three simple steering behaviours:

Separation gives an agent the ability to maintain a certain distance from others nearby. This prevents agents from crowding too closely together, allowing them to scan a wider area.

Cohesion gives an agent the ability to cohere (approach and form a group) with other nearby agents. Steering for cohesion can be computed by finding all agents in the local neighbourhood and computing the *average position* of the nearby agents. The steering force is then applied in the direction of that *average position*.

Alignment gives an agent the ability to align with other nearby characters. Steering for alignment can be computed by finding all agents in the local neighbourhood and averaging together the 'heading' vectors of the nearby agents.

3.2 SPARROW: a flocking algorithm for spatial clustering

SPARROW is a multi-agent adaptive algorithm able to discover clusters in parallel. It uses a modified version of standard flocking algorithm that incorporates the capacity for learning that can find in many social insects. In our algorithm, the agents are transformed into hunters with a foraging behavior that allow them to explore the spatial data while searching for clusters.

SPARROW starts with a fixed number of agents that occupy a randomly generated position. Each agent moves around the spatial data testing the neighborhood of each location in order to verify if the point can be identified as a *core point*. In case it can, all points of the neighborhood of a core point are given a temporary label. These labels are updated as multiple clusters take shape concurrently. Contiguous points belonging to the same cluster take the label corresponding to the smallest label in the group of contiguous points.

Each agent follows the rules of movement described in Reynolds' model. In addition, our model considers four different kinds of agents, classified on the basis of the density of data in their neighborhood. These different kinds are characterized by a different color: *red*, revealing a high density of interesting patterns in the data, *green*, a medium one, *yellow*, a low one, and *white*, indicating a total absence of patterns. The main idea behind our approach is to take advantage of the colored agent in order to explore more accurately the most interesting regions (signaled by the red agents) and avoid the ones without clusters (signaled by the white agents). Red and white agents stop moving in order to signal this type of regions to the others, while green and yellow ones fly to find more dense clusters. Indeed, each flying agent computes its heading by taking the weighted average of alignment, separation and cohesion.

The following are the main features which make our model different from Reynolds':

- *Alignment* and *cohesion* do not consider yellow boids, since they move in a not very attractive zone.

- *Cohesion* is the resultant of the heading towards the average position of the green flockmates (centroid), of the attraction towards reds, and of the repulsion from whites, as illustrated in figure 1.
- A *separation* distance is maintained from all the boids, apart from their color.

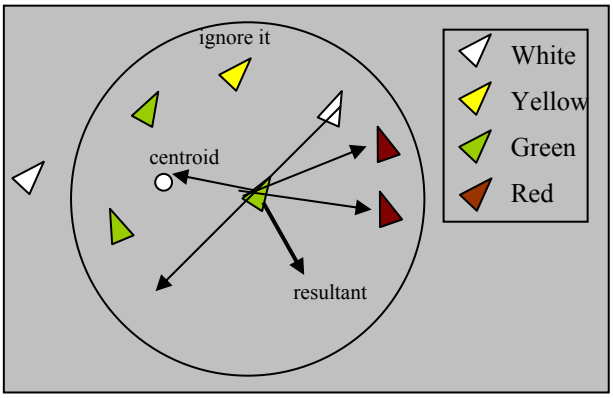

Fig. 1. Cohesion.

In the following we use the Starlogo language to describe our algorithm and to perform the simulations. SPARROW consists of a setup phase and a running phase shown in Figure 2. During the setup phase agents are created, data are loaded, some general settings are made and the turtles choose their color. In the running phase four distinct procedures are repeated by each turtle for a fixed number of times (*MaxNumberOfGenerations*). In fact, *ask-turtles* is a StarLogo instruction that makes all the turtles execute a procedure in parallel and waits for the completion of the operation before continuing.

The *choiceColor* procedure chooses the color and the speed of the boid with regard to the local density of the clusters in the data. It is based on the same parameters used in the DBSCAN algorithm: MinPts, the minimum number of points to form a cluster and Eps, the maximum distance that the boids can look at. In practice, the boid computes the density (*localdensity*) in a circular neighborhood (with a radius determined by its limited sight) and then executes the following instructions:

if localdensity > MinPts [**set** color red **set** speed 0]
if MinPts/4 < localdensity < MinPts [**set** color green **set** speed 1]
if 0 < localdensity < MinPts/4 [**set** color yellow **set** speed 2]
if localdensity = 0 [**set** color white **set** speed 0]

Thus, red and white boids will stop indicating interesting and desert regions to the others, while greens will move more slowly than yellows since they will explore denser zones of clusters. In the running phase, the yellow and green agents will compute their heading, according to the rules previously described, and will move following this direction and with the speed corresponding to their color. Afterwards, they will compute their new color, deriving from the movement. According to whether they have become red or white, a new boid will be generated in order to

maintain a constant number of turtles exploring the data. In case the turtle falls in the same position of an older it will die.

```
To setup
   import-data;
   load  the data and the clusters;
   create-turtles number ;
   create turtles in random positions
   . . . . . .
   ask-turtles [choiceColor]
   . . . . . .
end

To run
   repeat MaxNumberofGenerations [

  ask-turtles[if color = green or color = yellow [computeDir]

  ask-turtles[if color = green or color = yellow
      [move choiceColor
      if color = red or color = white
           [generateNewBoid
      if count-turtles-here > 1 [die]]]
           ask-turtles [if color = red [mergeCluster]]

  ask-turtles[if color = green or color = yellow
       [set age age + 1
     if age > maxLife [ generateNewBoid die ]]]

    ] ;end repeat

end ; run procedure
```

Fig. 2. Starlogo code of the setup and run procedure of Sparrow.

At this point red boids will run the mergeColor procedure, which will merge the neighboring clusters. The merging phase considers two different cases: when we have never visited points in the circular neighborhood and when we have points belonging to different clusters. In the first case, the points will be labeled and will constitute a new cluster; in the second case, all the points will be merged into the same cluster, i.e. they will get the label of the cluster discovered first.

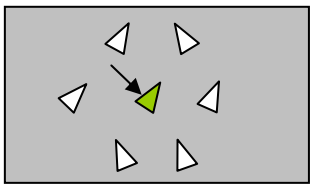

Fig. 3. The cage effect.

The last part of code invoked by ask-turtles was added to the original algorithm to avoid a ‘cage effect’ (see figure 3), which occurred during the first simulations; in

fact, some boids could remain trapped inside regions surrounded by red or white boids and would have no way to go out, wasting useful resources for the exploration. So, a limit was imposed on their life; hence, when their age exceeded a determined value (maxLife) they were made to die and were regenerated in a new randomly chosen position of the space.

4 Experimental results

We evaluated the accuracy of the solution supplied by SPARROW in comparison with the one of DBSCAN and the performance of the search strategy of SPARROW in comparison with the standard flocking search strategy and with the linear randomized search. Furthermore, we evaluated the impact of the number of agents on foraging for clusters performance.

To this purpose, we implemented the three different search strategies in Starlogo and compared their performance with a publicly available version of DBSCAN.

(a) GEORGE: 5463 points (b) DS4: 8000 points

Fig. 4. The two data sets used in our experiments.

For the experiments we used two synthetic data sets. The structure of these data sets is shown in figure 4(a) and 4(b). The first data set, called GEORGE, consists of 5463 points. The second data set, called DS4, contains 8843 points. Each point of the two data sets has two attributes that define the x and y coordinates. Furthermore, both data sets have a considerable quantity of noise. Table 1 and table 2 show, for the two data sets, the number of clusters and the number of points for each cluster found by DBSCAN and SPARROW and the relative error associated with each cluster.

Although DBSCAN and SPARROW produce the same results our experiments show that SPARROW can obtain the same number of clusters with a slightly smaller number of points for each cluster using a smaller number of spatial queries. The same results cannot be obtained by DBSCAN because of the different strategy of attribution of the points to the clusters. In particular, for the GEORGE data set each cluster found in SPARROW has a number of points that is about 2 percent lower than that discovered by DBSCAN and for the DS4 data set about the 3 percent. The spatial queries performed by SPARROW are for the GEORGE data set about the 27 percent of those performed by DBSCAN and for the DS4 dataset about the 45 percent.

Table 1. Number of clusters and number of points for clusters for GEORGE data set.

Number of clusters	Number of points for cluster (SPARROW)	Number of points for cluster (DBSCAN)	Relative error (percent)
1	832	848	-1.89%
2	690	706	-2.27%
3	778	800	-2.75%
4	782	815	-4.05%
5	814	818	-049%
6	712	718	-0.84%

Table 2. Number of clusters and number of points for clusters for DS4 data set.

Number of clusters	Number of points for cluster (SPARROW)	Number of points for cluster (DBSCAN)	Relative error (percent)
1	844	876	-3.65%
2	920	928	-0.86%
3	216	220	-1.82%
4	1866	1924	-3.01%
5	522	534	-2.25%
6	491	502	-2.19%
7	278	291	-4.47%
8	2308	2406	-4.07%
9	272	280	-2.86%

To verify the effectiveness of the search strategy we have compared SPARROW with the random-walk search (RWS) strategy of the standard flock algorithm and with the linear randomized search (LRS) strategy.

Figure 5 gives the number of clusters found through the three different strategies in 250 time steps for the DS4 data set. Figure 5 reveals that the number of clusters discovered at time step 65 from RWS and LRS strategy is slightly higher than that of SPARROW. From time step 66 to 110 the behavior of SPARROW is better than that of RWS but worse than LRS. SPARROW presents a superior behavior on both the search strategies after the 110 time step because of the adaptive behavior of the algorithm that allows agents to learn on their previous experience. A similar behaviour is also present in the GEORGE data set.

Finally, we present the impact of the number of agents on the foraging for clusters performance. Figure 6 gives, for the DS4 data set, the number of clusters found in 250 time steps for 25, 50 and 100 agents.

A comparative analysis reveals that a 100-agents population discovers a larger number of clusters than the other two populations with a smaller number of agents.

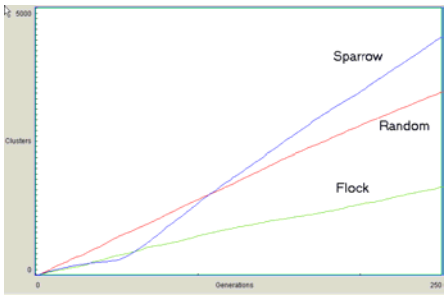

Fig. 5. Number of clusters found for the DS4 dataset.

This scalable behaviour of the algorithm determines a faster completion time because a smaller number of iterations are necessary to produce the solution.

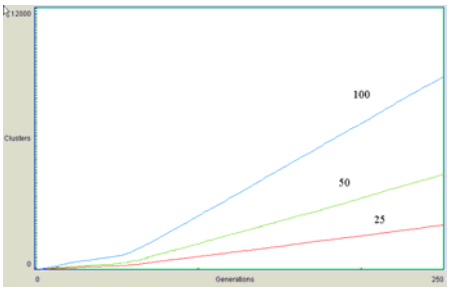

Fig. 6. The impact of the number of agents on foraging for clusters strategy.

5 Conclusions

In this paper, we have described the parallel clustering algorithm SPARROW, which is based on the use of swarm intelligence techniques. The algorithm combines a smart exploratory strategy based on a flock of birds with a density-based cluster algorithm to discover clusters of arbitrary shape and size in spatial data. The algorithm has been implemented in STARLOGO and compared with DBSCAN using two synthetic data sets. Measures of accuracy of the results show that SPARROW exhibits the same behaviour of DBSCAN although it needs a smaller number of spatial queries. Moreover, the adaptive search strategy of SPARROW is more efficient than those of the random-walk search (RWS) strategy of the standard flock algorithm and of the linear randomized search (LRS). Among the possible perspectives, we are currently testing SPARROW using real data sets such as the raster data of the AMBIENTE GIS concerning the landslides events that have occurred in the Campania Region in May 1998. In addition, we are studying how to parallelize SPARROW on a Linux cluster in order to tackle large size problems.

References

1. Han J., Kamber M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann 2000.
2. Kaufman L., Rousseeuw P. J., *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, 1990.
3. Karypis G., Han E., Kumar V.,: CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling, *IEEE Computer*, vol. 32, pp.68-75, 1999.
4. Sander J., Ester M., Kriegel H.-P., Xu X.: *Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications*, in: Data Mining and Knowledge Discovery, an Int. Journal, Kluwer Academic Publishers, Vol. 2, No. 2, 1998, pp. 169-194.
5. Wang W., Yang J., Muntz R., STING: A Statistical Information Grid Approach to Spatial Data Mining, *Proc. of Int. Conf. Very Large Data Bases (VLDB'97)*, pp. 186-195, 1997.
6. Han J., Kamber M., Tung A.K.H., *Spatial Clustering Methods in Data Mining: A Survey*, H. Miller and J. Han (eds.), Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2001.
7. Bonabeau E., Dorigo M., Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems,* Oxford University Press, 1999.
8. Deneubourg J. L., Goss S., Franks, N., Sendova-Franks A., Detrain C., and Chretien L., The Dynamic of Collective Sorting Robot-like Ants and Ant-like Robots, Proc. of the first Conf. on Simulation of Adaptive Behavior, J.A. Meyer et S.W. Wilson (Eds), MIT Press/Bradford Books, pp. 356-363, 1990.
9. Lumer E. D., Faieta B., Diversity and Adaptation in Populations of Clustering Ants, *Proc. of the third Int. Conf. on Simulation of Adaptive Behavior: From Animals to Animats (SAB94)*, D. Cliff, P. Husbands, J.A. Meyer, S.W. Wilson (Eds), MIT-Press, pp. 501-508, 1994.
10. Kuntz P. Snyers D., Emergent Colonization and Graph Partitioning, *Proc. of the third Int. Conf. on Simulation of Adaptive Behavior: From Animals to Animats (SAB94)*, D. Cliff, P. Husbands, J.A. Meyer, S.W. Wilson (Eds), MIT-Press, pp. 494-500, 1994.
11. N. Monmarché, M. Slimane, and G. Venturini, "On improving clustering in numerical databases with artificial ants", in *Advances in Artificial Life: 5th European Conference*, ECAL 99, LNCS 1674, Springer, Berlin, pp. 626-635, 1999.
12. Macgill, J., Openshaw, S., The use of Flocks to drive a Geographic Analysis Machine, in *Proceedings of the 3rd International Conference on GeoComputation*, University of Bristol, United Kingdom, 1998.
13. James Macgill, Using Flocks to Drive a Geographical Analysis Engine, *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*, MIT Press, Reed College, Portland, Oregon, pp. 1-6, 2000.
14. Reynolds C. W., *Flocks, Herds, and Schools: A Distributed Behavioral Model*, Computer Graphics vol. 21, n. 4, , (SIGGRAPH 87), pp. 25-34, 1987.
15. V. S. Colella, E. Klopfer, M. Resnick, *Adventures in Modeling: Exploring Complex, Dynamic Systems with StarLogo,* Teachers College Press, 2001.
16. Ester M., Kriegel H.-P., Sander J., Xu X., A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96)*, Portland, OR, 1996, pp. 226-231, 1996.