

Cellular Ants: Combining Ant-Based Clustering with Cellular Automata

Andrew Vande Moere & Justin James Clayden

Key Centre of Design Computing & Cognition, University of Sydney, Australia
{andrew,justin}@arch.usyd.edu.au

Abstract

This paper proposes a novel data clustering algorithm, coined ‘cellular ants’, which combines principles of cellular automata and ant colony optimization algorithms to group similar multi-dimensional data objects within a two-dimensional grid. The proposed method assigns data objects to unique ants, which actively move around, leave pheromones and follow trails of similar ants. Cellular automata principles based on simple, discrete neighborhood densities determine an ant’s directional movements, so that clusters emerge. The novel concept of ‘positional swapping’ organizes these clusters internally based on multi-dimensional data value similarity. As a result, shared cluster borders in grid space contain data objects that are nearby in parameter space. This method is algorithmically simple, as it is based on a few user-chosen variables and uses fixed discrete values instead of probability algorithms. This clustering technique is evaluated using several datasets, while its methodology and computational performance is compared to similar approaches.

1. Introduction

The process of clustering attempts to partition a dataset into specific meaningful subsets, by categorizing or grouping similar data items together. Several data clustering techniques already exist that are distinguished by algorithmic principles, data types, cluster shapes or specific user-determined variables. Most current approaches in ant-based data clustering mainly focus on algorithmic optimizations, and often do not consider the comprehensibility of the resulting data representation: although the resulting grid-based constellations show spatially separated groups of data types, visual aspects such as cluster size, the relative position in relation to one another and the internal

spatial organization within clusters often do not convey any meaningful information to users. Motivated by this observation, the cellular ant method proposes an initial approach to explore the potential benefits of self-organizing algorithms that are capable of clustering multi-dimensional datasets as well as creating meaningful and comprehensible data visualizations.

This paper presents a relatively simple approach that only requires two user-chosen parameters: a data value similarity threshold t and a grid size. The proposed cellular ant algorithm assumes that all ants are fully *decentralized* (e.g. ants can only communicate with other ants in their immediate neighborhood), have no information about the dataset (e.g. number of data types), have a limited perception of the world, and can only move one cell for each iteration. To prove the validity and simplicity of the proposed method, the current implementation does not dynamically alter any variable during the clustering itself, such as an increase of the perception range or the data similarity threshold. Obviously, these constraints will lead to an initially computationally less ‘optimized’ algorithm, but rather prove the potential qualities of combining ant-based clustering with cellular automata principles.

In comparison to multi-agent clustering and ant-based data mining techniques, the proposed approach enhances the visual representation by internally and spatially ordering the clusters according to data value similarity, instead of creating an arbitrary constellation of spatially separate and internally unordered groups. The cellular ants method combines the concepts of: 1) pheromone trails to rapidly derive cluster approximations; 2) grid cell-based cellular automata rules that increase cluster density; and 3) a novel swapping rule that allows ants to compare their data values with two of its neighbors, so that their grid positions can be swapped if their relative parameter space positions are not optimal. The combination of these three driving forces generates internally organized, meaningful representations of multi-

dimensional datasets in an unsupervised way and with a good performance. The main purpose of cellular ants is thus to self-organize data as visual representations that are more comprehensible than standard ant-based clustering techniques, so that similarities between ants of nearby clusters (e.g. shared borders), or patterns within clusters can be easily perceived.

2. Related Work

2.1. Agent-Based Visualization

Information visualization in the context of data mining is mostly limited to presenting data in a conventional manner to display agent data attributes [7, 13]. Multi-agent system visualization represents intrinsic relations between agents to generate data representations for monitoring and engineering purposes [15]. Multi-agent systems have also been used to organize the structural data flow in order to generate information visualizations of complex fuzzy systems [11]. Rule-based data visualization maps simple behavior rules to time-varying data objects, so that data alterations are interpreted as dynamic actions. The infoticle (information-particle) method uses a simple decentralized multi-agent system and rule-based behaviors to map time-varying data tendencies of stock market quotes into motion typologies as an emergent outcome of swarming and flocking [16].

2.2. Cellular Automata

Cellular automata (CA), a method originally proposed by Ulam and Von Neumann, consists of a number of cells that evolve by simple local rules governed by a discrete timeline [17]. Cellular automata rules determine the *state* (e.g. alive or dead) of each ‘cell’, which generally depends on the states of its neighboring cells. In contrast, ant-based simulations do not consider cell states, but the characteristics of ants and their perception of the world. Ants are able to roam inside and ‘actively’ sense the states within the environment, which determine their movements (e.g. direction and speed), actions (e.g. pick up or drop) and internal states (e.g. sleep or alive). CA simulations are mostly determined by environmental states, while ants can ‘act’ upon the environment and even change it to some degree, for instance by leaving pheromone trails. The cellular ant approach applies typical CA rules, traditionally used for grid cell states, to the perception and reasoning of ants: ants decide their actions depending on the discrete amount of ‘similar’ ants in their neighborhood, rather than using probabilistic mathematical functions.

2.3. Stigmergy

Real ants use pheromone trails to determine the shortest route to a specific destination, a phenomenon that has been successfully simulated by artificial ants [4]. The ant colony optimization heuristic simulates the ability of real ants to drop and follow pheromone trails [1]. The presence of pheromones enables the concept of *stigmergy*, a form of indirect communication used by social insects to coordinate their activities, usually by changing the environment with interpretable cues. The cellular ant method uses pheromones trails to allow ants to ‘sense’ and move towards similar ants within the world, hereby effectively overcoming their relatively limited one-cell sensorial perception.

2.4. Ant-Based Clustering

Ant-based data mining combines the nest-cleaning characteristics of ant colonies with the task of data clustering. The ant clustering algorithm starts from a toroidal grid on which data objects are randomly scattered. Ants pick up data items and move around in randomly chosen directions. Ants then probabilistically decide whether to drop the data item, preferably in the vicinity of similar data items. A specific *object distance measure* variable α determines the degree of similarity between pairs of data objects, so that dissimilar items will not be placed together and similar items will be clustered. The optimal value for α cannot be determined without prior knowledge of the dataset, unless its value is adaptable [6]. Different successful ant-based clustering approaches currently exist, which incorporate fuzzy-set theory [14], topographic maps [5], or genetic algorithms [12].

Traditional ant clustering algorithms consider data objects as ‘lifeless’ entities that are moved around by foraging ants [3, 8, 10]. The cellular ant method conceptually differs from these by considering the ants themselves as unique data objects that are capable of ‘actively’ moving around in search for similar ants to stay close to within the visualization space. The mapping of each data object directly on unique ants is not often used in ant-based clustering techniques, although some recent examples exist, such as those inspired by the chemical recognition system of ants [9]. This simple ‘data-to-ant’ mapping model generally will lead to less required iterations, as for each iteration step, all data objects are able to simultaneously move to increasing ideal positions. This approach generally requires more calculation time per iteration step, as all data objects simultaneously demand the execution of computing-intensive data similarity algorithms between pairs of neighboring ants.

Although unknown during the original conception of our method, Chen et al. [2] recently presented a conceptually similar approach for data clustering, called the Ants Sleeping Model (ASM). It also uses cellular automata principles and represents each unique data object as a single ant. The cellular ant algorithm differs from the ASM method in following aspects:

- **Discrete Similarity Discrimination:** neighboring cellular ants are considered to be ‘similar’ when their distance in parameter space is smaller than a *fixed* similarity threshold value, instead of using other well-known continuous similarity algorithms, such as proposed by Deneubourg et al. (1990), specific fitness determinations or other probability calculations.
- **Sleep State:** no concept of internal ant states is used.
- **State-Based Consideration:** cellular ant actions are mainly determined by the discrete number of similar ants in their neighborhood, which is very similar to *cell state rules* defined by cellular automata.
- **Stigmergy:** pheromone trails to minimize the time required for ants to detect similar ants in the world.
- **Data-Driven Positional Swapping:** swapping grid positions of neighboring ants based on their data similarity so that clusters are ordered internally.
- **Static Variable Determination:** no algorithmic variable is dynamically changed during runtime.
- **Comprehensible Visualization:** meaningful relative positioning of clusters to one another, and internal order within clusters themselves.

Section 4.4 further evaluates both algorithms in the context of performance and visual quality.

3. Approach

As shown in Figure 1, like the cells inside a cellular automaton, ants are positioned within a regular, finite grid of cells. Each ant is controlled by a discrete timeline and has a limited perception of the surrounding neighborhood, namely its eight neighboring cells. Every ant is determined by the same set of rules for updating from one discrete time step to the next. An ant’s behavior is solely based on the presence of other ants or pheromone trails in its local neighborhood and their according data values. At each iteration, the rules are simultaneously applied for all ants, causing them to move their position to one of the eight fields in their neighborhood or to stay put.

The cellular ant algorithm is derived from considering two, intrinsically contradicting concepts:

- **Edge Repulsion.** Ants have the freedom to roam around and meet enough other ants to ensure that *all* similar ants find each other and no clusters will be created that are separate but contain similar data items.

In addition, ants that are close to dissimilar ants should feel the tendency to move away to find better matches. This part of the algorithm is based on the ant colony optimization algorithm, in which ants follow the strongest pheromone scent of the most similar ant.

- **Surface Tension.** Ants that represent similar data items should stay close to each other and organize themselves ‘internally’. Clusters with dissimilar data items should have no common borders, unless they ‘share’ specific data values, so a meaningful visualization can be created. These tendencies are algorithmically derived through typical cellular automata rules in which the behavior of an ant is determined by the discrete amount of its immediate neighbors: ants with four or more good neighbors should stay put, those with a few good neighbors should move but attempt to stay in the direct neighborhood, and those with any dissimilar neighbor(s) should attempt to move away by following fresh pheromone trails originating from similar ants.

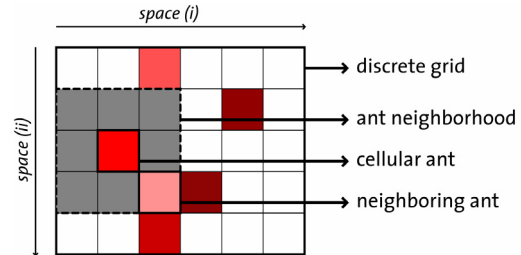


Figure 1. A cellular ant and its local neighborhood.

3.1. Algorithm

For each time step, the actions of each cellular ant is determined by the following inducements:

- **Discrete Data Tolerance.** Ants only consider (and thus ‘count’) other ants that are ‘similar’, that is when the distance between their data values in parameter space are below a discrete, predefined *similarity tolerance threshold* value t . Data similarity between pairs of ants is calculated as follows (with p as the dimensionality of the dataset):

$$data_i = (z_{i1}, z_{i2}, \dots, z_{ip}) \in R^p, p \in Z^+$$

$$d_{ij} = d(data_i - data_j) = \|data_i - data_j\|_p$$

$$d_{ij} < t \Rightarrow similar(ant_i, ant_j) = true$$

t seems to be similar to the object distance measure variable α in normal ant-based clustering approaches, but results in simple Boolean parameter (‘similar’, ‘not similar’), instead of a continuous similarity value.

- **Pheromone Trailing.** An ant will follow the trail of a) the most similar ant, that b) is the freshest, so that ants find similar ants rapidly. Each ant leaves a pheromone trail, consisting of following attributes: a)

data value(s) from that ant, b) an ant ID, and c) the time that has passed since the ant was occupying that cell. The data values allows an ant to follow the ‘most similar’ ant, the ID assures that an ant does not follow its own trail, while the time value enables evaporation.

- **Surface Tension.** Pheromone-following ants tend to generate small, separate clusters that have unstable cohesiveness. Therefore, cellular automata inspired algorithms determine ant actions depending on the discrete amount of similar neighboring ants. Here, an ant with less than 4 similar neighbors should move to a non-empty cell in its neighborhood that a) has no non-similar neighbors, and b) is next to the most similar ant. This rule will cause ants to form large, stable clusters.

- **Edge Repulsion.** Ants in a favorable setting, thus with 6 or more similar neighbors, should still attempt to move away when there are one or more non-similar ants in its neighborhood. This rule typically will cause large clusters to repulse each other at their outer edges, generating ‘empty’ cells around their perimeter.

- **Positional Swapping** orders ants internally within clusters in relation to data similarity, enabling ants to jump ‘over’ each other to reach more ideal positions within a cluster. In addition, swapping will cause ants that are trapped or positioned in ‘wrong’ clusters to be rapidly ‘pushed’ out to the outer cluster borders. This concept is made possible because the cellular ants method considers ants as cellular automata cells that are able to ‘sense’ data values of neighboring ants. Ants should organize their positions in the grid relatively to each other according to *relative data value gradients* (of neighbors in all grid directions) that are as *monotonic* as possible. At each iteration, each ant picks a random direction (horizontal, vertical, or one of both diagonals) in its neighborhood, with itself as the middle ant. It then reads the data values of the corresponding neighbors, and calculates the (one-dimensional) *data value distances* d_{ij} between all ants in the multi-dimensional parameter space. Based on these three pair-wise values, an ant is able to determine if it needs to swap its position with one of both its outer neighbors, or if the current constellation is ideal, even for multi-dimensional datasets. If the distance in parameter space between the middle agent and an outer ant is larger than between the outer ants themselves, the middle ant has to swap (Figure 2). Subsequently, the swapping rule will linearly order ants in the chosen grid direction by data similarity, so that ‘more similar’ ants are positioned closer to each other and dissimilar ones are put further apart in the grid. Although this rule organizes ants recursively in randomly chosen directions, an ordered structure will emerge due to the multitude of simultaneous local interactions.

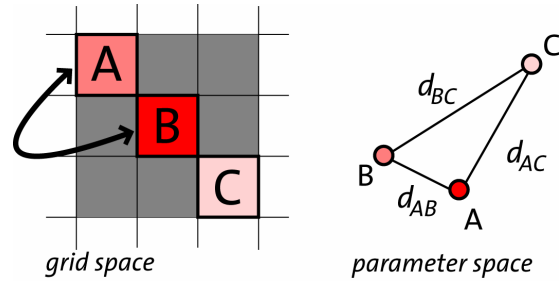


Figure 2. For a random direction of ant B (e.g. left-to-right diagonal), the largest data value distance in *parameter space* is d_{BC} , and ant A's data values lie between ant B and C. These dependencies are represented in *grid space* by swapping ants A and B, so that A lies between B and C in the grid.

The swapping rule ensures that for any three ants that are linearly neighboring each other, the pair of ants with the largest distance in parameter space will be positioned at the outer grid positions. This data similarity swapping rule still respects the concept of ant decentralization, as ant B only considers the data values of its immediate neighbors A and C. It is generally applicable for multi-dimensional datasets, as it applies to the one-dimensional distance measure d_{ij} in parameter space, calculated between pairs of ants. For a setting of ant_A , ant_B and ant_C as shown in Fig. 2, following swapping rule is valid:

$$d_{AC} > d_{AB}, d_{AC} > d_{BC} \Rightarrow ok$$

$$d_{AC} < d_{AB}, d_{BC} < d_{AB} \Rightarrow swap(ant_B, ant_C)$$

$$d_{AC} < d_{AB}, d_{BC} > d_{AB} \Rightarrow swap(ant_A, ant_B)$$

All previously mentioned rules are represented in pseudo code, as shown on the next page (right column).

3.2 Implementation

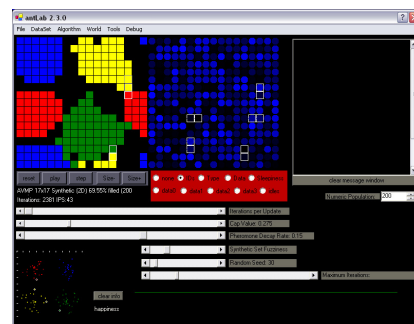


Figure 3. AntLab application screenshot.

Because of the sheer multitude of simultaneously applied local interactions within the collection of ants, it is often impossible to exactly predict the emergent outcomes of specific behavior rules. A process of trial

and error was required to determine, evaluate and fine-tune diverse combinations of ant-based and cellular automata rules in the context of stable clustering performance and meaningful data representation. A software application was developed, coined “AntLab” (Figure 3) enabling interactive experimentation with combinations of behavior rules and parameter values.

4. Evaluation

4.1. Influential Factors

Figure 4 shows the typical ‘cellular automata’ characteristics of the proposed method: all ants act as separate grid cells that cluster after a relatively long period of apparent instability and chaos. Our method not always finishes with a static result: a very small number of ants might be swapping indefinitely, or individual ants (e.g. data outliers) might continuously roam around. Therefore, we propose that some quantifiable measurement (e.g. clustering analysis or cost function) should inform the user how the data representation progresses towards clustering order.

As our method is based on cellular automata principles, the world grid size plays an important role in both the cluster quality and calculation performance. ‘Simple’ datasets with many ‘similar’ data objects have a large set of possible stable ant constellation solutions, so that too large grids will not assure that all similar ants meet, often resulting in small separate clusters containing similar data. Too small grids increase the importance of the swapping rule, so that large clusters form with separate sub-clusters. Grid size influence plays a complex role for multi-dimensional datasets, which have less possible stable ant constellation solutions: in order that all ants ‘meet’, larger grids will have more success and more iterations will be required for a stable constellation. Experiments show that grids should contain less than 15-20% ‘empty’ cells to ensure good clustering. Subsequently, the cellular ant method utilizes screen space more effectively in comparison to other ant-based algorithms.

4.2. Synthetic Dataset

Figure 4 shows the results for an artificial dataset of 500 data objects, 2 dimensions and 4 distinct classes. The dataset obeys a (x, y) normal distribution which is randomized for each separate run. Table 1 lists the according numerical results. The method is capable of clustering the four class types significantly well for 200 data objects. This can be explained by a relatively small search space, and thus a decrease in importance of the world size for smaller amounts of ants.

```

Cellular Ant Algorithm
initialize all agents at random locations
/* clustering */
for each agent a do
  for each neighborhood cell n do
    if (n != empty) then
      if (|n.data - a.data| < tolerance t) then
        store n in array N
        neighbors = neighbors + 1
      else store n in array E
  /* no neighbors */
  if (neighbors = 0) then
    for each cell c in E do
      if (c has pheromone p) then
        array diffs = |p.data - a.data|
        array strengths = p.strength
        best = c with lowest diff & highest strength
        move a to best
  /* less than four data similar neighbors */
  if (neighbors < 4) then
    for each cell c in E do
      l = c.clockwiseNeighbor
      r = c.counterclockwiseNeighbor
      if (l != empty && r != empty && |l.data - a.data| < tolerance && |r.data - a.data| < tolerance) then array diffs = |c.data - a.data|
      best = c with lowest diff
      move a to best
  /* six or more similar neighbors */
  if (neighbors > 5) then
    for each cell c in N do
      if (|N.data - a.data| > tolerance) then
        tooclose = true
      if (tooclose) then
        for each cell c in E do
          l = c.clockwiseNeighbor
          r = c.counterclockwiseNeighbor
          if (l != empty && r != empty && |l.data - a.data| < tolerance || |r.data - a.data| < tolerance) then
            array diffs = |c.data - a.data|
            best = c with lowest diff & highest strength
            move a to best
  /* swapping */
  dir = random direction
  l = cell on left side of a in direction dir
  r = cell on right side of a in direction dir
  if (l != empty && r != empty) then
    if (|l.data - a.data| < |r.data - a.data|) then
      swap l, a
    else swap r, a
  /* system actions */
update pheromone on previous occupied position
evaporate pheromones at all grid cells
visualize all ants

```

This experimental data also shows there is a significant correlation between the amount of iterations required and the dataset size (and thus the amount of ants), with an almost constant average cluster performance (effectively recognized amount of clusters).

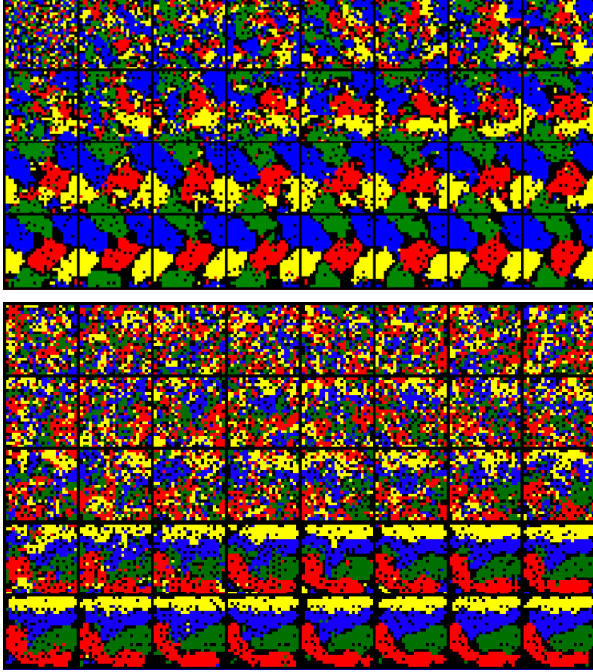


Figure 4. 2 timelines for different random seeds, with snapshots taken after each 100 iterations for a synthetic dataset (2D, 4 data types, 500 data objects, toroidal 26x26 grid, $t=0.23$).

The cluster quality performance drop (Table 1) for different grid sizes can be explained as larger grid sizes generally do not allow ants to travel around the ‘whole’ visualization space to detect ‘all’ similar ants, as they become assimilated in separate clusters too soon. This effect could be overcome with invoking time-separated clustering phases (e.g. first pheromone following, than swapping) or creating ants with a dynamically changing or larger neighborhood perception range. One should note that the calculation time required per iteration increases exponentially in relation to the total amount of ants (and thus data objects) in the world.

# ants	grid	# trials	avg. # clusters	avg. #iterations	std
200	16x16	100	4.02	994	385
500	24x24	100	4.05	3030	916
500	25x25	100	5.40	2515	790

Table 1. Performance comparison data for synthetic 2D datasets with 4 data types.

4.3. Data-Driven Relative Positioning

Figure 5 demonstrates that the relative positions of the data objects within *parameter space* are respected accordingly in *grid space*: ants that are positioned on borders that are shared between two (or more) clusters in the grid represent data values that are on nearby boundaries of their respective clusters in parameter space. For instance, the purple highlighted ants share a

border between the red and green data type clusters, which all corresponds to data objects of those clusters that have a small distance in parameter space. The same phenomenon can be perceived on all other areas on the grid where ants of different data types are positioned next to each other.

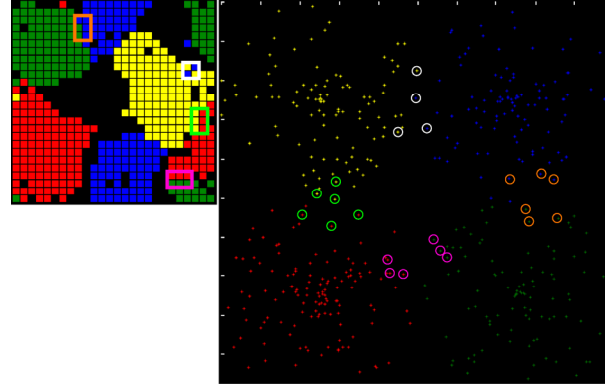


Figure 5. Relative positioning according to data similarity for a comprehensible visualization. (500 ants, toroidal 26x26 grid, 2D synthetic dataset, 4 classes, $t=0.27$). Left: ants in grid space; right: corresponding data objects in parameter space. Internal order: ants on shared borders of two clusters are also shared in parameter space. Global order: diagonal clusters in data space have no shared borders in grid space.

This data-driven, relative ant positioning enhances the comprehensibility of the resulting data clustering in the context of data visualization, a feature which is especially valuable for representing highly dimensional datasets on a two-dimensional display. The cellular ant method is capable of meaningfully clustering multi-dimensional datasets within a two-dimensional grid space, and is capable of organizing them spatially according to data similarity. Because of the continuous toroidal aspect of the grid, ants that are similar and nearby in parameter space not ‘necessarily’ are also grouped together in grid space, although always are positioned somewhere along shared cluster borders.

In Figures 4 and 5, one can also observe that the yellow and green (and equally, red and blue) clusters, seen as separate whole data value entities, have the largest data value distance both in data space (as they are positioned diagonally versus vertically or horizontally). More specifically, these specific pairs of clusters are positioned accordingly in grid space, as they do not directly border nor touch each other, so that the global visual representation is *true* to global dataset similarity characteristics, and clusters are positioned relatively in grid space in relation to their data similarity seen as a whole.

4.4. IRIS Dataset

The cellular ant method has been evaluated for the IRIS dataset (150 data items, 4 data dimensions, 3 data types), retrieved from the online Machine Learning Repository. As shown in Figure 6, the cellular ant method mostly groups the IRIS dataset in two separate clusters, with a spatial ordering of two subclusters within one larger cluster. This phenomenon is caused by close data similarities between data types in the second cluster, which is shown accordingly as subclusters for data visualization purposes.

# ants	grid	# trials	avg. # clusters	avg. #iterations	std
150	14x14	100	1	>5000	
150	15x15	100	1.95	1914	916
150	16x16	100	2.15	1565	576

Table 2. Performance data for four-dimensional IRIS dataset.

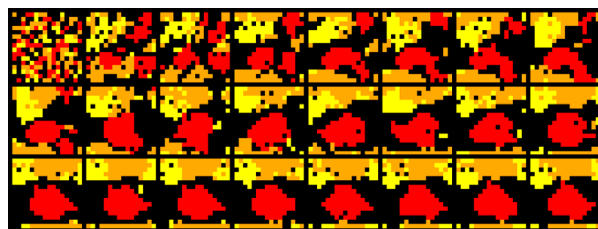


Figure 6. Timeline (snapshot each 100 iterations) for IRIS dataset (150 ants, 3 data types, 4 data dimensions, 16x16 toroidal grid, $t=0.51$).

Table 2 demonstrates that grid size plays a more complex role for multi-dimensional datasets: larger grid sizes have better clustering results as ants have more movement freedom in order to ‘find’ the much smaller number of similar ants, as the multi-dimensional data space is larger and similarities are rarer. Subsequently, such ants are also not as quickly assimilated in subclusters as in the synthetic dataset.

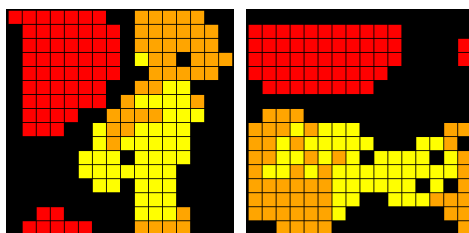


Figure 7. Two typical cellular ant clustering results for the IRIS dataset (150 ants, 4 data dimensions, 3 data types, toroidal 16x16 toroidal grid, $t=0.51$).

4.5. Performance

As with any parallel algorithm, the cellular ant method execution is relatively calculation intensive, and increases exponentially with the number elements (ants). The current prototype implementation is not yet optimized for performance issues, as all ants calculate

pair-wise data similarities in real time instead of taking similarity values from a static similarity matrix. Notably, these similarity calculations are performed for all ants and for each iteration, between an ant’s data values and those from a) any neighboring ants, b) any nearby pheromone trails and c) two neighbors for swapping testing purposes. Time measurements show that the cellular ant method requires approximately 20 seconds for a synthetic dataset with 4 data types and 200 ants. The cellular ant method could be significantly optimized by using data similarity lookup tables, not invoking swapping or pheromone following for each single iteration, or by recognizing ants in ideal positions, so that computational resources could be dedicated to poorly adapted ants.

The cellular ant method performs better, if measured by required iterations, than the standard ant clustering method [10] and the cellular automata inspired ASM approach [2]. The cellular ants method requires about 2500 iterations for a 2D synthetic dataset with 500 data items, which is significantly better than the standard ant-based method (1.000.000 iterations) and ASM (20.000 iterations [2]). This magnitude of order difference is not present for the IRIS dataset, as the results are quite similar (1600 iterations). Its current non-optimized performance is similar to the original ant-based clustering technique when considered for each ant separately (500 ants x 1.600 iterations = 800.000 ‘global’ iterations).

5. Discussion & Future Work

This paper presented a novel, decentralized data clustering algorithm that is based on the combination of cellular automata principles and ant colony optimization algorithms. Conceptually, the cellular ant method is based on three principles. Firstly, all data objects are mapped to separate ants, which are fully decentralized and are capable of actively roaming around by following the freshest pheromone trails of the most similar ants. Secondly, it uses insights from cellular automata rules by using static, discrete values (such as the amount of neighboring ‘similar’ ants and the simple Boolean, non-probability based data similarity tolerance threshold value t) to decide whether an ant should stay put or move around. Lastly, it employs positional swapping based on multi-dimensional data similarity to order clusters internally and relative to each other within the grid space.

Some of the limitations of the cellular ant method include its dependency on the grid size, which influences the quality of the emergent clusters, and the fact that some of the resulting clusters might ‘touch’

each other, although those borders normally consist of ants that are significantly similar in parameter space. The calculation performance (measured in time) correlates with the dataset size.

The cellular ant advantages include its algorithmic simplicity based on a few user-chosen variables and a small set of ant behavior rules that are fully determined by fixed, discrete values. Clusters are generated with relatively good calculation performance, better than traditional ant-based clustering techniques, even without using Denaubourg's data similarity algorithms or other probability-based functions. The cellular ant method's most significant contribution probably consists of the decentralized positional swapping rule, capable of substituting grid positions of ants within an ant's local neighborhood in any randomly chosen direction depending on their multi-dimensional data values. This swapping rule is determined by a (one-dimensional) monotonic increasing or decreasing tendency of (multi-dimensional) data value differences, so that the relative positions and pair-wise distances of ants in grid space resemble those in parameter space.

In the context of data visualization, the emergent data-driven, relative ant positioning generates more comprehensible visual representations than current clustering approaches. Clusters of multi-dimensional data are organized on a two-dimensional grid in a spatially meaningful way in relation to one another, so that: clusters containing common data items will share borders and clusters that are nearby each other contain according data objects in parameter space (and vice versa). Clusters are ordered internally based on multi-dimensional data similarity, so that common borders between clusters in grid space contain data objects that are similar to both clusters in parameter space. The cellular ant method utilizes available screen space more effectively than other ant-based clustering techniques by creating denser clusters that require fewer empty cells, a characteristic which is potentially valuable for the spatial grouping of very large datasets.

Future work will focus on the optimization of the algorithms to generate clusters more quickly and with better quality, for larger and more complex datasets. Several features will be implemented that will optimize the comprehensibility of the data representations by exploiting the decentralized communication and negotiation aspect of the cellular ants.

6. Acknowledgements

This research has been supported by a University of Sydney Research & Development Grant.

6. References

- [1] Bonabeau, E., Dorigo, M. and Theraulaz, G., *From Natural to Artificial Swarm Intelligence*, Oxford University Press, New York, 1999.
- [2] Chen, L., Xu, X., Chen, Y. and He, P., A Novel Ant Clustering Algorithm Based on Cellular Automata, *Intelligent Agent Technology (IAT 2004)*, 2004, pp.148-154.
- [3] Deneubourg, J., Goss, S., Franks, N., A., S.F., Detrain, C. and Chretien, L., The Dynamics of Collective Sorting: Robot-Like Ants and Ant-Like Robots, *From Animals to Animats: International Conference on Simulation of Adaptive Behaviour*, 1990, pp.356-363.
- [4] Goss, S., Beckers, R., Deneubourg, J.L., Aron, S. and Pasteels, J.M., How Trail-Laying and Trail Following can Solve Foraging Problems for Ant Colonies. in Hughes, R.N. ed. *Behavioural Mechanisms of Food Selections*, Springer Verlag, 1990.
- [5] Handl, J., Knowles, J. and Dorigo, M., Ant-based Clustering and Topographic Mapping, *Artificial Life*, MIT Press, 2005, to appear.
- [6] Handl, J. and Meyer, B., Improved Ant-based Clustering and Sorting in a Document Retrieval Interface, *International Conference on Parallel Problem Solving from Nature (PPSN VII)*, 2002, Springer, pp.913-923.
- [7] Hiraishi, H., Sawai, H. and Mizoguchi, F., Design of a Visualization Agent for WWW Information, *Lecture Notes in Computer Science*, 2112, pp.249-258.
- [8] Kuntz, A., Layzell, P. and Snyers, D., A Colony of Ant-Like Agents for Partitioning in VLSI Technology, *European Conf. on Artificial Life*, 1997, MIT, pp.417-424.
- [9] Labroche, N., Monmarché, N. and Venturini, G., A New Clustering Algorithm Based on the Chemical Recognition System of Ants, *European Conference on Artificial Intelligence*, 2003, pp.345-349.
- [10] Lumer, E.D. and Faieta, B., Diversity and Adaptation in Populations of Clustering Ants, *From Animals to Animats: Conference on Simulation of Adaptive Behaviour*, 1994, pp.501-508.
- [11] Pham, B. and Brown, R., Multi-Agent Approach for Visualisation of Fuzzy Systems, *Lecture Notes in Computer Science*, 2659, pp.995-1004.
- [12] Ramos, V. and Abraham, A., Evolving a Stigmergic Self-Organized Data-Mining, *Conf. on Intelligent Systems, Design and Applications (ISDA-04)*, 2004, pp.725-730.
- [13] Robinson, N. and Shapcott, M., Data Mining Information Visualisation - Beyond Charts and Graphs, *International Conference on Information Visualisation*, 2002, IEEE, pp.577-583.
- [14] Schockaert, S., De Cock, M., Cornelis, C. and Kerre, E.E., Fuzzy Ant Based Clustering, *Lecture Notes in Computer Science*, 3172, pp.342-349.
- [15] Schroeder, M. and Noy, P., Multi-Agent Visualisation based on Multivariate Data, *International Conference on Autonomous Agents*, 2001, ACM Press, pp.85-91.
- [16] Vande Moere, A., Time-Varying Data Visualization using Information Flocking Boids, *IEEE Symposium on Information Visualization*, 2004, IEEE, pp.97-104.
- [17] Von Neumann, J., *Theory of Self-Reproducing Automata*, University of Illinois Press, Illinois, 1966.