2017

Cellular Automata Ants

Bitsakidis, Nikolaos P.

Springer

Chapter 22
Cellular Automata Ants

Nikolaos P. Bitsakidis, Nikolaos I. Dourvas, Savvas A. Chatzichristofis and Georgios Ch. Sirakoulis

Abstract During the last decades much attention was given to bio-inspired techniques able to successfully handle really complex algorithmic problems. As such Ant Colony Optimization (ACO) algorithms have been introduced as a metaheuristic optimization technique arriving from the swarm intelligence methods family and applied to several computational and combinatorial optimization problems. However, long before ACO, Cellular Automata (CA) have been proposed as a powerful parallel computational tool where space and time are discrete and interactions are local. It has been proven that CA are ubiquitous: they are mathematical models of computation and computer models of natural systems and their research in interdisciplinary topics leads to new theoretical constructs, novel computational solutions and elegant powerful models. As a result, in this chapter we step forward presenting a combination of CA with ant colonies aiming at the introduction of an unconventional computational model, namely "Cellular Automata Ants". This rather theoretical approach is stressed in rather competitive field, namely clustering. It is well known that the spread of data for almost all areas of life has rapidly increased during the last decades. Nevertheless, the overall process of discovering true knowledge from data demands more powerful clustering techniques to ensure that some of those data are useful and some are not. In this chapter it is presented that Cellular Automata Ants can provide efficient, robust and low cost solutions to data clustering problems using quite small amount of computational resources.

N.P. Bitsakidis (✉) · N.I. Dourvas · G.Ch. Sirakoulis
Democritus University of Thrace, Xanthis 67100, Greece
e-mail: nbitsakidis@gmail.com

N.I. Dourvas
e-mail: ndourvas@ee.duth.gr

G.Ch. Sirakoulis
e-mail: gsirak@ee.duth.gr

S.A. Chatzichristofis
Information Technologies Institute Centre of Research and Technology,
Thessaloniki, Greece
e-mail: schatzic@iti.gr

A. Adamatzky (ed.), *Advances in Unconventional Computing*,
Emergence, Complexity and Computation 23,
DOI 10.1007/978-3-319-33921-4_22

22.1 Introduction

Cellular Automata (CA) are an idealization of a physical system in which space and time are discrete, and the physical quantities take only a finite set of values. CA originally proposed by John von Neumann, who by following a suggestion of Stanislaw Ulam [46] presented them as formal models of self reproducing organisms that can capture the essential features of systems where global behaviour arises from the collective effect of simple components which interact locally [34]. Non-trivial CA are obtained whenever the dependence on the values at each site is non-linear. As a result, any physical system satisfying differential equations may be approximated by a CA, by introducing finite differences and discrete variables [44]. CA forms theoretical background and, at the same time simulation tools and implementation substrates, of mathematical machines with unbounded memory, discrete theoretical structures, digital physics and modelling of spatially extended non-linear systems; massive-parallel computing, language acceptance, and computability; reversibility of computation, graph-theoretic analysis and logic; chaos and undecidability; evolution, learning and cryptography [3, 42, 47]. It is almost impossible to find a field of natural and technical sciences, where CA are not used.

On the other hand, the Ant Colony Optimization (ACO) algorithms are basically a colony of artificial ants or cooperative agents, designed to solve combinatorial optimization problems. These algorithms are probabilistic in nature because they avoid the local minima entrapment and provide very good solutions close to the natural solution [8]. ACO algorithms are extensively used to a variety of applications such as the travel salesman problem [18], image retrieval [28, 29], classification [31], electrical load pattern grouping [11], video games [38], seismic methods [13], communications networks [15], etc. Moreover, ACO algorithms were also used for solving the path planning in a team of robots and most of the effort was to implement the algorithm in real and virtual systems [4, 19, 20, 25–27, 40, 41].

In the last decade the amount of the stored data related to almost all areas of life has rapidly increased. However, the overall process of discovering knowledge from data demands more powerful clustering techniques to ensure that this knowledge is useful. To give a definition, data clustering is an assignment of a set of observations into subsets, namely clusters, so that observations in the same cluster are similar in some trait-often proximity according to a predefined distance. It is an unsupervised learning method, used mainly from statistics and database communities in fields like machine learning, information retrieval, image analysis and pattern recognition. There are several conventional algorithms trying to optimize the clustering techniques in the literature [2, 39]. But there are also some unconventional algorithms inspired from biology that are increasingly considered in clustering because of the low computational cost that they provide. There are efficient approaches in studies based on the ant colonies that try to address the clustering problems [18, 24] as well as modified methods like the Ant Colony Optimization with Different Favor (ACODF) algorithm [45], Constrained Ant Colony Optimization (CACO) algorithm [12] and the modified ACO algorithm proposed by Tiwari et al. [43].

Based on the fact that ants' social behavior, originates from a feature that is common to CA, namely self-organization, i.e. a set of dynamical mechanisms ensuring that the global aim of the system could be achieved through low level interactions between its elements, some new computational intelligence techniques, namely cellular ants have been proposed [10, 27, 47] for application in different scientific fields. In this chapter a cellular automata ants clustering model inspired by the cellular ants algorithm of Vande Moere and his colleagues [32, 33] is presented providing new features in order to overcome some of the previous model limitations. Original rules were modified and extended (long interaction rules) to successfully simulate more complex situations found in different size databases. Furthermore, some limitations like the excessive mobility of cellular ants as well as the discrete data tolerance proposed method limits and the practical flaws of the small size of the grid have been successfully addressed. More over, the idea of proportional distance measurement, the prediction for weights on data dimensions have been effectively applied and tested in different databases. On the other hand, the presented CA model is characterized by as much as low complexity as possible so that the computational recourses are kept low while its computation speed is kept high.

22.2 Cellular Automata and Ant Colony Optimization Principles

Cellular Automata (CA) were originally introduced by John Von Neumann [34] and his colleague Stanislaw Ulam [46]. CA can be considered as dynamical systems in which space and time are discrete and interactions are local. In general, a CA is consisted of a large number of identical entities with local connectivity arranged on a regular array. A finite Cellular Automaton could be defined by the quadruple:

$$\{d, q, N, F\} \tag{22.1}$$

From Eq. 22.1, variable d is a vector of two elements, m and n, denoting the vertical and horizontal CA dimensions, respectively. Both of these variables are expressed in number of cells. At each time step, the state of each cell is updated using a value from the set $Q = 1, 2, \ldots, q - 1$, called set of states. The neighborhood of each cell is defined by the variable N. For a 2-D CA, two neighborhoods are often considered, Von Neumann and Moore neighborhood . Von Neumann neighborhood is a diamond shaped neighborhood and can be used to define a set of cells surrounding a given cell $(x0, y0)$. Equation 22.2 defines the Von Neumann neighborhood of range r.

$$N^{v}_{(x0,y0)} = (x, y) : |x - x0| + |y - y0| \leq r \tag{22.2}$$

For a given cell $(x0, y0)$ and range r, Moore neighborhood can be defined by the following equation:

$$N^{M}_{(x0,y0)} = (x, y) : |x - x0| \leq r, |y - y0| \leq r \tag{22.3}$$

The transition rule f determines the way in which each cell of the CA is updated. The state of each cell is affected by the cell values in its neighborhood and its value on the previous time step, according to the transition rule or a set of rules. The state of every cell is updated simultaneously in the CA, thus, providing an inherent parallel system.

CA have sufficient expressive dynamics to represent phenomena of arbitrary complexity and at the same time can be simulated exactly by digital computers, because of their intrinsic discreteness, i.e. the topology of the simulated object is reproduced in the simulating device. The CA approach is consistent with the modern notion of unified spacetime. In computer science, space corresponds to memory and time to processing unit. In CA, memory (CA cell state) and processing unit (CA local rule) are inseparably related to a CA cell. Furthermore, CA are an alternative to partial differential equations [36, 44] and they can easily handle complicated boundary and initial conditions, inhomogeneities and anisotropies [21, 37].

The basic element of ACO algorithms is "ants" that is, agents with very simple capabilities which, to some extent, mimic the behavior of real ants [17]. Real ants are in some ways much unsophisticated insects. Their memory is very limited and they exhibit individual behavior that appears to have a large random component. However, acting as a collective, ants collaborate to achieve a variety of complicated tasks with great reliability and consistency [14], such as defining the shortest pathway, among a set of alternative paths, from their nests to a food source [5]. This type of social behavior is based on a common feature with CA, called self-organization, a set of dynamical mechanisms ensuring that the global aim of the system could be achieved through low level interactions between its elements [22]. The most vital feature of this interaction is that only local information is required. There are two ways of information transfer between ants: a direct communication (mandibular, antennation, chemical or visual contact, etc.) and an indirect communication, which is called stigmergy (as defined by Grassé [23]) and is biologically realized through pheromones, a special secretory chemical that is deposited, in many ant species, as trail by individual ants when they move [7]. More specifically, due to the fact that ants can detect pheromone, when choosing their way, they tend to choose paths marked by strong pheromone concentrations. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. This behavior is known as "auto catalytic" behaviour or the positive feedback mechanism in which reinforcement of the previously most followed route, is more desirable for future search. In ACO algorithms, an ant will move from point i to point j with probability:

$$\rho_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})} \tag{22.4}$$

where, $\tau_{i,j}^{\alpha}$ and $\eta_{i,j}^{\beta}$ are the pheromone value and the heuristic value associated with an available solution route, respectively. Furthermore, α and β are positive real parameters whose values determine the relative importance of pheromone versus heuristic information.

During their search for food, all ants deposit on the ground a small quantity of specific pheromone type. As soon as an ant discovers a food source, it evaluates the quantity and the quality of the food and carries some to the nest on their back. During the return trip, every ant with food leaves on the ground a different type of pheromone of specific quantity, according to the quality and quantity of the food. In ACO algorithms, pheromone is updated according to the equation:

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j} \tag{22.5}$$

where, $\tau_{i,j}$ is the amount of pheromone on a given position (i, j), ρ is the rate of the pheromone evaporation and $\Delta\tau_{i,j}$ is the amount of pheromone deposited, typically given by:

$$\Delta\tau_{i,j} = \begin{cases} 1/L_k & \text{if ant } k \text{ travels on edge } i, j \\ 0, & \text{otherwise} \end{cases} \tag{22.6}$$

where L_k is the cost of the kth tour of an ant (typically is measured as length). Finally, the created pheromone trails will guide other ants to the food source.

Consider for example the experimental setting shown in Fig. 22.1. The ants move along the path from food source F to the nest N. At point B, all ants walking to the nest must decide whether to continue their path from point C or from point H (Fig. 22.1a). A higher quantity of pheromone on the path through point C provides

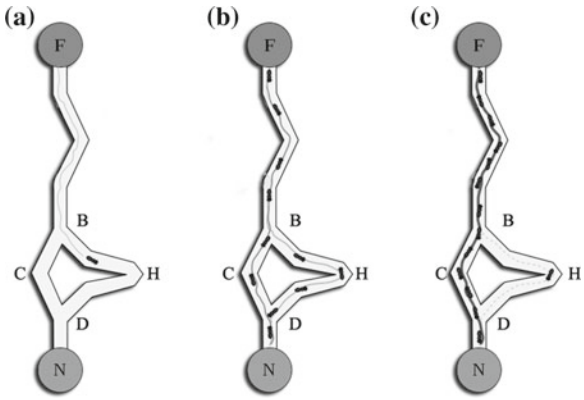

Fig. 22.1 An example of real ants colony: **a** An ant follows BHD path by chance. **b** Both paths are followed with same probability and **c** Larger number of ants follow the shorter path

an ant a stronger motivation and thus a higher probability to follow this path. As no pheromone was deposited previously at point B, the first ant reaching point B has the same probability to go through either point C or point H. The first ant following the path BCD will reach point D earlier than the first ant which followed path BHD, due to its shorter length. The result is that an ant returning from N to D will trace a stronger trail on path DCB, caused by the half of all the ants that by chance followed path DCBF and by the already arrived ones coming via BCD. Therefore, they will prefer path DCB to path DHB. Consequently, the number of ants following this path will be increased during time than the number of ants following BHD. This causes the quantity of pheromone on the shorter path to grow faster than the corresponding longer one. Consequently, the probability with which any single ant chooses the path to follow is quickly biased towards the shorter one.

22.3 Cellular Automata Ants for Clustering Problems

22.3.1 Initial Algorithm

In the initial algorithm [32], ants are positioned within a regular, finite grid of cells, similar to the cells inside a CA. Each ant is controlled by a discrete timeline and has a limited perception of the surrounding Moore neighbourhood and is also determined by the same set of rules in order to update its position 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 neighbourhood 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 neighbourhood or to stay put. As mentioned in [32], the cellular ant algorithm is derived from considering two, intrinsically contradicting concepts: *Edge Repulsion* and *Surface Tension*.

In more details, 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", namely 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):

$$\begin{aligned} &data_i = \left(z_{i1}, z_{i2}, \ldots, z_{ip}\right) \in R^p, p \in Z^+ \\ &d_{ij} = d\left(data_i - data_j\right) = \left\|data_i - data_j\right\|_p \\ &d_{ij} < t \Rightarrow similar\left(ant_i, ant_j\right) = true \end{aligned} \tag{22.7}$$

where t seems to be similar to the object distance measure variable a in normal ant-based clustering approaches, but results in a 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 the 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, CA algorithms determine ant actions depending on the discrete amount of similar neighbouring ants. An ant with less than 4 similar neighbours should move to a non-empty cell in its neighbourhood that (a) has no nonsimilar neighbours, 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 neighbours, should still attempt to move away when there are one or more non-similar ants in its neighbourhood. 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 CA cells that are able to "sense" data values of neighbouring ants. Ants should organize their positions in the grid relatively to each other according to relative data value gradients (of neighbours 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 neighbourhood, with itself as the middle ant. It then reads the data values of the corresponding neighbours, 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 neighbours, 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. 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. The swapping rule ensures that for any three ants that are linearly neighbouring 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 neighbours 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

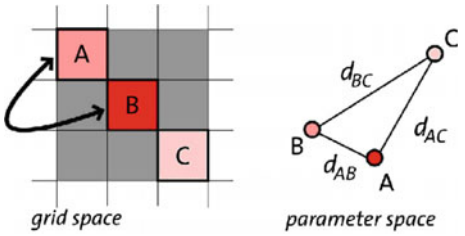

Fig. 22.2 For a random direction of ant B (e.g. *left*-to-*right diagonal*), the largest data value distance in parameter space is d_{BC}, and ant A's data values lie between ant B and C. These dependencies are represented in grid space by swapping ants A and B, so that A lies between B and C in the grid

of ants. For a setting of ants A, B and C, as shown in Fig. 22.2, following swapping rule is valid:

$$\begin{aligned} &d_{AC} > d_{AB}, d_{AC} > d_{BC} \Rightarrow ok \\ &d_{AC} < d_{AB}, d_{BC} < d_{AB} \Rightarrow swap\,(ant_B, ant_C) \\ &d_{AC} < d_{AB}, d_{BC} > d_{AB} \Rightarrow swap\,(ant_A, ant_B) \end{aligned} \tag{22.8}$$

The proposed algorithm succeeds, at least in theoretical level, to overcome the clustering problem based on its algorithmic simplicity and efficiency. However, when applied to Iris Data Set, used here for comparison reasons, the results although better, if measured by required iterations, than the standard ant clustering method [30] and the CA inspired ASM approach [10], are not so efficient as expected. This was more pronouncing in large data sets, while in small data sets the already proposed algorithm responded quite well. As mentioned above, the algorithm was originally implemented for the IRIS data set, (using the same parameters, i.e. 150 data items, 4 data dimensions, 3 data types), retrieved from the online Machine Learning Repository [1] and the same algorithmic properties, i.e. threshold t equals to 0.51 and grid sizes 15×15. Some simulation results for different number of iterations are depicted in Fig. 22.9. Taking into consideration the presented algorithm, we tried to enrich its features in order to overcome some of the previous model limitations and to present, if possible, better clustering results.

For example, one of the possible causes for the non-expected results, is the excessive mobility of cellular ants which jump from one side of the grid to its opposite very easily. This problem was tried to be addressed by abandoning the concept of wrapping and replacing the toroidal grid with a rectangular one, an idea that was tested on experimental level with satisfying results.

On the other hand, the presented algorithmic improvements are characterized by as low complexity as possible so that the computational recourses are kept low while its computation speed is kept high. In the next subsections some possible different solutions to probable limitations and/or insufficiencies are examined and presented in details.

22.3.2 The Discrete Data Tolerance Proposed Method Limits

The concept of discrete data tolerance, faces two problems at its implication. The first one, is the diverse (not equally measured) weight of differences between the values of two data dimensions, when the first dimension has a wide allocation of values and the latter a short one. In order to calculate the similarity between two agents that carry data objects, a relatively small difference of values on the first data dimension is valued very important, while, on the contrary, a major difference on the second dimension is valued insignificant. This is a common problem when the similarity of two data objects is tried to be calculated using the absolute value of their dimensions' difference (the Euclidean distance between the two data objects). The common way to solve it is the normalization of data before their further processing, but on this case, a supervised preprocessing of data is not desirable. The second problem is the difficulty, on experimental level, of reaching and finding the ideal and appropriate tolerance threshold. Each data set has its own (that can vary from a very small threshold like 0.5 to a very large one like 3,000), which can be found by continuous try-false experiments, but this process slows down significantly the whole assignment of clustering on large data sets. This delay can make the method nearly useless in practice. In order to solve these problems, a new measurement called proportional distance is proposed.

The idea of Proportional Distance measurement, is the examination of the proportion between the values of two data objects. Two data objects that are identical should have a proportional distance of 0 %, two non-identical should have a larger one. The similarity threshold should distinct now two agents as similar or non-similars, accordingly to their percentage proportional distance.

$$dist\,(AB) = 2\left|\frac{100 \times \sum_{i=j}^{i=1} \frac{b_i}{a_i + b_i}}{j} - 50\right| \tag{22.9}$$

where j is the number of data dimensions, b_i is the second agent's value on data dimension i and a_i is the first agent's value on data dimension i.

Another interesting idea could be the necessity of imposing different pre-defined weights for each data dimension in order to calculate the data distance between two data objects or the case of not taking on account one or more data dimensions in order to use the same data set for various kinds of clustering.

Taking into account different weights for each data dimension, is made possible by applying minor changes to the proposed proportional distance measurement rule, as follows:

$$dist\,(AB) = 2\left|\frac{100 \times \sum_{i=j}^{i=1} \frac{b_i}{a_i + b_i}}{\sum_{i=j}^{i=1} weight_i} - 50\right| \tag{22.10}$$

22.3.3 The Practical Flaws of the Small Size of the Grid

The small size of the grid (only 15–20 % of the cells should be empty, according to the initial method) helps cellular ants to find similar ants on their neighbourhood in order to form clusters, or pheromone trails of similar ants to follow. Trying an initial random allocation of ants inside a large grid leads to the fact that the clustering couldn't even start. But the proposed small size of the grid also causes problems. First of all, it doesn't help the cellular ants to move freely. The cellular ants have a few empty cells to move on, this can slow down the process of forming appropriate clusters. Furthermore, the emerging clusters can't really completely separated. The edge repulsion concept works fine in order to push dissimilar cellular ants away from a cluster to find their similar, but they can't really diverge because of the lack of empty space, which implies that the emerging clusters are stuck to each other. In addition, the smallness of empty space results to the fact that cellular ants are very often positioned on the sides of the grid. A direct Moore neighbourhood consideration is then impossible—one (if the cellular ant is on a side), two (if the cellular ants is on an edge) sides of Moore neighbourhood are outside the grid. Some rules can't be implied in such a small neighbourhood.

Our second approach was to leave the initial setting untouched and allow cellular ants to do an initial forming of clusters for a number of iterations, then expanding the size of grid by adding empty cells on its sides. This concept worked very well, the already emerged clusters had space to move on and expand. Additional expansions could also be tested. So we proposed a new concept, called Progressively Expansive Grid. This concept works fine if it isn't misused, because after one level it doesn't help the process clustering, it simply adds empty cells that restrict the valuable screen space.

22.3.4 The Long Interaction Rule

In order to overcome the problem of trapped cellular ants, which, with no trails to follow, cannot escape from a hostile neighborhood, the concept of Long Interaction Rule is proposed using Hyper-Cellular Automata Ants [6]. Hyper-Cellular Automata Ants have extended visibility and interaction capability, beyond their direct Moore neighbourhood, radius 1; their neighbourhood for interaction is by this way extended.

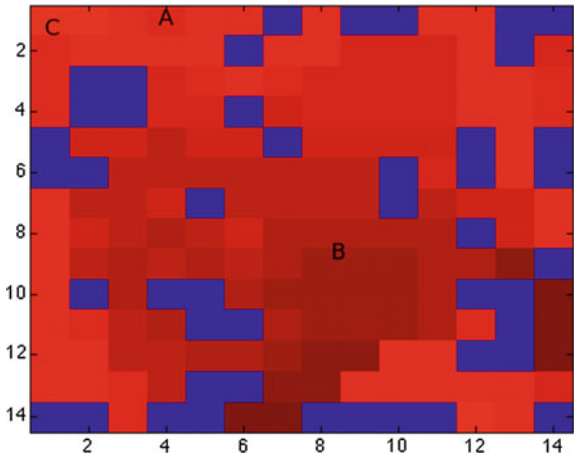

Fig. 22.3 *A* The lone cellular ant can't move freely to find its similar. *B* Neighborhooding clusters can't be separated. *C* The rules can't be applied correctly for cellular ants at the sides of the grid

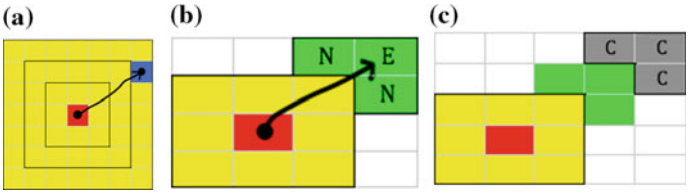

Fig. 22.4 **a** The hyper-cellular ant's neighborhood (on this image a Moore neighborhood, radius 3). Ant uses "fly-mode" in order to move to a cell far beyond its base. **b** The neighborhood of cellular ant is expanded on up-right direction. Two cells contain ants that are similar, the middle cell is empty—the cellular ant "flies" to the empty cell. **c** If the first extension doesn't provide results, the neighborhood is further extended. The chosen cells are again examined

Furthermore, Hyper-Cellular Automata Ants have the capability of using "fly mode" in order to move to a cell which is far from their base. There should be noted that on "fly mode", hyper-cellular automata ants don't drop pheromone trails (Figs. 22.3 and 22.4).

The Long Interaction Rule results from the restriction that the cellular ants neighbourhood is limited to the Moore neighbourhood, radius 1. The originators have considered this option, we tried to explore it. The two proposed variations of the Long Interaction Rule are the directional expanding long interaction rule and the extended neighbourhood long interaction rule.

More specifically, the directional expanding long interaction rule tries to extend the neighbourhood of an examined cellular ant to one strict direction, vertical, horizontal or diagonal, that faces the center of the grid. The direction to the center is universal, in order that all cellular ants that are trapped should try to form clusters at a single territory, and not being dispersed at all sides of the grid. Examining the three cells

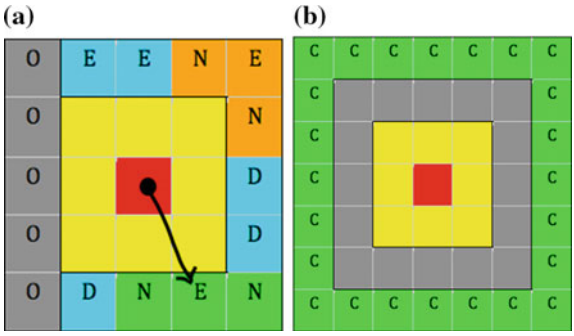

Fig. 22.5 The extended neighborhood long interaction rule. **a** The neighborhood of cellular ant is extended at all sides. **b** If the first extension doesn't provide results, the neighborhood of cellular ant is further extended at all sides

that compose the extended neighbourhood of the examined ant, if one cell is occupied by a similar ant and another is empty, the ant moves to the empty cell. Elsewhere we continue expanding the neighbourhood at this direction until the previous statement is met. The previous rule is overridden and the search is stopped without results, if the whole of extended neighbourhood is outside of the grid or a predefined extension limit is reached.

In the case of the extended neighbourhood long interaction rule, we are trying to extend all the sides of Moore neighbourhood of the examined cellular ant. The new neighbourhood is placed on the outside limits of this square. If there are any cells outside the grid on the new neighbourhood, these are not taken into account. Thereafter a consideration of every sequence of three—continuous on the neighbourhood—cells is examined, until a sequence of a "similar ant—empty cell—similar ant" is taking place. This sequence is checked. At the end of neighbourhood examination, the cellular ant moves to the empty cell of the sequence that has the minimum average data distance difference from it—considering both similar ants. If this statement can't be met (no sequence has been found), a further extension is taking place, until one or more desired sequences are found. The search for a desired movement to a better neighbourhood is stopped, if the whole neighbourhood is outside grid or if an extension limit is reached. As shown in Fig. 22.5a the neighborhood of cellular ant is extended at all sides. Some cells are outside the grid and as a result, they are not examined. Two series of neighborhooding ant-empty cell-neighborhooding cell are found. The cellular ants "flies" to the empty cell of the series that has the two neighborhooding ants with the smallest average data distance from examined cellular ant. In second thought (Fig. 22.5b) if the first extension doesn't provide results, the neighborhood of cellular ant is further extended at all sides. The chosen cells are again examined.

22.3.5 Progressively Expansive Grid with Inline Gaps

The inline gaps at the neighbourhoods of the grid, can serve two important goals. The first one, is giving cellular automata ants that are still alone or on groups of 2–3, far from their ideal neighborhood, the chance to move to it. The Long Interaction Rule causes the rapid forming of clustering neighborhoods but their closing too, so some cellular automata ants that should belong to them are excluded. In order to solve this problem, an expansion of the grid combined with insertion of random inline gaps inside the neighborhoods and an exclusive implication of Long Interaction Rule—and no other rule—for alone ants, or ones with no more than 3 similar ants on their neighborhood for this iterations only, is proposed. The exclusive implication of Long Interaction Rule preserves that the empty inline cells won't be captured by neighboring cellular automata ants, but from those ants that should be on those neighborhoods but are until then excluded. The second problem that inline gaps are solving, is the clustering of dynamic data sets. 'Dynamic' here has the meaning either of data objects that change their values during time or of new data objects that enter the grid in order to be clustered. Without inline gaps, the first ones could stay on their previous neighborhoods unable to escape to a neighborhood that pairs with their new values, and the latter could not find their way to their similar data objects that are already clustered as cellular automata ants on tight, unreachable neighborhoods.

As shown in Fig. 22.6a the IRIS data set is clustered after 150 iterations using long interaction rule with proportional distance measurement. Some data objects are lone, far beyond their appropriate clusters. Afterwards, the grid is extended at all sides Fig. 22.6b and the inline gaps are created for each line of the grid Fig. 22.6c. At the already changed grid, inline gaps are created for each column (Fig. 22.6d). The grid is now filled with inline gaps. Practicing long interacting rule only for lone ants, three of them now found their way to their appropriate cluster. Only one is left alone. The clustering is continued on the altered grid, this is the image of grid after further 50 iterations (Fig. 22.6e). The progressive expansion with inline gaps can be further practiced in order to finally cluster all the lone ants (Fig. 22.6f).

22.4 Application of Image Processing on Cellular Automata Ants Clustering

Cellular Automata Ants method had the intention to visualize the clustering of data objects, and it manages well on this purpose. But if a way to identify and separate the emerged clusters was found, it would signal a huge improvement in relation to algorithm's value, because the separated clusters could form new data sets that can be used for further clustering or for statistical analysis. We tried to face this problem, by approaching the visualization as an image. On the scientific field of Digital Image Processing, there are a plenty of ways to unify pixels with nearly common gray level, in order to form large territories and reduce the number of color levels. One common

way, described by [35] is by finding a threshold of gray level on a local neighborhood of image—usually the average of gray level on this neighborhood—and unifying the local pixels that have a local variability to the threshold value that is below one predefined level, by altering their previous value on gray level attribute with the threshold value. A second level of unification of similar pixels on gray level, can be done by using Region Growing Method [9], which unifies areas that have weak boundaries (meaning pixels in the boundaries of two areas that have similar values on gray-scale level), in order to form larger areas. This solution can't be really applied on Cellular Automata Ants Method, because finding thresholds (local or global) requires a prior knowledge of the data set, which is a supervised static process—we cannot for example "feed" the grid with dynamic continuous data sets, because they will alter these threshold levels. Instead of using pre-defined or calculated thresholds, we considered a different approach.

The identification algorithm selects random cells that are captured by cellular automata ants, and then extends the area of them, by adding cells that are connected to initial cell and are being captured by similar ants (See [16] for explanation of this process on Digital Image Processing). The continuation of area is preserved by using von Neumann neighborhoods in order to compare the examined cells to the initial. In specific, the whole process is depicted in the following Figs. 22.7. At the beginning, a random cell containing a cellular automata ant is found (Fig. 22.7a). The von Neumann neighborhood of chosen ant is examined. If similar ants are found, the area is extended with their cells as shown in Fig. 22.7b. Then, the von Neumann neighborhood of each area's cell is again examined. If there are ants that are similar to the initial ant, then the area is extended with the cells they occupy (Fig. 22.7c). As a result, the final area which consists of cells that contain the initial ant and ants

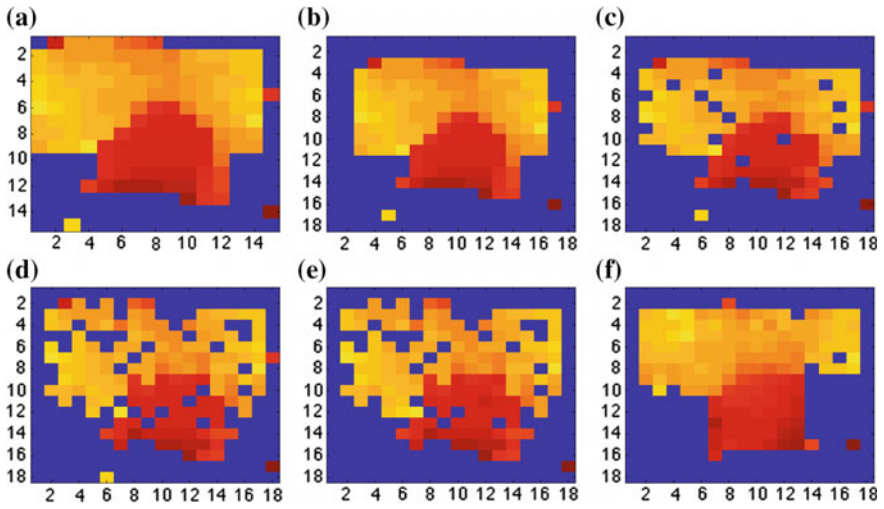

Fig. 22.6 The graphical representation of the progressively expansive grid method with inline gaps

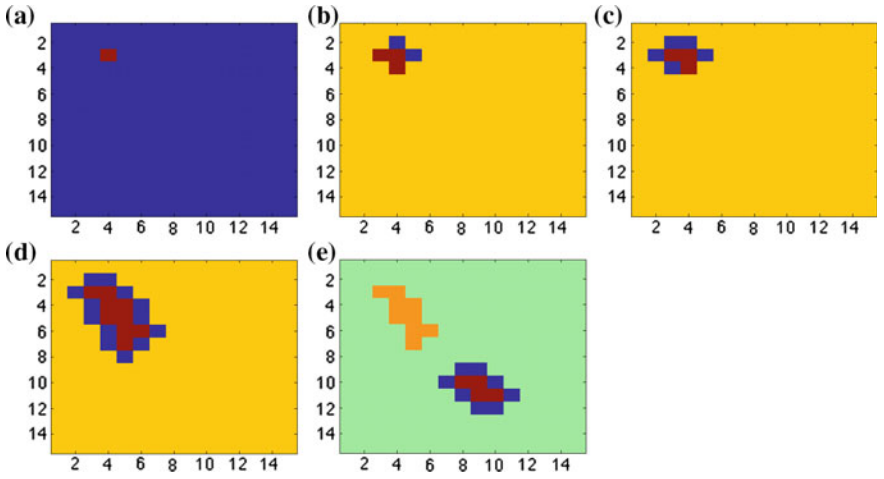

Fig. 22.7 The different steps of the identification algorithm

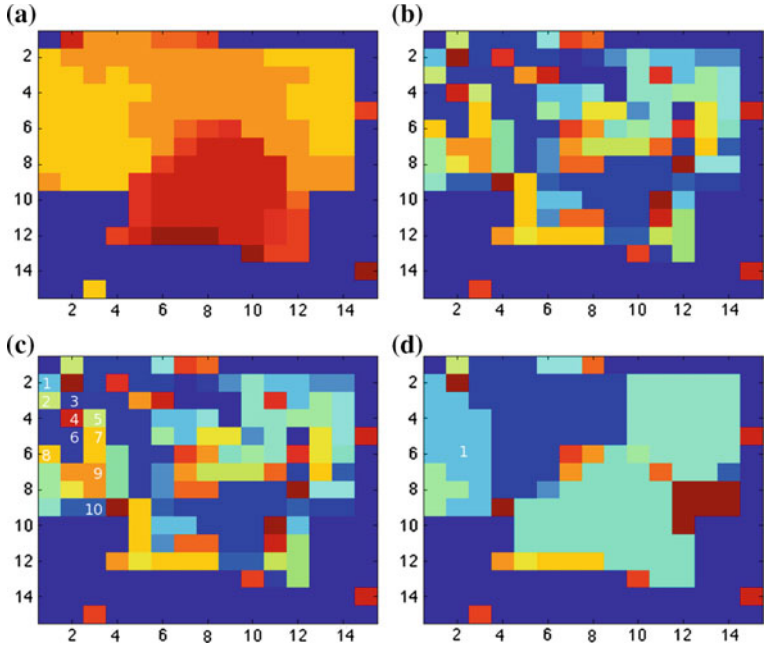

Fig. 22.8 The different steps of the identification algorithm plus

that are similar and presented in Fig. 22.7d. Finally, as given in Fig. 22.7e, using the same technique, another area is found and so on.

The unification of recognized areas is made possible, by merging the areas that have weak boundaries (See [9]). If the cellular automata ants from each side of

neighboring areas, are similar, the areas are unified. This process continues until no other areas can't be merged. The last step is recognizing the cellular automata ants that compose each final area—the data objects each area contains form the recognized clusters. In specific, taking into consideration the initial map (Fig. 22.8a) in order to find the clusters that are only visualized in original method, we use Identification Method. More specifically, each cluster consists of agents that are very near ($\pm 1\%$ similarity threshold) on data space (using Proportional Distance measurement) in relation to a random chosen agent as shown in (Fig. 22.8b). We use a small threshold, so the randomness of results is limited as possible. Figure 22.8c visualizes the next step of the method that is to unify the clusters that are very close on data space. We use a bigger similarity threshold (on this example $\pm 3.9\%$) comparing the common borders of neighboring clusters. If more that 90 % of common borders between two clusters are similar, then the clusters are unified. On this figure (Fig. 22.8c), we number the clusters which are going to be unified as one cluster. The final cluster takes the number of one (1) of the similar, but previously separated clusters, as depictured in Fig. 22.8d. Consequently, the result of unifying all similar clusters—each cluster's agents are recognized, and the clustering is presented in Fig. 22.8d. The whole concept of Identification, Unification and Separation concluded with a result that not only visualized the clusters—as Cellular Automata Ants original method tried to achieve—but also recognized, without supervision, the members of each cluster.

22.5 Simulation Results

For readability reasons, Iris database [1], perhaps the most known database to be found in the pattern recognition literature was selected as our initial test bed. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other. In Fig. 22.9a the initial pseudo-random allocation of cel-

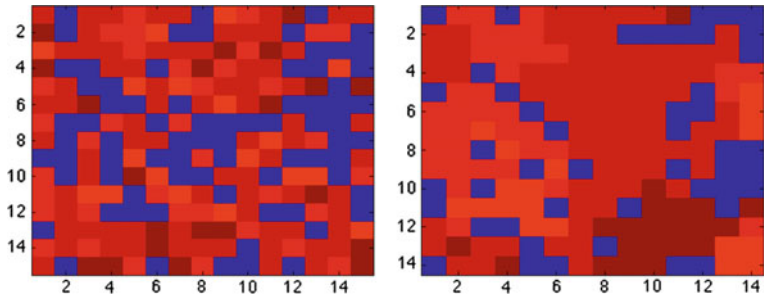

Fig. 22.9 Simulation results for (*left*) the initial grid, (*right*) after 1,500 iterations for the initial algorithm without wrapping

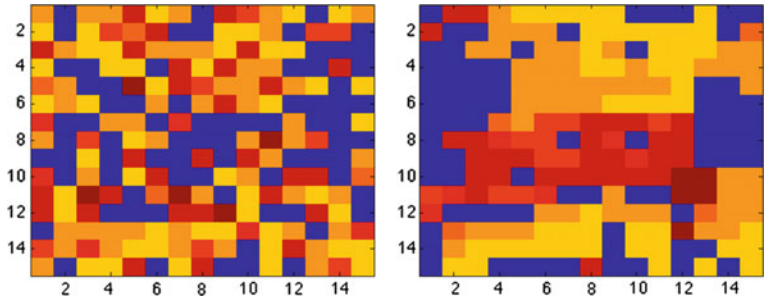

Fig. 22.10 Simulation results for (*left*) initial grid for proportional distance and (*right*) after 1,500 iterations for proportional distance measurement without wrapping

lular automata ants on grid, which has been used on all tests, is presented. The shades represent the difference of data values between each cellular automata ant-agent in relation to a hypothetical cellular automata ant that "carries" a data object with zero data values at each dimension. Two cellular automata ants with similar shade "carry" two similar data objects. The initial algorithm using the aforementioned threshold ($t = 0.51$) and the grid size (15×15) suggested by Vande Moore and Clayden, is not performing as well as expected. The clustering can't provide good results, the cellular automata ants are left on their starting neighbourhoods or switching sides with wrapping, unable to find their similar. The problem is answered by leaving the concept of wrapping, adopting a rectangular grid instead of toroid one. The results were much better but even now not completely satisfactory. In Fig. 22.9b, the grid, after using initial algorithm without wrapping for 1,500 iterations, is presented.

For these reasons we have proceeded with the proportional distance. The same initial pseudo-random allocation has been used on all tests with methods that use proportional distance measurement as depictured in Fig. 22.10a. The difference is that now, a sorting of cellular automata ants is made in prior, in order to determine with which shade each cellular automata ant will be represented at the grid. In Fig. 22.10b, the grid after 1,500 iterations, using proportional distance measurement and setting the threshold to 10 ($\pm 10\%$), is presented. It is clear that the results are good, but a little worse in relation to authentic method after abandoning the concept of wrapping.

According to the presented algorithm improvements, the idea of long interactions has also tested in order to improve the aforementioned results. More specifically, in Fig. 22.11a the grid after 1,500 iterations, using long interaction rule is presented. This time the threshold was set to $t = 0.71$, and the long interaction range was set to the half of the size of the grid. The results were much better, on average the 95.44 % of the first class of IRIS Data Set is clustered together. However, still some lone ants, are not clustered where they should belong. As a result, the combination of long interaction in accordance with proportional distance was applied. Figure 22.11b presents the classification results of the initial grid after only 150 iterations, using long interaction rule and proportional distance measurement. The threshold was set to $t = 10$ ($\pm 10\%$), and the long interaction range was set to the half of the size of

the grid. The results were satisfactory, on average the 74.48 % of the first class of IRIS Data Set is clustered together. The most important issue was that algorithm was coming to an end at 150 iterations—only minor movements of cellular automata ants as a result of swapping rule are performed from then on. This corresponds to an improvement of 1066.66 % in relation to all competing algorithms, combined with satisfactory results. The remaining issue is that the 74.48 % of successful clustering is having a devation of 20.3 %, meaning that the least successful clustering attempts could have only about 55 % success.

The idea of inline gaps was also applied. More specifically, in Fig. 22.12a it is shown the grid after 1,500 iterations, using long interaction rule and progressively expansive grid with inline gaps. The results were almost perfect, the clustering is very good and only a small amount of cellular automata ants are positioned outside of their appropriate clusters. Using long interaction rule, proportional distance measurement and progressively expansive grid with inline gaps, the results were almost as good (even better) than when no proportional distance measurement is used. In Fig. 22.12b these results were depictured for the initial grid after 1,500 iterations.

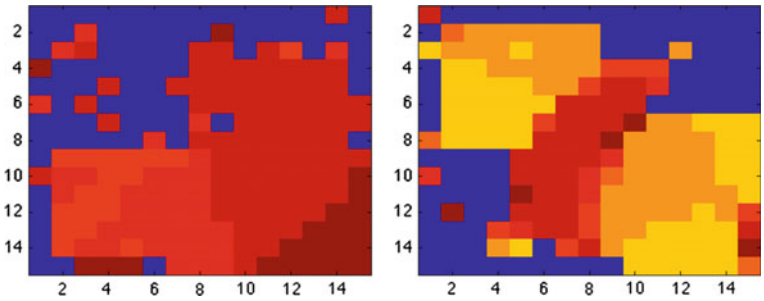

Fig. 22.11 Simulation results (*left*) after applying the long interaction rule at the initial grid for 1,500 iterations and (*right*) after applying the long interaction rule combined with proportional distance for only 150 iterations

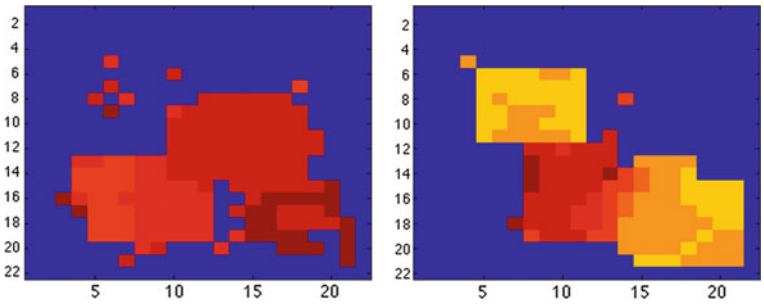

Fig. 22.12 Simulation results (*left*) ater applying long interaction rule and progressively expansive grid with inline gaps at the initial grid for 1,500 iterations and (*right*) after applying the long interaction rule, proportional distance measurement and progressively expansive grid with inline gaps for 1,500 iterations

The last trial was to use a progressively expansive grid with inline gaps. Using both Euclidean measurement and Proportional Distance measurement, the results were completely satisfactory. With classic data distance measurement, setting the step for each expansion of the grid on 600 iterations and ending clustering on 1,500 iterations, the results were almost perfect. The first class of IRIS data set is recognized on average at 96.4 %, but also the deviation from that number is minimized to 2.45 %, meaning that the least successful clustering attempts have a success of 93.5 %. With Proportional Distance measurement, there was a further improvement. Setting the step to each expansion to 150 iterations, and ending the clustering on 400, on average the 88.48 % of the first class of IRIS data set is correctly recognized as a cluster, but also the deviation from that numbered is minimized to 15.46 %, meaning that the least successful clustering attempts have a success of 72 %. Some comparison results can be found in Table 22.1.

Table 22.1 Simulation results for the IRIS dataset when applied different improvements

Method	Time Av.	Success Av.
Initial	126.7303625	60.28
Prop. Dist.	167.4780554	49.92
Long Inter.	220.2520988	95.44
Long Inter. and Prop. Dist.	98.4326502	74.48

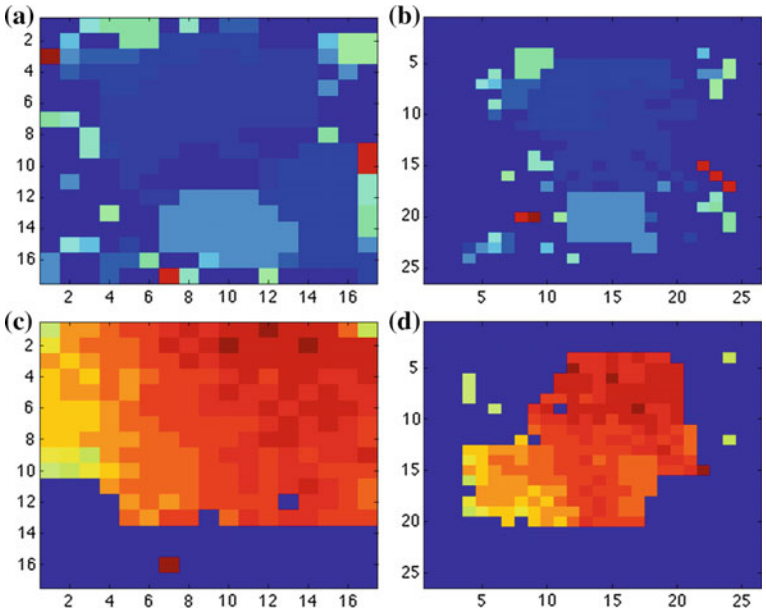

Fig. 22.13 The application of hybrid cellular automata ants to computer hardware dataset

Furthermore, in order to test the functionality of the proposed hybrid cellular automata ants algorithm another dataset was considered, namely the Computer Hardware Dataset, created by Phillip Ein-Dor and Jacob Feldmesser. This database concerns multivariate data set characteristics with 209 number of instances and 9 number of attributes. As it can be found out in Fig. 22.13a, when the computer hardware data set was clustered with long interaction rule the results were satisfactory. In addition, when the computer hardware data set was clustered with long interaction rule with progressively expansive grid and inline gaps the results were even better, as presented in Fig. 22.13b. Moreover, when long interaction rule and proportional distance measurement were applied the results were further improved (Fig. 22.13c) and with all the above, as well as progressively expansive grid with inline gaps, the results were almost perfect. In conclusion, we tried the algorithm using a non-standard data set. The results were satisfactory judging from the view, but we couldn't appreciate quantitatively the success of the clustering because of the lack of class information about this data set.

Finally, the ideas of identification and separation when applied to the hybrid cellular automata ants, were tested exhaustively on IRIS dataset. More specifically, in

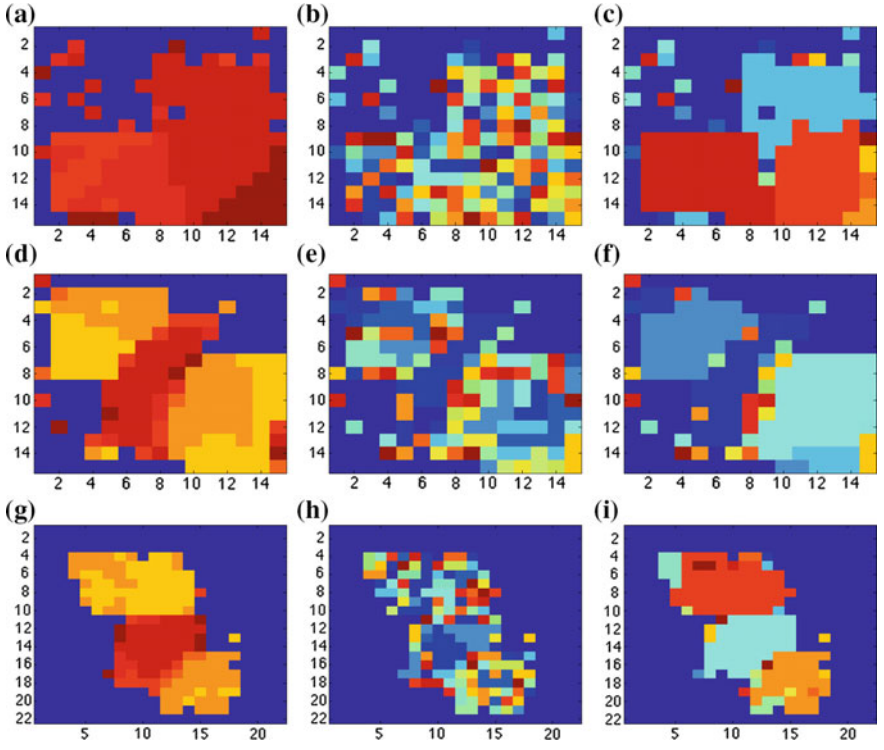

Fig. 22.14 Graphical results of the identification and separation methods when applied to hybrid cellular automata ants in combination with all the previous methods

Fig. 22.14a the initial clustering using long interaction rule is presented. The first class of IRIS data set is on left bottom, waiting to be recognized. During the procedures of identification and separation the identification of small clusters is succeeded (Fig. 22.14b). When the unification process is finished as shown in Fig. 22.14c, the presented results were very good, especially at the recognition of the first class of IRIS data set (which is shown here on red). Some more tests were performed with the application of long interaction rule and proportional distance measurement. As it can be shown in corresponding Figs. 22.14d–f, the results were quite good. Finally, the results were slightly better when expansive grid with inline gaps is used combined with the use of long interaction rule and proportional distance measurement (Figs. 22.14g–i).

From a quantitative point of view, we tried the identification and separation process on three different improvements on algorithm. On long interaction rule, setting the identification threshold to 0.1 and unification threshold to the same as the threshold for clustering (0.71), the results were almost perfect. The first class of IRIS data set, is recognized and separated at 91.92 %, with a deviation of 3.1 %, meaning that even the least successful attempts on identification and separation had 88.8 % success. The large number of clusters (24.88 % on average) remain a problem although. Using long interaction rule and proportional distance measurement, setting the identification threshold to 1 %, and unification threshold to 3.9 %, the results weren't so good. On average the 52 % of first class of IRIS data set is recognized as a cluster. The results were slightly better when expansive grid with inline gaps is used combined with the use of long interaction rule and proportional distance measurement. Setting the identification threshold to 1 %, and the unification threshold to 4.5 %, on average the 61.12 % of the first class of IRIS data set is recognized as a cluster. The explanation on latter—not quite good—results is that we haven't tried to find the ideal threshold for clustering when we used proportional distance measurement, we picked up one ($\pm 10\%$) that simply worked well. As the success of attempt to identificate and separate using long interaction rule prior for clustering indicates, the ideal threshold for clustering should be very near to the one used for identification and separation.

22.6 Conclusions

In this chapter, a cellular automata ants clustering model inspired by the cellular ants algorithm of [32, 33] that provides new features in order to overcome some of the previous model limitations was presented. The original rules were modified and extended (long interaction rules) by introducing the Hyper-Cellular Ants in order to successfully simulate more complex situations found in different size databases. Furthermore, some limitations like the excessive mobility of cellular ants as well as the discrete data tolerance proposed method limits and the practical flaws of the small size of the grid have been successfully addressed. More over, the idea of proportional distance measurement, the prediction for weights on data dimensions

have been effectively applied and tested in different databases. On the other hand, the presented CA model is characterized by as much as low complexity as possible so that the computational resources are kept low while its computation speed is kept high. As future work concerns, the expansion of the cellular ants model for handling more complicated clustering situations should be considered. More specifically, at this point, some ideas like the progressively expansive grid with inline gaps in order to help the long interaction rules to boost the model's performance should be further investigated although some preliminary results are rather encouraging. Furthermore, the application of image processing methods on cellular automata ants clustering for cluster identification and separation could be rather beneficial. Finally, the application of the proposed model to different databases with different properties should be also taken under consideration and the preliminary results, for example computer hardware database are also quite promising.

References

1. Asuncion, A.D.N.: UCI machine learning repository (2007). http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Abonyi, J., Feil, B.: Cluster Analysis for Data Mining and System Identification. Birkhäuser, Basel (2007)
3. Adamatzky, A.: Reaction-Diffusion Automata: Phenomenology, Localisations Computation. Springer Publishing Company Incorporated, Berlin (2014)
4. Akst, J.: Send in the bots. Scientist **27**(10) (2013). Cited By (since 1996)
5. Beckers, R., Deneubourg, J.L., Goss, S.: Trails and u-turns in the selection of a path by the ant lasius niger. J. Theor. Biol. pp. 397–415 (1992)
6. Bitsakidis, N.P., Chatzichristofis, S.A., Sirakoulis, G.C.: Hybrid cellular ants for clustering problems. Int. J. Ubiquitous Comput. **11**(2), 103–130 (2015)
7. Blum, C.: Ant colony optimization: introduction and recent trends. Phys. Rev. **2**(4), 353–373 (2005)
8. Bonabeau, E., Dorigo, M., Theraulaz, G.: Inspiration for optimization from social insect behaviour. Nature **406**, 39–42 (2000)
9. Brice, C.R., Fennema, C.L.: Scene analysis using regions. Artif. Intell. **1**(3–4), 205–226 (1970)
10. Chen, L., Xu, X., Chen, Y., He, P.: A novel ant clustering algorithm based on cellular automata. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology. IAT '04, pp. 148–154. IEEE Computer Society, Washington, DC, USA (2004)
11. Chicco, G., Ionel, O.M., Porumb, R.: Electrical load pattern grouping based on centroid model with ant colony clustering. IEEE Trans. Power Syst. **28**(2), 1706–1715 (2013)
12. Chu, S.C., Roddick, J.F., Su, C.J., Pan, J.S.: Constrained ant colony optimization for data clustering. In: C. Zhang, H.W. Guesgen, W.K. Yeap (eds.) In: Proceedings of the PRICAI 2004: Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, 9–13 August 2004. Lecture Notes in Computer Science, vol. 3157, pp. 534–543. Springer (2004)
13. Conti, C., Roisenberg, M., Neto, G., Porsani, M.: Fast seismic inversion methods using ant colony optimization algorithm. IEEE Geosci. Remote Sens. Lett. **10**(5), 1119–1123 (2013)
14. Deneubourg, J., Goss, S.: Collective patterns and decision-making. Ethol. Ecol. Evol. **1**(4), 295–311 (1989)
15. Di Caro, G., Dorigo, M.: Antnet: distributed stigmergetic control for communications networks. J. Artif. Int. Res. **9**(1), 317–365 (1998)

16. Dillencourt, M.B., Samet, H., Tamminen, M.: A general approach to connected-component labeling for arbitrary image representations. J. ACM **39**(2), 253–280 (1992)
17. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. thesis, Politecnico di Milano, Italy (1992)
18. Dorigo, M., Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. **1**(1), 53–66 (1997)
19. Garnier, S., Tache, F., Combe, M., Grimal, A., Theraulaz, G.: Alice in pheromone land: an experimental setup for the study of ant-like robots. In: Proceedings of the Swarm Intelligence Symposium, SIS 2007. IEEE, pp. 37–44 (2007)
20. Garnier, S., Combe, M., Jost, C., Theraulaz, G.: Do ants need to estimate the geometrical properties of trail bifurcations to find an efficient route? A swarm robotics test bed. PLoS Comput Biol **9**(3), e1002,903+ (2013)
21. Georgoudas, I., Sirakoulis, G., Scordilis, E., Andreadis, I.: A cellular automaton simulation tool for modelling seismicity in the region of Xanthi. Environ. Modell. Softw. **22**(10), 1455–1464 (2007)
22. Goss, S., Beckers, R., Deneubourg, J., Aron, S., Pasteels, J.: How trail laying and trail following can solve foraging problems for ant colonies. In: Hughes, R. (ed.) Behavioural Mechanisms of Food Selection. NATO ASI Series, vol. 20, pp. 661–678. Springer, Berlin (1990)
23. Grassé, P.P.: La reconstruction du nid et les coordinations interindividuelles chezBellicositermes natalensis etCubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. Insectes Sociaux **6**(1), 41–80 (1959)
24. Handl, J., Knowles, J., Dorigo, M.: Ant-based clustering and topographic mapping. Artif. Life **12**, 35–61 (2006)
25. Herianto., Kurabayashi, D.: Realization of an artificial pheromone system in random data carriers using rfid tags for autonomous navigation. In: Proceedings of the International Conference on Robotics and Automation, ICRA '09, pp. 2288–2293. IEEE (2009)
26. Herianto., Sakakibara, T., Kurabayashi, D.: Artificial pheromone system using RFID for navigation of autonomous robots. J. Bionic Eng. **4**(4), 245–253 (2007)
27. Ioannidis, K., Sirakoulis, G.C., Andreadis, I.: Cellular ants: a method to create collision free trajectories for a cooperative robot team. Robot. Auton. Syst. **59**(2), 113–127 (2011)
28. Konstantinidis, K., Sirakoulis, G., Andreadis, I.: Design and implementation of a fuzzy-modified ant colony hardware structure for image retrieval. IEEE Trans. Syst., Man, Cybernetics, Part C: Appl. Rev. **39**(5), 520–533 (2009)
29. Konstantinidis, K., Andreadis, I., Sirakoulis, G.C.: Chapter 3 - application of artificial intelligence methods to content-based image retrieval. In: P.W. Hawkes (ed.) Advances in Imaging and Electron Physics. Advances in Imaging and Electron Physics, vol. 169, pp. 99–145. Elsevier (2011)
30. Lumer, E., Faieta, B.: Diversity and adaptation in populations of clustering ants, from animals to animats. In: Conference on Simulation of Adaptative Behaviour, pp. 501–508 (1994)
31. Martens, D., De Backer, M., Haesen, R., Vanthienen, J., Snoeck, M., Baesens, B.: Classification with ant colony optimization. IEEE Trans. Evol. Comput. **11**(5), 651–665 (2007)
32. Moere, A.V., Clayden, J.J.: Cellular ants: combining ant-based clustering with cellular automata. In: Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '05, pp. 177–184. IEEE Computer Society, Washington, DC, USA (2005). http://dl.acm.org/citation.cfm?id=1105924.1106055
33. Moere, A.V., Clayden, J.J., Dong, A.: Data clustering and visualization using cellular automata ants. In: Australian Conference on Artificial Intelligence, pp. 826–836 (2006)
34. Neumann, J.V.: Theory of Self-Reproducing Automata. University of Illinois Press, Champaign (1966)
35. Niblack, W.: An Introduction to Digital Image Processing. Strandberg Publishing Company, Birkeroed (1985)
36. Omohundro, S.: Modelling cellular automata with partial differential equations. Physica D: Nonlinear Phenom. **10**(1–2), 128–134 (1984)

37. Progias, P., Sirakoulis, G.C.: An FPGA processor for modelling wildfire spreading. Math. Comput. Modell. **57**, pp. 1436–1452 (2013)
38. Recio, G., Martin, E., Estebanez, C., Saez, Y.: Antbot: ant colonies for video games. IEEE Trans. Comput. Intell. AI Games **4**(4), 295–308 (2012)
39. Runkler, T.A.: Ant colony optimization of clustering models. Int. J. Intell. Syst. **20**(12), 1233–1251 (2005)
40. Russell, R.A.: Heat trails as short-lived navigational markers for mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 4, pp. 3534–3539 (1997)
41. Sirakoulis, G.C., Adamatzky, A.: Robots and Lattice Automata. Springer International Publishing, Cham (2015)
42. Sirakoulis, G.C., Bandini, S. (eds.): Cellular Automata. In: 10th International Conference on Cellular Automata for Research and Industry, ACRI 2012, Santorini Island, Greece, 24–27 September 2012. Lecture Notes in Computer Science, vol. 7495. Springer (2012)
43. Tiwari, R., Husain, M., Gupta, S., Srivastava, A.: Improving ant colony optimization algorithm for data clustering. In: Proceedings of the International Conference and Workshop on Emerging Trends in Technology, ICWET '10, pp. 529–534. ACM, New York, NY, USA (2010)
44. Toffoli, T.: Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. Phys. D: Nonlinear Phenom. **10**(1–2), 117–127 (1984)
45. Tsai, C.F., Wu, H.C., Tsai, C.W.: A new data clustering approach for data mining in large databases. In: Proceedings of the ISPAN, pp. 315–320 (2002)
46. Ulam, S.: Random processes and transformations. Int. Congr. Math. **2**, 264–275 (1952)
47. Was, J., Sirakoulis, G.C., Bandini, S. (eds.): Cellular Automata. In: Proceedings of the 11th International Conference on Cellular Automata for Research and Industry, ACRI 2014, Krakow, Poland, 22–25 September 2014. Lecture Notes in Computer Science, vol. 8751. Springer (2014)