



HAL
open science

Amoebae for clustering: a bio-inspired cellular automata method for data classification

Amaury Saint-Jore, Nazim A. Fatès, Emmanuel Jeandel

► To cite this version:

Amaury Saint-Jore, Nazim A. Fatès, Emmanuel Jeandel. Amoebae for clustering: a bio-inspired cellular automata method for data classification. *Automata and Complexity*, 42, Springer, Cham, pp.417-432, 2022, 978-3-030-92553-6. 10.1007/978-3-030-92551-2_23 . hal-02973830

HAL Id: hal-02973830

<https://inria.hal.science/hal-02973830v1>

Submitted on 21 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Amoebae for clustering: a bio-inspired cellular automata method for data classification

Amaury Saint-Jore, Nazim Fatès and Emmanuel Jeandel
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

October 21, 2020

Abstract

We present a bio-inspired mechanism for data clustering. Our method uses amoebae which evolve according to cellular automata rules: they contain the data to be processed and emit reaction-diffusion waves at random times. The waves transmit the information across the lattice and causes other amoebae to react, by being attracted or repulsed. The local reactions produce small homogeneous groups which progressively merge and realise the clustering at a larger scale. Despite the simplicity of the local rules, interesting complex behaviour occur, which make the model robust to various changes of its settings. We evaluate this prototype with a simple task: the separation of two groups of integer values distributed according to Gaussian laws.

Introduction

Biology is an inexhaustible source of inspiration for computer science. Indeed, many phenomena we observe in living organisms can be understood in terms of simple interacting processes which create a self-organising process with a rich behaviour, e.g. self-reproduction [NB66], cellular sorting [VBD10], etc. The model we present here aims at performing a clustering task. Various nature-inspired models have already been explored to perform this task [HM07, NP14]. The particularity of our model is that it inspired from the behaviour of the social amoeba *Dictyostelium discoideum* and that it is a cellular automaton model, that is, space, time and states of components are discrete and the interactions are purely local.

Our proposition relies on a previous cellular automata model of this social amoeba [Fat10]. This original model was used to gather thousands of particles initially scattered randomly on a lattice without any central coordination. From this original model, a variant was proposed to locate a hidden source on a lattice with the process called *infotaxis*, which is specific to the case where the detection of the source needs to be achieved with rare events [Fat16].

Our goal now is to continue this work of exploration of the amoebae models and examine here whether they could be used to perform data clustering: imagine that we have set of objects initially random scattered on a lattice and we need to group this data into classes which share some similarity. The difficulty of the task lies in the propagation of information which needs to be achieved by simple cells with as few states as possible and which obey local rules, more specifically they should interact only with their nearest neighbours.

We present our model with formal definitions in Sec. 1. This presentation is then followed by an empirical observation of its behaviour (Sec. 2.1), the description of a quantification tool (Sec. 2.2) and a first sequence of statistical experiments (Sec. 2.3). We then briefly explore some properties of robustness of the model (Sec. 3) and say a few words on the work that remains to be tackled.

1 Presentation of the model

Our model is described as a stochastic dynamical system where space, time and states are discrete. It is composed of two layers: the first layer corresponds to the environment, which propagates reaction-diffusion waves. The second layer contains the amoebae, which move and interact with the environment. Our objective is to design a system which will be as simple as possible in terms of cell states and behaviour, and for which the amoebae form groups according to the data they contain.

Space is modelled as a two-dimensional lattice with periodic boundary condition (torus), represented by $\mathcal{L} = (\mathbb{Z}/X \cdot \mathbb{Z}) \times (\mathbb{Z}/Y \cdot \mathbb{Z})$, where X and Y are the dimensions of the lattice. In this text, we only consider square lattices, that is, we take $X = Y = L$, where L is called the *size* of the lattice.

The set of states is that each cell can hold is itself composed of two sets: the first part is a ternary state and the second part corresponds to the data. The set of states is thus a Cartesian product $Q \times \mathcal{D}$. The first set is defined as $Q = \{\mathbf{N}, \mathbf{E}, \mathbf{R}\}$, where the states \mathbf{N} , \mathbf{E} , \mathbf{R} , respectively correspond to the *excited*, *neutral* and *refractory* states. The second set \mathcal{D} is dependent on the data set which is considered. We will assume that the set \mathcal{D} contains a particular value, called the *void* value, denoted by V , which represents the absence of data in a given cell.

Formally, for a cell $c \in \mathcal{L}$ and time $t \in \mathbb{N}$, we denote by $\sigma_c^t \in Q$ the state of this cell and by $\rho_c^t \in \mathcal{D}$ its data. For the sake of brevity, we will say *state* only to refer to the first part of the cell and say *data* for the second part. The reader should note that the data of a cell typically do not correspond to the data of its amoeba. It corresponds to the data propagated by the environment.

The set \mathcal{A}_c^t represents the amoebae contained in a cell c . Formally, \mathcal{A}_c^t is represented by a *multi-set* of elements of \mathcal{D} , that is, it may contain several elements of \mathcal{D} with an identical value. Recall that each amoeba is in charge of carrying an element of the dataset that has to be processed.

The interactions between cells follow a local rule and we use here the Moore neighbourhood, that is, a cell c will only “see” its own state and the state of its 8 nearest neighbours. This set is denoted by $\mathcal{N}(c)$.

In all the experiments that follow, the initial condition of the system is that all cells are initially in the neutral state \mathbf{N} and the elements of the dataset are randomly and uniformly distributed on the grid. For the data, we simply assign to each cell a given probability d_{ini} to contain a data, which is drawn at random according to a non-uniform distribution that will be described later. The probability d_{ini} represents intuitively the initial *density* of data, hence the use of the letter d .

Informally, our model relies on three main features: (a) the amoebae have the ability to randomly trigger reaction-diffusion waves that will make other amoebae react; (b) these waves propagate without attenuation and by partially colliding and merging when they meet ; (c) when amoebae sees such a wave in its environment, it reacts by being attracted or repulsed according to the data it sees and its own data. Let us now describe these steps more precisely.

1.1 The reaction-diffusion waves

The rules that govern the propagation of reaction-diffusion waves are simple: (a) A neutral cell becomes excited if it has an excited neighbour; (b) an excited cell always becomes refractory at the next time step; (c) a refractory cell always become neutral at the next time step. Once a neutral cell gets excited, the excitations propagate in form of waves formed by lines of excited cells that we call *fronts*, which are followed lines of refractory cells (Fig. 1). The role of the refractory state is to prevent the retropropagation of excitations. When two waves meet, they annihilate if the fronts

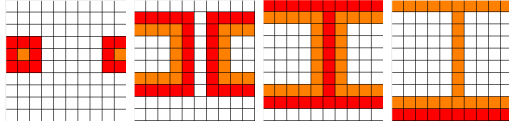


Figure 1: Propagation of two excitation waves: red, orange and white squares respectively correspond to the excited, refractory and neutral cells. (colour online)

propagate in opposite directions and merge if the fronts propagate in the same direction.

In addition to the possibility to being excited by its neighbours, a cell which contains at least one amoeba becomes excited with a probability λ , called the *excitation rate*. In this case, the excited cell takes the state E, and its data ρ_c^t will be chosen randomly and uniformly among the data of amoebae it contains. This choice is represented by $R[\mathcal{A}_c^t]$ where the operation of random uniform choice in a multiset X is denoted by $R[X]$. We note $B(p)$ as a Bernoulli law of probability p .

We denote by $E_c^t = \{c' \in N(c) : \sigma_{c'}^t = \mathbf{E}\}$ the set of cells which are in the excited state in the neighbourhood of c and by $D_c^t = \{\rho_{c'}^t : c' \in E_c^t\}$ the set of data that these cells contain. It should be noted that in general the values of the data contained in a given wave are uniform, as they come from the same amoeba source. However, when two waves of excitation collide and merge, a given cell might see in its neighbour two excited cells with different data, in which case it will choose randomly and uniformly in the multiset of values it sees.

The status of a cell is updated as follows:

$$\begin{aligned}
 &\text{If } \sigma_c^t = \mathbf{N} \\
 &\quad \text{if } \mathcal{A}_c^t \neq \emptyset \text{ and } B(\lambda) = 1 \text{ then } \sigma_c^{t+1} = \mathbf{E} \text{ and } \rho_c^{t+1} = R[\mathcal{A}_c^t] \\
 &\quad \text{else if } \text{card}(E_c^t) > 0 \quad \text{then } \sigma_c^{t+1} = \mathbf{E} \text{ and } \rho_c^{t+1} = R[D_c^t] \\
 &\quad \text{else if } \sigma_c^t = \mathbf{E} \quad \text{then } \sigma_c^{t+1} = \mathbf{R} \text{ and } \rho_c^{t+1} = V \\
 &\quad \text{else if } \sigma_c^t = \mathbf{R} \quad \text{then } \sigma_c^{t+1} = \mathbf{N} \text{ and } \rho_c^{t+1} = V \\
 &\quad \text{else } \sigma_c^{t+1} = \mathbf{N} \text{ and } \rho_c^{t+1} = V.
 \end{aligned}$$

We find our different cases: the self-excitation of a cell by an amoeba, the excitation by a neighbouring cell, the de-excitation and the return to the neutral state.

1.2 Movements of the amoebae

Let us describe the movements of the amoebae, which take place in the second layer of the model. To end, we introduce some intermediary definitions. A cell is *empty* if it does not contain any amoeba and it is *available* if it is either empty or it contains only one amoeba. In order to limit the number of amoebae (that is data) contained per cell we use the following rules: (a) Amoebae can move only to the available cells of their neighbourhood. (b) If a cell contains several amoebae, only one may move to a neighbour. (c) There is no limit on how many amoebae an available cell can simultaneously receive.

Let us now describe how the amoebae react to the presence of excited cells in their neighbourhood. The key point of the model is that an excitation can be either attractive or repulsive for an amoeba, depending on the value of the excitation and the data of the amoeba. The use of a repulsive interaction is a novelty compared to the previous models using amoebae

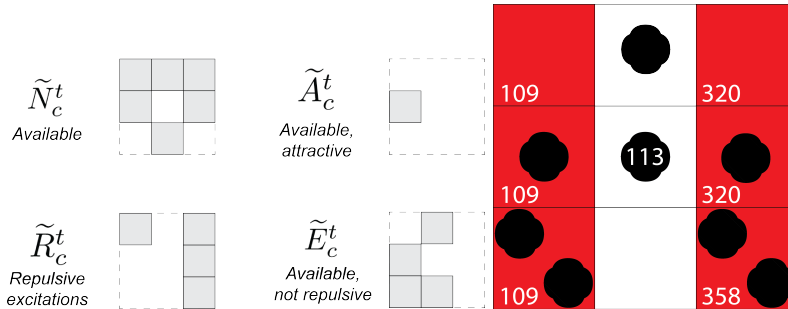


Figure 2: Representation of the four sets used to define the movements of an amoeba. Amoebae are represented in black and excited cells in red (colour online). The data contained in the excitations are represented in white.

(see Introduction) and its purpose is to separate the groups of amoebae that will correspond to different data clusters.

The idea is that for each excited cell in the neighbourhood, we will determine the excitation is attractive cell or repulsive according to a probability law, denoted by p_{att} , depending on the absolute value of the difference δ between the value of the amoeba and the value of the nearby excitation.

The probability p_{att} to be attractive is defined with the following equation: $p_{\text{att}}(\delta) = e^{-\delta/\Lambda}$, that is, the higher δ is, the lower the probability to be attractive and thus the higher the probability to be repulsed. The constant K is set by the user of the model and depends on the dataset that is used; it roughly corresponds to the expected distance between the clusters of the dataset.

Naturally, a problem arises when a cell contains several amoebae. In this case, we simply choose uniformly at random which of the amoebae is selected to move and we calculate δ according the data of the selected amoeba.

For a given cell $c \in L$ which contains one amoeba or more, we define for each cell c' in the neighbourhood of c the quantity $\delta_{c,c'} = |\rho_{c'}^t - R[\mathcal{A}_c^t]|$. (Note that the random value $R[\mathcal{A}_c^t]$ is chosen once and not for each neighbour.) We map $\delta_{c,c'}$ to a value $\delta_{c,c'}^* \in \{\mathbf{A}, \mathbf{R}\}$, where \mathbf{A} and \mathbf{R} respectively denote an attractive or repulsive signal.

To express the local rules which define the movements of the amoebae, we need to define three sets. For a given cell c the set \tilde{N}_c^t is the set of available neighbours: $\tilde{N}_c^t = \{c' \in \mathcal{N}(c) : \text{card}(\mathcal{A}_{c'}^t) \in \{0, 1\}\}$. The sets \tilde{A}_c^t (resp. \tilde{R}_c^t) is the set of available neighbours that are attractive (resp. repulsive): $\tilde{A}_c^t = \{c' \in \tilde{N}_c^t : \delta_{c,c'}^* = \mathbf{A}\}$ and $\tilde{R}_c^t = \{c' \in \tilde{N}_c^t : \delta_{c,c'}^* = \mathbf{R}\}$. The set \tilde{E}_c^t is the complement of the previous set; it is the set of cells where an amoeba can *escape* in case of repulsion: $\tilde{E}_c^t = \tilde{N}_c^t \setminus \tilde{R}_c^t$. All these sets are represented in fig. 2.

The movements will determined by a list of rules with an imposed order of priority: an amoeba may make a random movement to an available cell according to the probability p_{ag} , the phenomenon of attraction and finally that of repulsion. Second, the priority is given to attraction over repulsion, that is, if an amoeba sees at least one excited cell \mathbf{A} , it chooses attraction, otherwise it will choose a repulsive movement.

Denoting by Δ_c^t the cell where the selected amoeba moves, this yields:

$$\Delta_c^t = \begin{cases} R[\tilde{N}_c^t] & \text{if } B(P_a) = 1 \text{ and } \text{card}(\tilde{N}_c^t) > 0 \\ R[\tilde{A}_c^t] & \text{if } \sigma_c^t = \mathbf{N} \text{ and } \text{card}(\tilde{A}_c^t) > 0 \\ R[\tilde{E}_c^t] & \text{if } \sigma_c^t = \mathbf{N} \text{ and } \text{card}(\tilde{R}_c^t) > 0 \text{ and } \text{card}(\tilde{E}_c^t) > 0 \\ c & \text{otherwise.} \end{cases}$$

These rules represent the four possible movements: by agitation, by

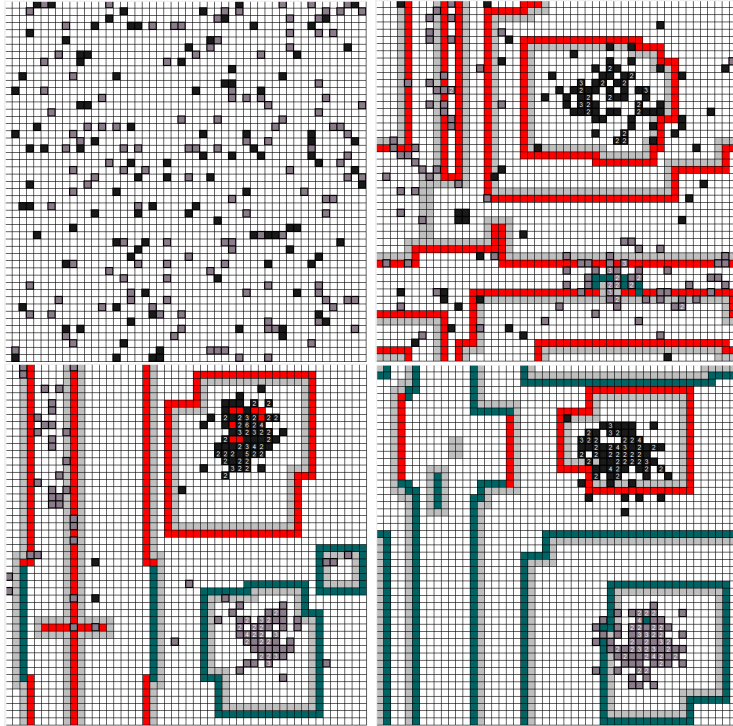


Figure 3: Evolution of the clustering process for times: 0, 450, 880 and 1270. The amoebae are represented in black and dark grey: this colour differentiation is only visual and indicates whether the value they contain are greater or lower than the average value 200. When several amoebae are present in the same cell, the number of amoebae is indicated by a white number (this number can be magnified in the online version). A neutral cell is white and a refractory cell in light grey. Excited cells are coloured in red and blue depending on whether the data they contain is greater or lower than the average value 200.

attraction, by repulsion and finally staying in the same cell. The positions of the amoebae are then updated synchronously for each non-empty cell c with the movement from c to Δ_c^t .

2 Study of the model

2.1 A first observation of the behaviour

Let us now observe some simulations. Figure 3 shows the clustering process on a lattice of size $L = 50$ cells. The initial condition is set with $d_{\text{ini}} = 0.1$, that is, each cell has a probability of 10% to contain a data. The value initially attributed to the amoebae is an integer drawn randomly in the interval $I = [0, 400]$. The data is not uniformly distributed: its distribution is the sum of two Gaussian laws of parameters (μ_1, σ_1) and (μ_2, σ_2) where the mean values are set to $\mu_1 = 100$ and $\mu_2 = 300$ and the standard deviations are set to $\sigma_1 = \sigma_2 = 0.2$.

The parameters settings were determined empirically after a series of tests. We set the value of the emission rate of waves to $\lambda = 0.01$ and the attraction probability law p_{att} is set with $\Lambda = 40$. In order to introduce some “difficulty” in the clustering process, we also add some agitation, setting $p_{\text{ag}} = 0.1$, that is, the amoebae will make a random move every ten steps in average.

One can observe the following general behaviour: the amoebae start forming small groups where the type is rather homogeneous. Thanks to

the randomness of the excitation waves, these groups progressively merge until they form larger groups.

At this stage two possibilities are generally observed: either the large groups directly form two well-sorted clusters, or the sorting stagnates with approximately two or three formed groups and a few isolated amoebae. In this case, the sorting process is somehow delayed due to two main reasons: (a) isolated amoebae may find themselves blocked between the different excitations, and (b) two groups of amoebae that contain amoebae of the same type might try to get closer but get repelled by another large group formed of amoebae of the opposite type. However, as one can observe on Fig. 3, the clusters are well formed after a few thousand steps and remain stable over time.

2.2 Quantify sorting progress and efficiency

We now describe how to evaluate more quantitatively the evolution of the clustering process. We continue to consider the simple case where the system should separate the data into two clusters only. Our problem is to evaluate, for a given distribution of amoebae on the grid, how far we are from our goal. Our clustering estimator will associate to each data a type, A or B, depending on its value. For example if the data is distributed in the range of integers from 0 to 100, *type-A* data will cover the range 0 to 49 and *type-B* data will cover the rest of the interval.

With this particular setting of the problem, two main features can be isolated: (a) *homogeneity*: once the groups of connected amoebae are identified, we estimate to which extent they are mixed or not. (b) *compactness*: we estimate how far the global configuration of amoebae is from forming two main groups.

For a given configuration, we thus partition the set of amoebae into connected components of amoebae, where Moore's neighbourhood is used to define the connexity property. Let us denote by C_1, \dots, C_k the components obtained, with the convention that C_1 and C_2 are respectively the largest and second largest sets (in terms of number of amoebae they contain). Given two components i and j , we denote by $d_H(i, j)$ the Hausdorff distance between these two components. This distance is frequently used to evaluate distances between sets of points and displays various good analytical properties [Fuj13]. We then make a partition of the set components $\{C_1, \dots, C_k\}$ into two sets of components P_1 and P_2 by assigning each component C_i to P_1 if the distance $d_H(C_i, C_1)$ is smaller than $d_H(C_i, C_2)$ and to P_2 otherwise.

To define the *homogeneity* criterion, we simply calculate the number of amoebae n_1^A and n_1^B (resp. n_2^A and n_2^B) that are of respective type A and B that are contained in P_1 (resp. in P_2). The *homogeneity* of P_m for $m \in \{1, 2\}$ is then set equal to $h_m = |n_m^A - n_m^B| / (n_m^A + n_m^B)$. The global homogeneity h is then simply the minimum of h_1 and h_2 . This form is chosen such that a homogeneous set of components will lead to $h = 1$ while a set of components with an equal number of amoebae of each type will give us $h = 0$.

Let us now examine the *compactness criterion*. We first calculate the average distance between the components and the main component they belong to; that is,

$$\tilde{d} = \frac{1}{k} \cdot \left(\sum_{C_i \in P_1} d_H(C_i, C_1) + \sum_{C_i \in P_2} d_H(C_i, C_2) \right)$$

We compare this value to the distance between the two main components of the system $d_H(C_1, C_2)$. The compactness κ is then defined as $\kappa = 1 - \tilde{d}/d_H(C_1, C_2)$; a form that is chosen in order to approach 1 as \tilde{d} gets smaller (while $d_H(C_1, C_2)$ remains stable).

The combination of these two criteria will give us an estimation of how the clustering process evolves in time. Note that criteria are not defined

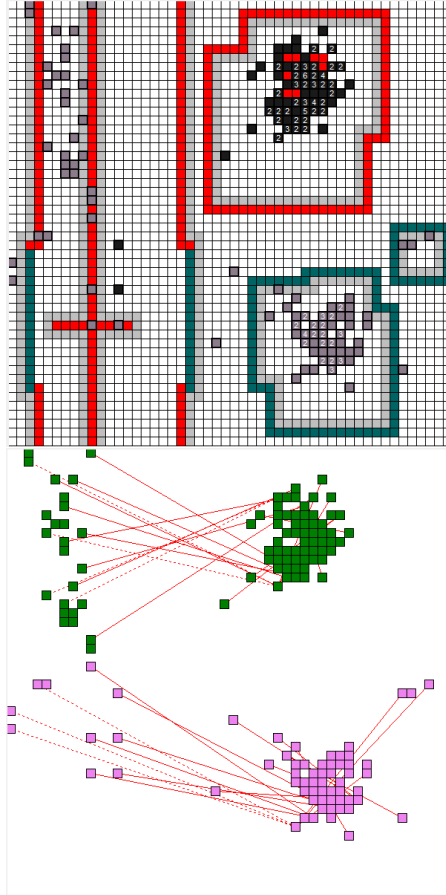


Figure 4: Representation of a particular configuration obtained at time 880. (left) The same conventions are applied, in particular amoebae are differentiated in black and dark grey depending on whether their value is higher or lower than the average value 200. (right) The amoebae are coloured in green or magenta according to the partition realised to calculate the homogeneity and compactness criteria (see Sec. 2.2). The red segments materialise Hausdorff distances; the dotted lines indicate that the shortest distance are calculated with periodic boundary conditions.

when we have only one component and note that we considered here the case of two types, but this method can of course be extended to any number of types. It should also be emphasised that the types are only given for the analysis of the process, but that the process itself never uses this information in the local rules it applies. Let us now observe the clustering process on some simple examples.

2.3 Quantitative analysis of the behaviour

Let us now analyse the relevance of the two criteria introduced above to quantify the progress of the sorting of amoebae in the system. The two connected components of amoebae are illustrated Fig. 4 and the evolution of the temporal values of these criteria are represented on Fig. 5. The parameters are set identically to those exposed in Sec. 2.1.

Let us first put our attention on the *homogeneity* criterion. One can observe that during the first moments of the simulation, the amoebae are scattered in space and the two main sets are very heterogeneous, which leads to a low homogeneity. This continues until amoebae form larger groups of amoebae with similar data, which yields a gradual increase in the homogeneity criterion, with an additional noise due to the randomness inherent to the model. As mentioned earlier, after this stage is complete,

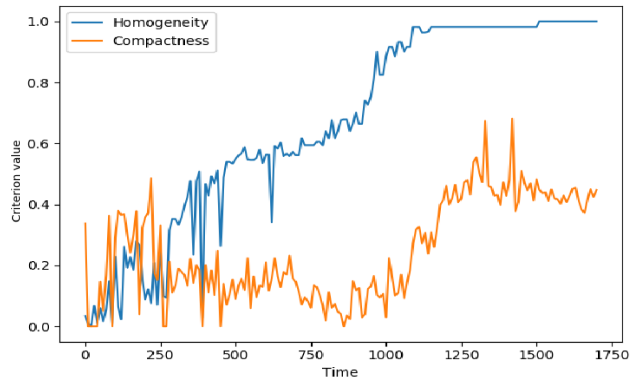


Figure 5: Temporal evolution of the criteria of homogeneity and compactness of a particular simulation.

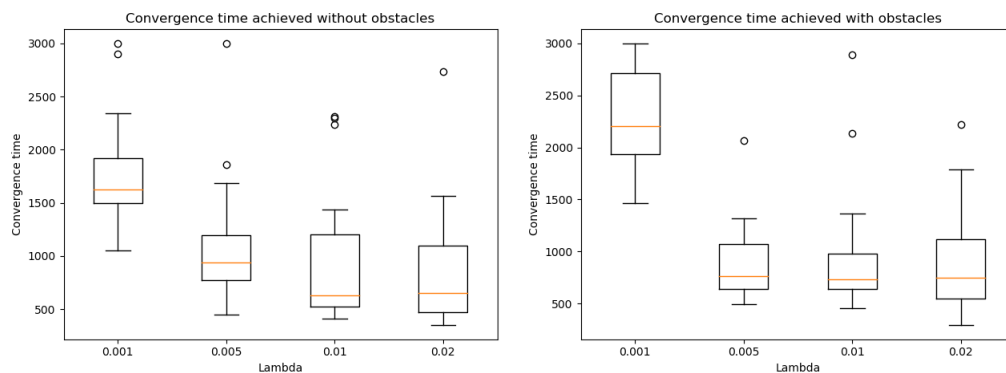


Figure 6: Convergence time as a function of the excitation rate λ : (left) regular lattice (right) lattice with obstacles.

two possibilities generally occur: we either have a rather steady increase of the criteria, which indicates that the amoebae are sorted into two groups, or the values will stagnate for a while (as shown on Fig. 4), and then find the sorted stable.

The stochastic nature of the model makes that some evolutions may last longer to reach the sorted state. For instance, when three main groups of amoebae are formed, with one group of one type and the other two of the second type, the value of homogeneity remains around 50% for a moment until the two groups of the same type can merge.

The *compactness* criterion allows us to complete our analysis of the clustering process. Recall it estimates the average relative distance of the connected components of the amoebae to the closet main component. In the beginning of the simulation, since amoebae are scattered randomly in space and these main components are not stable and the criterion is of little relevance. As one can observe, the compactness criterion becomes relevant only when larger groups of amoebae are formed, in which case its values become greater than 25%.

Note that when the clustering has occurred (that is, when we have only two groups), the homogeneity approaches 100% but the compactness stabilises at lower values (between 40 and 50%). Indeed, the rules we imposed on the movements of amoebae (agitation, limit of their number per cell) do not allow them to form perfect components.

3 Robustness of the model

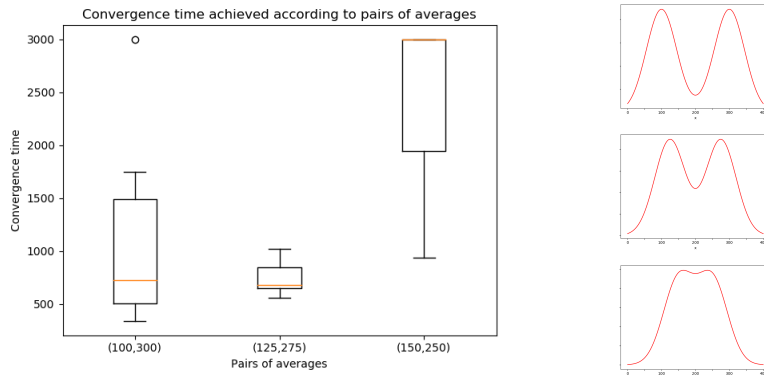


Figure 7: (left) Convergence time for three distribution laws which are sums of two Gaussian laws (right) Representation of the distribution laws.

Data. Let us examine how the model responds to variations of the distribution of data. We take again the interval $I = \llbracket 0, 400 \rrbracket$ and generate the data distribution according to the sum of two Gaussian laws of parameters (μ_1, σ_1) and (μ_2, σ_2) . We fix the value of Λ to $\Lambda = 40$ and the standard deviation to $\sigma_1 = \sigma_2 = 0.4$.

We perform different experiments by varying (μ_1, μ_2) . We define the *convergence time* as the number of time steps that are needed to attain both a homogeneity greater than $h^* = 0.8$ and compactness of $\kappa^* = 0.25$. These values are fixed empirically by observing the situations which correspond to a good clustering quality and a relatively stable global state. Since we aim at achieving a rapid convergence, we stop the simulation after 3000 time steps.

The results are presented on Fig. 7. We observe that the clustering is rapid when the two Gaussian laws are well separated $(\mu_1, \mu_2) = (100, 300)$. More surprising, we note that when the averages are made closer $(\mu_1, \mu_2) = (125, 275)$, and even though the two laws begin to overlap, the convergence time remains low and is even better than the previous case. A limit case is observed for $(\mu_1, \mu_2) = (150, 250)$ where the convergence time jumps above the limit value of 3000 steps. A visual inspection of the simulations and an observation of the evolution of the criteria shows that on the two first case, the clusters remain stable and well-sorted after the convergence time. Of course, Λ values are more suitable for some datasets but we here fix Λ for the sake of brevity.

We also examined the behaviour of the model when the data is a superposition of three Gaussian laws: in this case, we observe that the clustering process occurs as expected by sorting the data into three groups, without any significant delay when λ is set to the appropriate value.

Initial conditions. One may also examine what happens if a large number of amoebae is initially introduced. A new series of simulations is carried out with the following initial condition for an initial density between 10% and 75%: a cell has a chance of initially containing an amoeba containing a data. We empirically observed that although the movements of the amoebae are more constrained and although the propagation of cell excitations are more difficult, the model succeeds in achieving the clustering operation in times that are comparable to those expressed above. It is only when the initial density reaches $d_{\text{ini}} = 0.75$ that strong difficulties appear and that the convergence time significantly rises. This suggest that the size of the lattice should always be set in agreement with the quantity of data that needs to be processed; with say at least three times as many cells as amoebae.

Obstacles. As mentioned earlier, one of the motivations for the use of bio-inspired models is their robustness to errors and failures in the computing medium. We thus consider the case where obstacles can be present on the lattice, that is, cells that cannot transmit any information which cannot hold any amoeba.

Simulations in such conditions are represented on Fig. 6: we consider different values of the emission rate λ with and without obstacles. One can observe that the overall behaviour is not perturbed by the presence of obstacles, although some isolated amoebae may take a longer time to join other groups. The convergence of the amoebae can be slowed down but we empirically observed that the process always manages to achieve the clustering task.

4 Openings

These first tests are rather encouraging and show that our model remains robust to various modifications of its settings. The particularity of our bio-inspired method is its specific use of *randomness*: the moments where the amoebae emit their waves is determined randomly, and should be set in order to allow a good balance between “speaking” and “hearing”. The clustering task is also realised by achieving an appropriate setting to determine whether the data seen in the excitation waves are attractive or repulsive. We here only presented a prototype tested on a rather simple clustering task with only two groups to form. The model thus now needs to be evaluated on more realistic datasets and its robustness should be examined thoroughly in more stringent conditions. Our first informal observations on other conditions are rather encouraging and such explorations could be performed by all researchers who interested in the exploration of the field of bio-inspired cellular models.

References

- [Fat10] Nazim Fatès. Solving the decentralised gathering problem with a reaction–diffusion–chemotaxis scheme. *Swarm Intelligence*, 4:91–115, 2010.
- [Fat16] Nazim Fatès. Collective infotaxis with reactive amoebae: A note on a simple bio-inspired mechanism. In Samira El Yacoubi, Jaroslaw Was, and Stefania Bandini, editors, *Proceedings of ACRI 2016*, volume 9863 of *Lecture Notes in Computer Science*, pages 157–165. Springer, 2016.
- [Fuj13] Osamu Fujita. Metrics based on average distance between sets. *Japan Journal of Industrial and Applied Mathematics*, 30:1–19, 2013.
- [HM07] Julia Handl and Bernd Meyer. Ant-based and swarm-based clustering. *Swarm Intelligence*, 1(2):95–113, 2007.
- [NB66] John von Neumann and Arthur W. Burks. *Theory of Self-Reproducing Automata*. University of Illinois Press, Urbana, USA, 1966.
- [NP14] Satyasai Jagannath Nanda and Ganapati Panda. A survey on nature inspired metaheuristic algorithms for partitionial clustering. *Swarm and Evolutionary Computation*, 16:1 – 18, 2014.
- [VBD10] A. Voß-Böhme and A. Deutsch. The cellular basis of cell sorting kinetics. *Journal of Theoretical Biology*, 263(4):419–436, 2010.