Fuzzy c–Means Herding

Thomas A. Runkler

Siemens Corporate Technology, Munich, Germany
Email: Thomas.Runkler@siemens.com

Abstract— *Herding is the process of bringing individuals (e.g. animals) together into a group. More specifically, we consider self–organized herding as the process of moving a set of individuals to a given number of locations (cluster centers) without any external control. We formally describe the relation between herding and clustering and show that any clustering model can be used to control herding processes. For the specific case of the fuzzy c–means model we derive the equations of the fuzzy c–means herding algorithm using a gradient descent approach with limited step size. Several experiments related to an autonomous mobile robot scenario show that fuzzy c–means herding yields smooth trajectories, well–balanced clusters, and fast convergence.*

Keywords— Fuzzy clustering, herding, robot swarms, swarm intelligence.

1 Introduction

This paper deals with the problem of *online herding* of n individuals specified by the state vectors $x_1, \ldots, x_n \in \mathbb{R}^p$, into c clusters. Application examples for herding are illustrated by the following four scenarios:

1. A swarm of autonomous micro–robots explores the Mars surface. Just before their individual energy supplies are exhausted, the robots move to a couple of individual locations from where they are collected and returned to the spaceship.
2. Museum visitors individually walk through an exhibition. At the beginning of the guided tours, each visitor receives instructions where to go to join one of the tour groups.
3. Mountain hikers have been caught by a severe thunderstorm. They receive instructions on their mobile phones where to go, so that they can be picked off from several evacuation points by helicopter.
4. Wireless sensor devices are attached to individual components of an industrial plant for condition monitoring purposes. At power on, each sensor automatically adapts its communication protocol so that several clusters of networked sensors emerge.

Herding mobile robots (as in our first scenario) is a very active field in the robotics community. A neuro–fuzzy control approach for herding (also called convoying) of multiple robots in presented in [1]. A similar approach to herding very heterogeneous types of robots is presented in [2]. Also herding in evacuation processes (as in our third scenario) is currently actively discussed in the scientific community, for example in [3]. Herding has also been applied to swarm based learning processes in order to balance exploration and exploitation behavior [4]. Some work has also been done concerning herding using outside agents [5], for example the so–called *dog and sheep* herding [6]. In this paper, however, we only consider *self–organizing* herding, where all individuals equally contribute to the herding process, without any external control.

In each of the four scenarios presented above, the individual state vectors $x_1, \ldots, x_n$ (for example the locations of the micro–robots) are successively updated, until a clustering is achieved, i.e. each of the individuals x_k is (fuzzily) assigned to cluster i with a membership $u_{ik} \in [0, 1]$. When the herding process is converged, each state vector is equal to the prototypical state $v_i \in \mathbb{R}^p$, $i = 1, \ldots, c$, of its corresponding cluster, and is crisply assigned to this cluster, $u_{ik} = 1$ and $u_{jk} = 0$ for all $j \neq i$. More formally, herding can be described by a sequence

$$X \to (U, V, X) \to (U, V, X) \to (U, V, X) \to \ldots \quad (1)$$

which converges to $X = V$ and $U \in M_{hcn}$,

$$M_{hcn} = \left\{ U \in \{0,1\}^{c \times n} \;\middle|\; \sum_{i=1}^{c} u_{ik} = 1, \quad k = 1, \ldots, n, \quad \sum_{k=1}^{n} u_{ik} > 0, \quad i = 1, \ldots, c \right\} \quad (2)$$

In practical scenarios we will not be able to run this algorithm for an infinite time until convergence, but will terminate after a previously specified number of t steps, or when an appropriate termination criterion holds, for example when the maximum change in the state vectors is lower than a previously specified threshold. Even though the limit of the final partition is crisp and somewhat trivial (because $X = V$), we pursue a fuzzy approach here to *smoothly* control the herding process.

Apparently, herding is closely related to clustering. A discussion of the relation between herding and clustering from a theoretical physics perspective can be found in [7]. Just as herding, clustering also assigns each object to a cluster, and each cluster can be described by a prototype as well, but clustering does not change the data vectors X. More formally, in contrast to (1) clustering is described by a sequence

$$X \to (U, V) \to (U, V) \to (U, V) \to \ldots \quad (3)$$

where the initial X remains constant. Despite this difference between clustering and herding, the herding process can be performed by optimizing *any* conventional clustering model. The main difference is that in clustering we consider (U, V)

ISBN: 978-989-95079-6-8

as the set of free variables for optimization, and in herding we consider (U, V, X). In this paper we illustrate this for the case of the *fuzzy c–means* [8] clustering model. Other suitable clustering models are, for example, hard c–means [9], possibilistic c–means [10], noise clustering [11], or high–dimensional fuzzy c–means [12].

This paper is structured as follows: Sections 2 and 3 briefly repeat the fuzzy c–means objective function and how this model is applied to clustering. Section 4 shows how the fuzzy c–means model can be used to control a herding process, leading to the fuzzy c–means herding algorithm. Section 5 presents experiments with the fuzzy c–means herding algorithm referring to the Mars robot scenario illustrated above. Section 6 finally summarizes the conclusions and gives an outline for future work in this field.

2 The Fuzzy c–Means Objective Function

The *fuzzy c–means* objective function [8] is a dissimilarity–based measure for the clustering of a data set $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ into $c \in \{2, \ldots, n-1\}$ clusters specified by a partition matrix $U \in M_{fcn}$ where

$$M_{fcn} = \left\{ U \in [0,1]^{c\times n} \;\middle|\; \sum_{i=1}^{c} u_{ik} = 1, \quad k = 1, \ldots, n, \quad \sum_{k=1}^{n} u_{ik} > 0, \quad i = 1, \ldots, c \right\} \tag{4}$$

and a set of prototypes (cluster centers) $V = \{v_1, \ldots, v_c\} \subset \mathbb{R}^p$. The FCM objective function is the sum of the dissimilarities between data and prototypes, weighted by the (fuzzy) memberships. Using a fuzziness index $m > 1$, it is formally defined as

$$J(U, V, X, c, m) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m \, |x_k - v_i|^2. \tag{5}$$

For $m \to 1$ this objective function converges to the *hard c–means objective function* [9]. For many applications good results were reported for $m = 2$. In this paper we always use the Euclidean distance $|.| = \|.\|$. Lower values of J indicate a better clustering of (U, V, X). The theoretical minimum of J is zero. This minimum is achieved, if for each $k = 1, \ldots, n$ we have an $i \in \{1, \ldots, c\}$ with $x_k = v_i$, $u_{ik} = 1$ and $u_{jk} = 0$ for all $j \neq i$, i.e. $X = V$ and $U \in M_{hcn}$. This means that herding converges to $J = 0$.

3 Fuzzy c–Means Clustering

As pointed out above we assume X to be given and fixed in clustering and only find U and V. In fuzzy c–means clustering we find U and V by minimizing J. In clustering we are usually not able to reduce J to zero. We may use different optimization methods such as genetic algorithms [13, 14], particle swarm optimization [15, 16, 17], ant colony optimization [18], or wasp swarm optimization [19]. In this paper we apply direct minimization using the necessary conditions for local extrema or saddle points of J. An equation to compute the optimal U for a given V under the constraints in (4) is obtained by defining the Langrangian

$$L = J - \sum_{k=1}^{n} \lambda_k \left(1 - \sum_{i=1}^{c} u_{ik} \right) \tag{6}$$

and setting $\partial L / \partial u_{ik} = 0$, from which follows

$$u_{ik} = 1 \Bigg/ \sum_{j=1}^{c} \left(\frac{|x_k - v_i|}{|x_k - v_j|} \right)^{\frac{2}{m-1}}, \tag{7}$$

$i = 1, \ldots, c$, $k = 1, \ldots, n$. An equation to compute the optimal V for a given U is obtained by setting $\partial J / \partial v_i = 0$, from which follows

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m}, \tag{8}$$

$i = 1, \ldots, c$. The equations for V depend on U and the equations for U depend on V, so optimization can be done in an alternating scheme, the so–called alternating optimization. The clustering algorithm terminates when an appropriate termination criterion holds, e.g. when successive changes in V are lower than a threshold ε. The fuzzy c–means clustering algorithm can then be formally defined as follows:

Algorithm 1: Fuzzy c–Means Clustering.

```
input X, c, m
randomly initialize V ∈ ℝ^p
repeat
  update U(V,X,m) by (7)
  update V(U,X,m) by (8)
until max max ΔV < ε
output U, V
```

4 Fuzzy c–Means Herding

As pointed out in the introduction, herding does not only update U and V at each optimization step but also X. If we do the optimization of $J(U, V, X)$ in an alternating scheme, then we also have to determine an appropriate update equation for X. In the scenarios described above we do not arbitrarily change the values of X (for example the position of the robots) in each step but we can only *gradually* change X (for example because the robots can only move with a limited speed). Therefore, in contrast to the *direct* optimization of U and V in clustering, we only apply a *gradient descent* of X with a limited step size $s \in \mathbb{R}^+$ in herding. The step size s divided by the time for one step represents the maximum speed of the individuals, e.g. robots. The direction vectors for the gradient descent on X are obtained as

$$\partial J / \partial x_k = \Delta x_k = -\sum_{i=1}^{c} 2 u_{ik}^m \, (x_k - v_i), \tag{9}$$

$k = 1, \ldots, n$. So, the update equations for X are

$$x_k = x_k + \frac{\min\{s, \|\Delta x_k\|\}}{\|\Delta x_k\|} \cdot \Delta x_k, \tag{10}$$

ISBN: 978-989-95079-6-8

$k = 1, \ldots, n$. This makes sure that the individuals move in the direction of the gradient but that the step length never exceeds the threshold s. The herding algorithm terminates when all elements of X have approached the corresponding elements of V, i.e. when $\max \Delta X < s$. The fuzzy c–means herding algorithm can then be formally defined as follows:

Algorithm 2: Fuzzy c–Means Herding.

input X, c, m, s
randomly initialize $V \in \mathbb{R}^p$
repeat
  update $U(V, X, m)$ by (7)
  update $V(U, X, m)$ by (8)
  update $X(U, V, m)$ by (10) and (9)
until $\max \|\Delta X\| < s$
output U, V, X

5 Experiments

We illustrate the operation of the fuzzy c–means herding algorithm using an artificial data set that is inspired by the autonomous robot scenario presented at the beginning of section 1. We randomly generate a set X of $n = 100$ data vectors of the dimension $p = 2$ using a Gaussian distribution with mean zero and variance one. This data set is illustrated in Fig. 1. The idea is that these data vectors represent the current positions of the exploring robots in the two–dimensional plane just before the herding process begins. Then, we start the fuzzy c–means herding algorithm with a fuzziness index of $m = 2$, a step size $s = 0.05$ (representing the speed of the robots), and different numbers of clusters $c \in \{2, 3, 4, 5\}$. Each experiment is performed for the same initial vectors X and V. Additional experiments with other random initializations of X and V (not presented in detail here) produced very similar results. The trajectories of the vectors V and X are shown in Fig. 2. The solid curves in these figures show the trajectories of X. Each of these trajectories starts at one of the initial points of X (as shown in Fig. 1), smoothly moves through the two–dimensional plane, and terminates in one of the final cluster centers. The data vectors do not directly move towards one of the cluster centers, but are apparently initially fuzzily assigned to several clusters, have a tendency to move towards the mean of the data set and then gradually focus on one of the cluster centers. We counted the number of steps t until all of the data points eventually reached one of the cluster centers and obtained the following table:

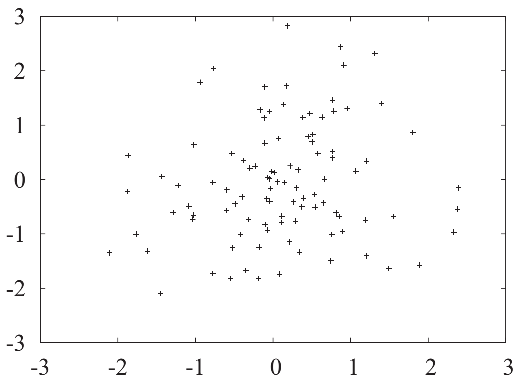


Figure 1: Initial data set (robot positions).

Table 1: Convergence steps for fuzzy c–means herding, $m = 2$.

c	2	3	4	5
t	34	29	25	23

A detailed study of the step size showed that lower values of s produce smoother trajectories and lead to higher numbers of convergence steps t. Correspondingly, higher values of s produce rather coarse trajectories and lead to lower numbers of convergence steps t. Despite the influence of s on the absolute values of t, the relation between the convergence steps for the different experiments are almost independent of s.

The ticks $\times$ in the figures show the trajectories of V. The cluster centers in V are not fixed but also move through the two–dimensional plane with an approximately decreasing step size as the herding process proceeds. The data points that are assigned to each cluster originally covered roughly the same area in the two–dimensional plane. Notice that a similar effect can be observed with the fuzzy c–means clustering algorithm.

The reader might wonder if it wasn't more efficient to arbitrarily pick fixed cluster centers, say c of the initial data vectors in X, assign each data point to the closest data center and let each data point directly move towards its pre–assigned data point. The trajectories obtained by this experiment are shown in Fig. 3. The trajectories of X (solid curves) form stars whose centers are the initially randomly chosen cluster centers. Unlike the clusters obtained by fuzzy c–means herding, the data points in each of these clusters do not represent similar areas, but some clusters obtain only very few, and others obtain many data points, depending on the cluster centers chosen. Apparently, the linear trajectories of some of the data points are considerably shorter than those obtained by fuzzy c–means herding. However, some of the trajectories are considerably longer than their herding equivalents. Also for this experiment we counted the number of steps t until all of the data points reached their closest cluster centers:

Table 2: Convergence steps for pre–assigned cluster centers (one specific run).

c	2	3	4	5
t	35	29	27	26

Notice that fuzzy c–means herding yields the same or even lower values of t for all numbers of clusters. However, this result depends on the particular choices of the cluster centers. Therefore, we repeated each of these experiments with 1000 different random choices of cluster centers and computed the numbers of steps until termination. The histograms of these step numbers are shown in Fig. 4 and the mean numbers of steps $\bar{t}$ are given in the following table:

Table 3: Convergence steps for pre–assigned cluster centers (average of 1000 runs).

c	2	3	4	5
$\bar{t}$	33.9	30.3	28.0	26.4

The average number of steps is almost equal or larger than the number of steps needed for fuzzy c–means herding with

ISBN: 978-989-95079-6-8

$m = 2$. In order to reduce the number of convergence steps in fuzzy c–means herding we reduce the value of the fuzziness parameter m, so the individual data points are less fuzzily assigned to the clusters in the beginning and more directly orientate towards their corresponding cluster centers. To avoid numerical problems that sometimes occur when m is very close to one, we pick $m = 1.1$. For this case, the trajectories of V and X are shown in Fig. 5. Just as in Fig. 2 and in contrast to Fig. 3 each cluster corresponds to approximately the same area in the two–dimensional plane. The trajectories are a little less smooth than in Fig. 2 and rather resemble the star–like trajectories in Fig. 3. This observation is corroborated by the fact that the number of convergence steps is much lower than in Fig. 2, as shown in the following table:

Table 4: Convergence steps for fuzzy c–means herding, $m = 1.1$.

c	2	3	4	5
t	28	22	20	18

A comparison of Table 4 with Table 3 reveals the fact that the number of convergence steps for fuzzy c–means herding is even lower than (on average) for pre–assigned cluster centers. We marked the numbers of convergence steps of fuzzy c–means herding for $m = 2$ and for $m = 1.1$ as vertical lines in Fig. 4. Apparently, for $m = 1.1$ the number of steps is very close to the minimum of the histograms, i.e. it compares with the number of steps when the clusters by chance have been approximately optimally assigned. In other words, fuzzy c–means herding implicitly finds an (almost) optimal cluster assignment, leads to a considerably low number of convergence steps, and yields smoothly herding trajectories at the same time.

6 Conclusions

In this paper we have shown that minimization of clustering models can be used to control herding processes. In particular, we have focussed on herding by minimization of the fuzzy c–means clustering model. Th experiments with a gradient descent of the data vectors with limited step size can be illustrated by a scenario with automous mobile robots that have to move to a certain number of locations (cluster centers) in a self–organizing manner. The experiments with fuzzy c–means herding yield very smooth trajectories. The individuals that are collected at each cluster center approximately represent the same area (or, more generally, hypervolume) of the initial state space. This means that the final clusters are nicely balanced. Compared with herding with pre–defined cluster centers, fuzzy c–means herding on average needs less steps until termination. For low values of the fuzziness index the trajectories become less smoother, but the number of steps is further reduced. For $m = 1.1$ the number of steps is even close to the global minimum. Thus, fuzzy c–means herding is an attractive herding algorithm that yields smooth trajectories, well–balanced clusters, and fast convergence.

References

[1] K. C. Ng and M. M. Trivedi. Multirobot convoying using neuro-fuzzy control. In *International Conference on Pattern Recognition*, pages 417–421, Vienna, Austria, August 1996.

[2] L. E. Parker, B. Kannan, X. Fu, and Y. Tang. Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2488–2493, Las Vegas, October 2003.

[3] A. Kirchner and A. Schadschneider. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A*, 312(1–2):260–276, 2002.

[4] M. R. Akbarzadeh, H. Rezaei, and M. B. Naghibi. A fuzzy adaptive algorithm for expertness based cooperative learning, application to herding problem. In *International Conference of the North American Fuzzy Information Processing Society*, pages 317–322, Chicago, July 2003.

[5] J. M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodriguez, and N. M. Amato. Shepherding behaviors. In *IEEE International Conference on Robotics and Automation*, pages 4159–4164, New Orleans, April 2004.

[6] P. Kachroo, S. A. Shedied, J. S. Bay, and H. Vanlandingham. Dynamic programming solution for a class of pursuit evasion problems: the herding problem. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 31(1):35–41, 2001.

[7] D. Costantini, S. Donadio, U. Garibaldi, and P. Viarengo. Herding and clustering: Ewens vs. Simon-Yule models. *Physica A*, 355(1):224–231, 2005.

[8] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[9] G. B. Ball and D. J. Hall. Isodata, an iterative method of multivariate analysis and pattern classification. In *IFIPS Congress*, 1965.

[10] R. Krishnapuram and J. M. Keller. A possibilistic approach to clustering. *IEEE Transactions on Fuzzy Systems*, 1(2):98–110, May 1993.

[11] R. N. Davé. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12:657–664, 1991.

[12] T. A. Runkler. A single objective fuzzy clustering model for very high–dimensional data. In *GMA/GI Workshop Fuzzy Systems and Computational Intelligence, Dortmund, Germany*, pages 234–243, December 2008.

[13] J. C. Bezdek and R. J. Hathaway. Optimization of fuzzy clustering criteria using genetic algorithms. In *IEEE Conference on Evolutionary Computation, Orlando*, volume 2, pages 589–594, June 1994.

[14] F. Klawonn and A. Keller. Fuzzy clustering with evolutionary algorithms. *International Journal of Intelligent Systems*, 13(10–11):975–991, 1998.

[15] C. Y. Chen and F. Ye. Particle swarm optimization and its application to clustering analysis. In *International Conference on Networking, Sensing and Control*, pages 789–794, Taipeh, Taiwan, March 2004.

[16] X. Cui, T. E. Potok, and P. Palathingal. Document clustering using particle swarm optimization. In *IEEE Swarm Intelligence Symposium, Pasadena*, pages 185–191, June 2005.

[17] T. A. Runkler and C. Katz. Fuzzy clustering by particle swarm optimization. In *IEEE International Conference on Fuzzy Systems*, pages 3065–3072, Vancouver, July 2006.

[18] T. A. Runkler. Ant colony optimization of clustering models. *International Journal of Intelligent Systems*, 20(12):1233–1261, December 2005.

[19] T. A. Runkler. Wasp swarm optimization of the c-means clustering model. *International Journal of Intelligent Systems*, 23(3):269–285, February 2008.

ISBN: 978-989-95079-6-8

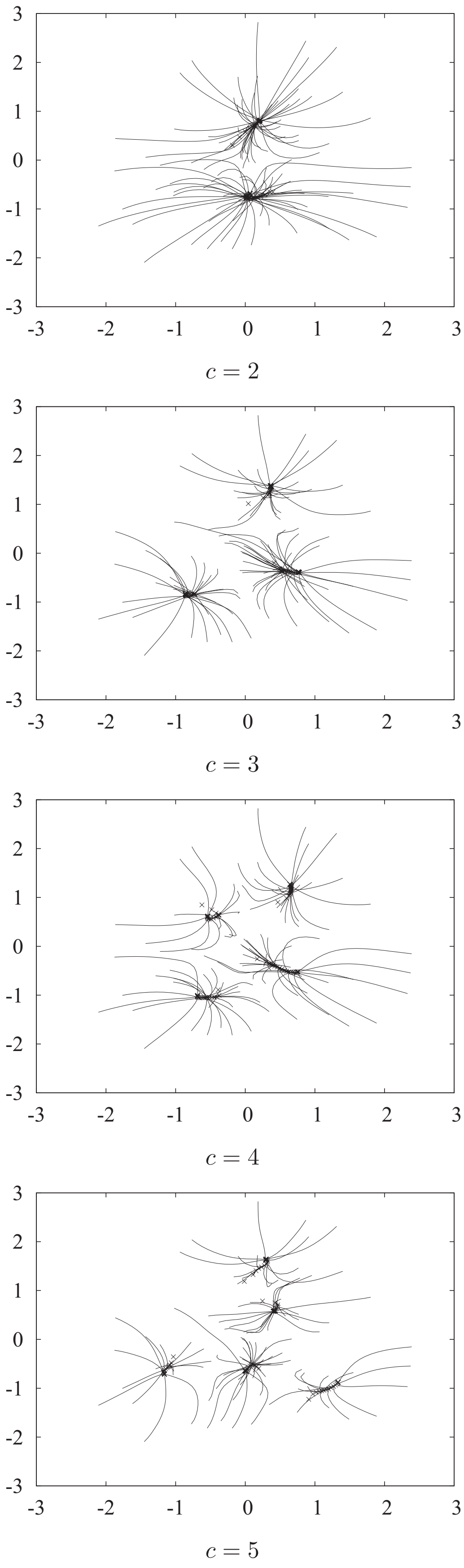


Figure 2: Herding trajectories for $m = 2$.

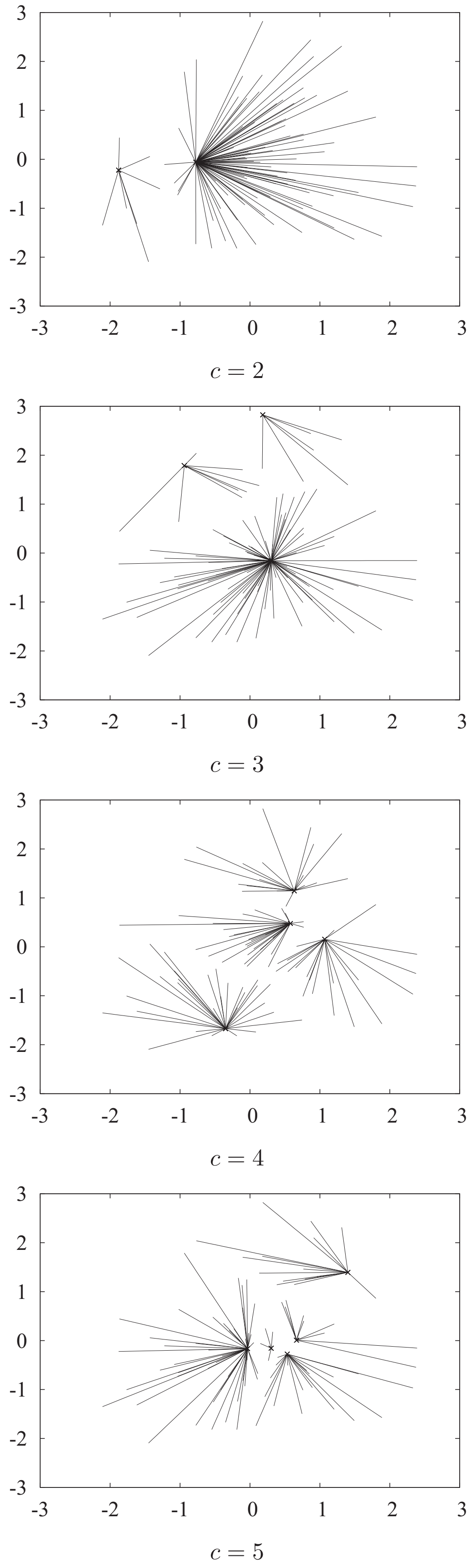
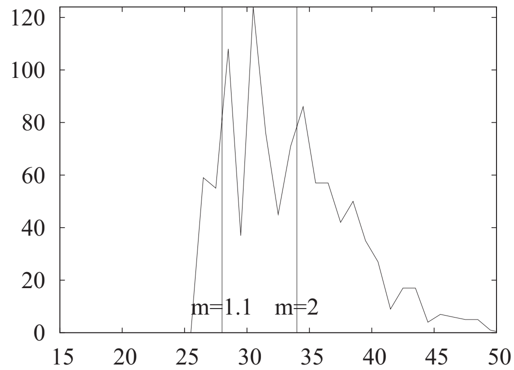
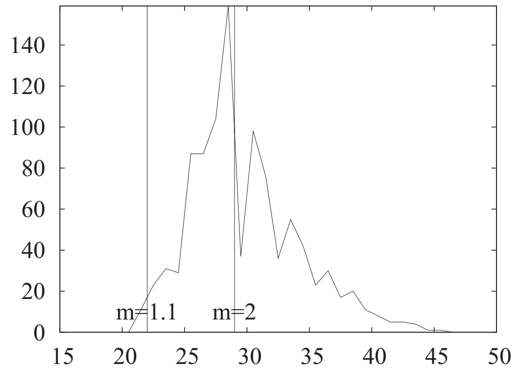


Figure 3: Trajectories with pre-assigned arbitrary cluster centers.

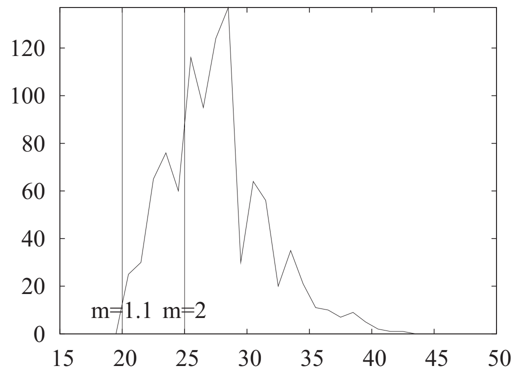
ISBN: 978-989-95079-6-8



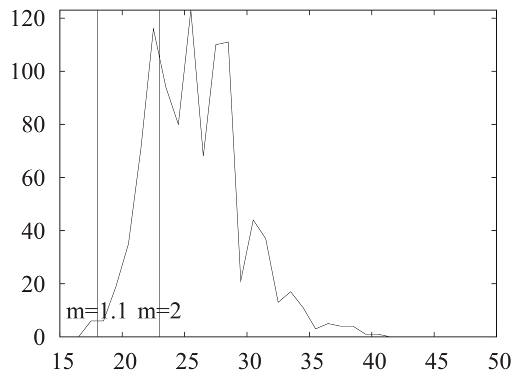
$c = 2$



$c = 3$

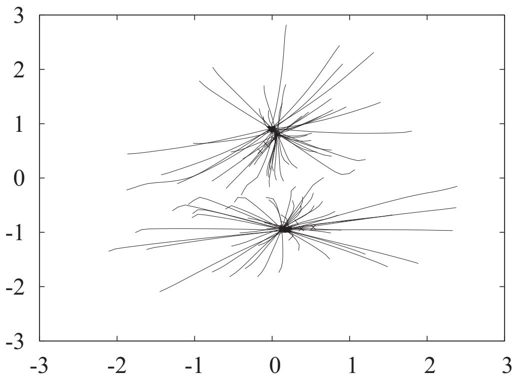


$c = 4$

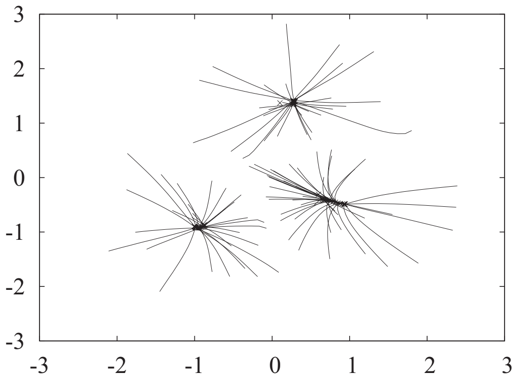


$c = 5$

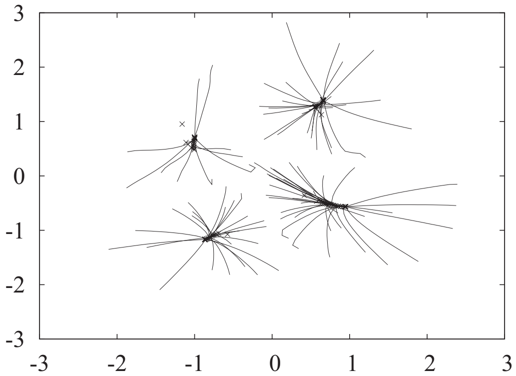
Figure 4: Histograms of the number of steps for preassigned cluster centers.



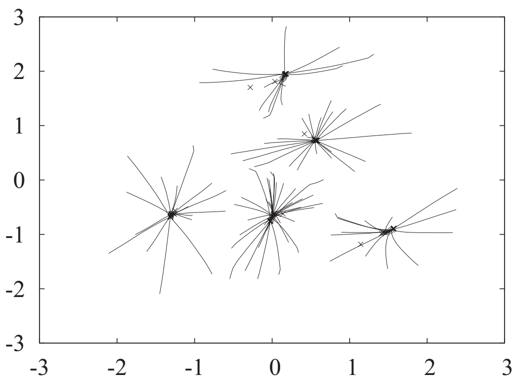
$c = 2$



$c = 3$



$c = 4$



$c = 5$

Figure 5: Herding trajectories for $m = 1.1$.

ISBN: 978-989-95079-6-8