TUTORIALS

Modeling a biological switch

Jean-Pierre Mazat & François Képès
La Colle sur Loup March 30th – April 3rd

Lambda Phage Switch: a spatialised and stochastic approach.

Pascal Ballet* Pascal Redou+ Sébastien Tripodi*
and Vincent Rodin*

European University of Brittany - UEB
UBO, EA 3883-LISyC
20 av Le Gorgeu, CS 93837
29238 Brest Cedex 3, FRANCE
Pascal.Ballet@univ-brest.fr, redou@enib.fr, Vincent.Rodin@univ-brest.fr

Abstract

The use of a multiagent system (MAS) to model and simulate biochemical reactions is useful for several reasons. Four of them are, first, it is possible to represent directly each molecular type using a specific agent type, second, it is possible to duplicate the agent types in differents quantities to retrieve the real molecular concentrations, third, it is possible to make the agent reacting with each other under certain conditions or events in order to represent the reactions between the real molecules and fourth, the agents take place in a 2D or 3D environment where many situations and phenomena can be simulated.

In this paper we detail different stages allowing the creation of a MAS representing one or more biochemical reactions occurring concurrently in a same environment.

KeyWords: multiagent systems, morphogenesis, embriogenesis, morphoBlock.

1 Introduction

The use of a multiagent system (MAS) to model and simulate biochemical reactions is useful for several reasons. Four of them are, first, it is possible to represent directly each molecular type using a specific agent type, second, it is possible to duplicate the agent types in differents quantities to retrieve the real molecular concentrations, third, it is possible to make the agent reacting with each other under certain conditions or events in order to represent the reactions between the real molecules and fourth, the agents take place in a 2D or 3D environment where many situations and phenomena can be simulated.

In this paper we detail different stages allowing the creation of a MAS representing one or more biochemical reactions occurring concurrently in a same environment.

In this paper we describe in section 2 our multiagent system (MorphoBlocks, Environment, Interactions and movements). Then in section 4 we present three different simulations we obtain by changing the parameters of our agents and a fourth simulation focused on the embriogenesis of an abstract multicellular organism. Finally, in section 5, we discuss on the interest of this approach with its advantages, drawbacks and possible enhancements.

2 Multiagent System

In this section, we first describe the agents their properties. Then we focus on the 2D grid which represents the environment. Finally we deal with the decision algorithm allowing the agents to interact and move in order to reproduce biochemecial mechanisms.

2.1 agents

In this part, we present the different properties of an agent (type, size, position, life time, diffusion speed). Then we describe its abilities to modify and perceive its environment in order to reproduce chemical reactions.

An agent is an autonomous entity able to perceive its local environment, to take decisions and to modify its local environment (see figure 1).

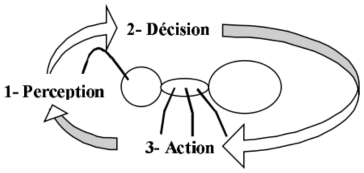


Figure 1: Life cycle of an agent: perception / decision / action. An agent is located in an environment with others agents to create a multiagent system (MAS).

An agent A has a $Type$ and a position (i, j, k) in a 3D grid. It also has a $Core$ surrounded with $AdhesiveReceptors$. So A can be characterised by a 4-uplet:

$$< Type, Position(i, j, k), Core, \{AdhesiveReceptors\} >$$

A $Core$ is the centre of the agent. A grid site contains no more than one $Core$.

An $AdhesiveReceptors$ is defined by $L_{Side,TypeA,ShapeA}$ with:

- $Side$ is an integer $\in [1, 8]$ that indicates the position around the core of the $Ligand$ with 1 = right, 2 = right-top, 3 = top, 4 = top-left, 5 = left, 6 = left-bottom, 7 = bottom and 8 = bottom-right.
- $TypeA$ is a word modelling the type of receptor. Only $AdhesiveReceptors$ with the same type can interact.
- $ShapeA$ is a decimal $\in [0, 1]$ indicating the concavity of the $AdhesiveReceptor$. If $ShapeA = 0$, the $AdhesiveReceptor$ is concave, if $ShapeA = 1$, the $AdhesiveReceptor$ is convex and if $ShapeA = 0.5$, the $AdhesiveReceptor$ is flat. The probability of adhesion during one time step between two

agents A_1 and A_2 having the $AdhesiveReceptor_1$ $L_{Side=1,TypeA=``T'',ShapeA=0.9}$ and $AdhesiveReceptor_2$ $L_{Side=1,TypeA=``T'',ShapeA=0.2}$ is the $abs(ShapeA_1 - ShapeA_2) = abs(0.9 - 0.2) = 0.7$.

The agents can bind together and form large rigid structures. A large *rigidStructure* is defined as a set of bound agents having the same movement in the environment. The section 2.2 focus on the *rigidStructures* computation.

2.2 Rigid structures

In this part, we describe how the *rigidStructures*, made of agents, move inside the environment. A *rigidStructure* can move in four directions (left, right, up, down) or can stay at its current position. The choice of the direction is probabilistic. It is computed using the diffusion probability of each agent which composes the *rigidStructure*: $probability = \prod p_i$ with p_i the probability of movement of each i agent composing the *rigidStructure*.

A rigid structure is made with adherating agents. All the *rigidStructures* are recomputed at each time step according to the probability of adhesion of each *AdhesiveReceptor*.

A *rigidStructure* contains at least one agent.

The algorithm to realise a simulation step is the following:

1. All the *rigidStructures* are determined thanks to the *AdhesiveReceptor* of each agent.

2. One untreated *rigidStructure* is chosen at random among all the others. The others are all frozen during the computation.

3. The probability of movement pm for the selected *rigidStructures* is evaluated.

4. A random number $rnd \in [0, 1]$ is generated. If $rnd < pm$ then a direction is chosen among left, right, up, down and current position. Else, the current position is chosen.

5. If the sites where the *rigidStructure* want to go are all free, the agents are moved to their new positions. Otherwise thes agents composing the *rigidStructure* stay at its current position.

6. The *rigidStructure* is marked as already treated.

7. More *rigidStructure* to treat? If yes, then go to point 1.

8. End.

The agents are not only in a direct interaction thanks to their *AdhesiveReceptor* but use the environment to modify themselves. The section 2.3 presents this environment and the section 2.4 explains how agents can be modified (replaced, created or removed).

2.3 Environment

The environment is a 3D grid. It allows the agents to interact. It can be seen as a shared blackboard. Each site of the grid can contains one kind of data. The first is the reference of the agent *Core*. If a grid element do not carry any *Core*, the reference is *empty*. A site contains no more than one *Core*.
When an agent changes its position in the grid, the *core* is removed from the old position to the new one to reflect the new organisation. This environment allows a quick treatment to recursively create *rigidStructures* from agents to their neigborhoods. It also allows the agents to interact and dynamically modify, add or remove agents.

The manner how the agents interact in the environment is described in the section 2.4.

2.4 Interactions

The agent probability of movement reproduces the diffusion of a particule thanks to a random walk. We also need behaviours allowing to reproduce the different molecular mechanisms like the molecule transformation, the molecule degradation, or the molecule creation. For that we create a reaction algorithm able to take into account specific configurations of agents inside the environment. For example:

- molecule replication: A ⇒ A+A (with a probability p chosen by the modeller). This means that if at time t the environment contains an agent A, at time t+1 a new agent A is created close to the other (with the probability p). This is repeated for each agent of type A presents at time t.
- molecule transformation: A ⇒ B (the agent A is replaced with the agent B).
- molecule degradation: A ⇒ $\emptyset$
- molecule absorbtion: A + S ⇒ B (the agent A is replaced with the agent B when S is in contact with A)
- molecule creation: A ⇒ A+S (the MorphoBlock A produces S)

In order to have quantitative results of the molecular reactions occuring in the MAS, we have to precisely compute the values of the probabilities of the behaviours.

If we wish to put a simple reaction between the agents A and the agents B to give an agent C, we have to:

- define the reacting neighbourhood of an agent,
- define the probability that the reaction occurs,
- indicate the reactive and the produced agents.

For example, A+B ⇒ C (p) signifies that if A is the neighbour of B then the reaction can occur with the probability p. This reaction will result in the removal of A and B and the creation of C.

Let us take the following reaction: A+B =0.2=¿ C
The previous example gives us 3 possible situations:

1. no reaction will occur (with p=0.2, this is the most probable situation),
2. the reaction occurs.
3. A conflict is possible. For example, the environment can contains several agents of type A and several agent of type B and this reaction can occur in several locations in the environment. Many A and B agents can be in contact and can give many C. Sometimes, an agent of type A can be in contact with 2 B. In this case, two different reactions can happen but only one is selected (to be relevant with the conservation of matter). The selection is equiprobable among all the possible reactions which are in conflict.

To initiate and execute the movements and reactions of the agents in the environment, the following algortihm is executed:

1. listing of all the possible reactions. This list is made with reactions that can occur. A reaction can take place if the agents are in contact and if the probability of each reaction is verified.
2. selection of one reaction at random,
3. realisation of the reaction if possible. Here, the possible conflicts are solved: if a previous reaction has already consumed the reactives, the current reaction will not occur.
4. the reaction is removed from the list
5. if there is more reaction to apply, go to point 2.

Another important point concerns the reaction speed. In our MAS it is expressed with a probability p. However, for biochemists, the speed of a reaction is characterised by a reaction speed k. The following section 2.5 will deal with an approximative method to convert the k parameter into a probability p.

2.5 From reaction speed to probability

Consider the irreversible first-order reaction
A + B ⇒ C, with speed k.

We suppose that our environment grid contains n grid elements. We put randomly two types of agents: a number of a agents of type A and a number of b agents of type B. Let X the number of contacts between A and B (there is a contact between 2 agents when they are at a distance equals to 0). The probability P that exactly X encounters occur is

$$P[X = p|(a, b)] = C(p, a)C(b - p, n - a)/C(b, n)$$

$$where$$

$$C(k, n) = n!/((n - k)!k!)$$

This makes possible to obtain the average number of encounters at each step of simulation, and with the assumptions made, we get:

$$E[X|(a,b)] = (ab)/n$$

Let p the probability for creating a molecule of type C when two molecules of types A and B encounter. The average evolution $Cmoy$ of the number of molecules of type C at each simulation time step $dtsim$ is:

$$Cmoy = (pabdtsim)/n$$

In terms of concentration, if V if volume of the environment, we get:

$$[C]moy = (pabdtsim)/nV$$

It is possible to compare this approach with the differential equation approach. Denote by $dtnum$ the step of the numerical resolution for the differential equation which represents the product evolution. We have, with a simple Euler method,

$$C = kdtnum[A][B] = (kabdtnum)/V$$

This gives, identifying variations of C, and after a basic simplification,

$$p = kn/Vdtnum/dtsim$$

It is clear that p is supposed to be less than or equal to 1, which imposes a constraint on the mesh depending on the reaction speed k.

A more simple computation is for a simple degradation. Considere the following chemical reaction with a speed k_1: A =k_1=¿ B

Let $T_1/2$ the half-time of A. $dA/dt = -A.k_1$ $dA/A = -k_1.dt$ $[lnA(u)]0t = -[k.u]0t$ $A(t) = e - lnA(0).e - k.t = -A(0).e - k.t$ $att = T_{1/2}, A(t) = A(0)/2$ $A(0)/2 = -A(0).e - k.T_{1/2} <=> T_{1/2} = ln2/k_1$

For one agent A, during an interval of time = T1/2, there is 1 chance over 2 it becomes B. In other terms:

- For a simulation time step = $T_{1/2}$, p1 = 1-(1/2)1
- For a simulation time step = 2.T1/2 , p1 = 1-(1/2)2
- For a time step t, p1 = 1 2-t/T1/2 = 1 - 2-t.k/ln(2)

Consider the following enzymatic reaction:

E + S ¡– k_1, k_{-1} –¿ ES – k_2 –¿ E + P t = 10^{-4} s, n = 2500 $k_1 = 1$ gives the probability: $p_1 = 2.5*10-1$ $k_{-1} = 1$ gives the probability: $p_{-1} = 9.9995*10-5$ $k_2 = 1$ gives the probability: $p_2 = 9.9995*10-5$

Four agents allow the representation of the different entities encountered in this reaction. The agent E represents the enzyme, the agent S the substrate, the agent ES the complex formed by the E and S and the agent P represent the product of the reaction.

At time t = 0, we place 10 agents E (green) and 373 agents S (red) on a 100x100 grid. We launch the simulation and at t=6760 time steps, more than 50 percent of the substrates S have been transformed in products P. Finally, at

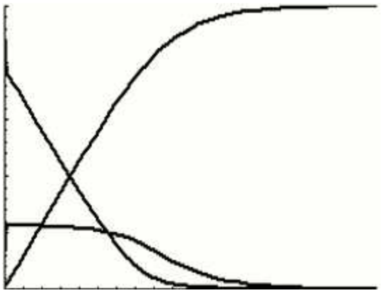


Figure 2: Runge-Kutta (order 4) resolution of the enzymatic system.

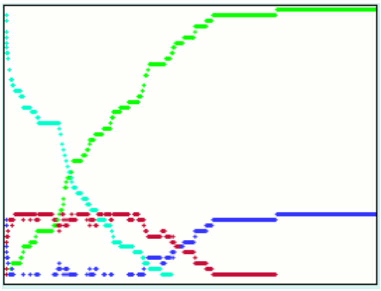


Figure 3: MAS resolution of the enzymatic system. It is simulated with NetBioDyn (see section 3).

t=35200 all the S agents have been changed in P agents (see the figure 3). This simulation is similar with the Runge-Kutta resolution (see the figure 2).

In the section 3 we show the software NetBioDyn we create to do MAS simulations.

3 NetBioDyn software

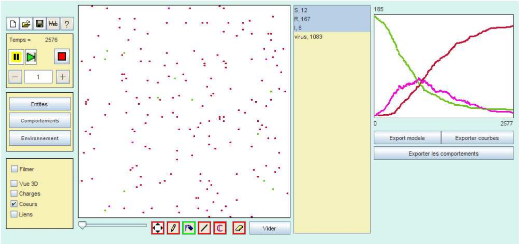
We develop a software named NetBioDyn based on the previous developped principles. It is made to help the design of biological model thanks to a user-friendly interface (see 4). It is made in $java^tm$ and is available as an *applet* at the URL: http://netbiodyn.tuxfamily.org Two versions are now online, one in 2D and one in 3D (see 5).

In the section 4 we study the lambda phage switch using netBioDyn.

4 Simulations

Switch du phage lambda

Dans la bactrie E. Coli infecte par le virus phage lambda, deux tats sont possibles: lythique ou lysognique. L'apparition d'un tat plutt qu'un autre est l'aboutissement de ractions se droulant au niveau de l'ADN de la bactrie et plus prcisment au niveau des gnes cI et cro. Le gne cI exprime la protine CI et le gne cro la protine CRO. Un mcanisme de rtroaction positive est observ autour des squences or1, or2 et or3. Si la protine CRO se fixe sur la squence or3, alors



Temps = 2576

S, 12
R, 167
I, 6
virus, 1083

☐ Filmer
☐ Vue 3D
☐ Charges
☑ Coeurs
☐ Liens

Figure 4: Interface of the netBioDyn software.

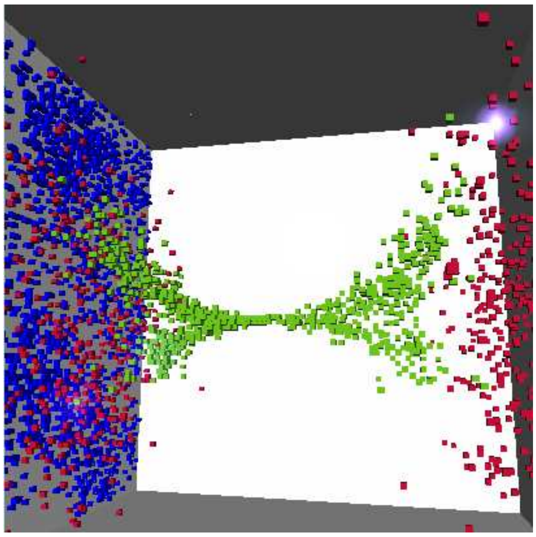


Figure 5: 3D interface of the netBioDyn software.

seule la protine CRO peut tre retranscrite. De le mme manire, si la protine CI est fixe sur la squence or1, alors seule la protine CI peut tre retranscrite. Le mcanisme de transcription du gne cro n'est possible que si une protine CRO est fixe sur le site or3. Le mcanisme de transcription du gne CI n'est possible que si une protine CI est fixe sur le site or1.

Ce systme induit au bout d'un certain temps soit l'expression exclusive de CRO (tat lysognique) soit l'expression exclusive de CI (tat lythique).

Pour la simulation, les entits suivantes sont cres:

$site_or321$ de mobilit nulle et de 1/2 vie infinie. Cette entit regroupe les 3 sites or1, or2 et or3. gne_c1 de mobilit nulle et de 1/2 vie infinie. gne_cro de mobilit nulle et de 1/2 vie infinie. la protine $C1$ de mobilit valant 1 et de 1/2 vie gale 1000. la protine CRO de mobilit valant 1.0 et de 1/2 vie gale 1000.

Un complexel seq_ADN regroupant respectivement le gne_CI, le $site_or321$ et le gne_cro se touchant est cr.

Afin de reproduire les mcanismes connus du switch du phage lambda, deux ractions semi-situes sont crs, l'un concernant la transcription de la protine CRO et l'autre sur la transcription de la protine CI:

$trans_CRO$: $(site_or321) + gene_c1 + gene_cro + CRO + vide = 0.1 => (site_or321)+gene_C1+gene_CRO+CRO+CRO$ $trans_C1 : (site_or321)+gene_c1+gene_cro + C1 + vide = 0.1 => (site_or321) + gene_c1 + gene_cro + C1 + C1$

Le comportement $trans_CRO$ indique qu'une protine CRO est transcrite si le $site_or321$, entour du gne_cI et du gne_cro, est fix par une protine CRO et seulement elle (vide).

Le comportement $trans_CI$ indique qu'une protine CI est transcrite si le $site_or321$, entour du gne_cI et du gne_cro, est fix par une protine CI et seulement elle (vide).

La mise en place de l'tat initial ncessite le placement de quelques molcules CRO et CI pour initier les ractions semi-situes dcrites prcdemment. En effet, dans ce systme simplifi la transcription de CRO ou de CI ne peut s'effectuer que si respectivement une protine CRO ou une protine CI se fixe sur le $site_or321$. La figure suivante montre l'tat initial de la simulation dans une grille de 30x30:

La simulation donne les rsultats suivants:

Les courbes du nombre de protines CRO (bleu) et CI (rouge) au cours du temps pour cette simulation sont:

Le switch est effectif 14000 puisqu' cette date plus aucune protine CRO n'est prsente dans la simulation. Une priode d'instabilit est observe initialement, montrant le caractre probabiliste de la simulation.

Voici des courbes du nombre de protines CRO (bleu) et CI (rouge) issues d'autres simulations avec le mme tat initial:

5 Discussion

The multiagent approach is interesting to study the biological phenomena for different reasons. First, it can reproduce the stochastic characteristic of the living (especially coming from the Brownian motion engendered by the thermic agitation). Second, we can use qualitative data instead of quantitative ones. Third, we can observe the creation of patterns in a spatialised environment. Fourth, we can simulate systems where there are few entities. Fifth, we can

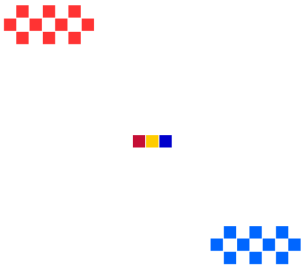


Figure 6: Initial state of all the simulations made. At the upper left there are the *CRO* proteins. At the bottom right there are the *CI* proteins. The genes are at the center of the figure.

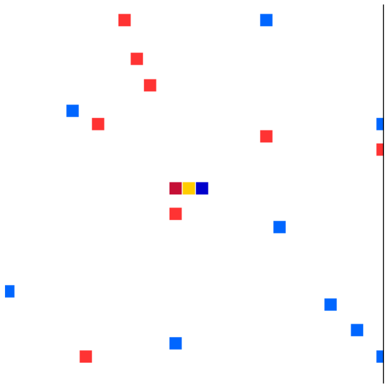


Figure 7: Exemple of the environment after 163 simulation steps.

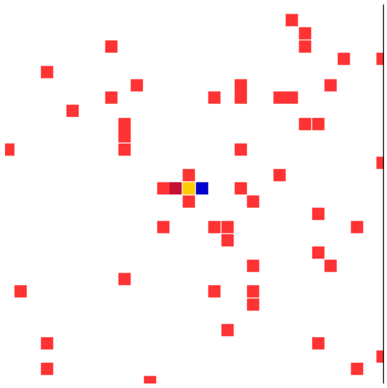


Figure 8: Exemple of the environment after 15008 simulation steps. The *CRO* proteins are dominants.

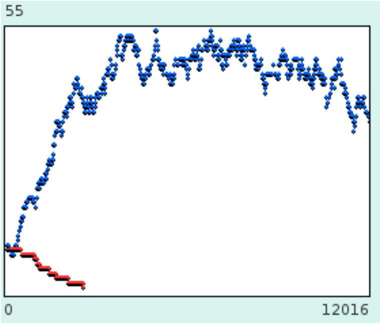


Figure 9: First simulation where the switch goes to the lityc state.

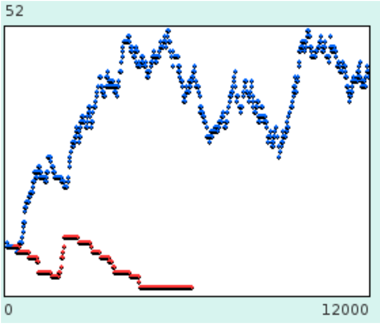


Figure 10: Second simulation where the switch goes to the lityc state.

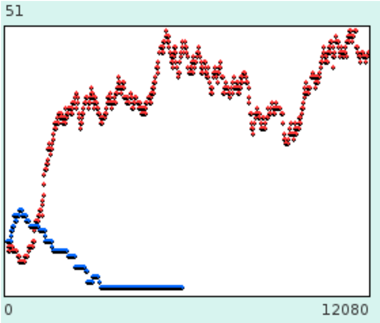


Figure 11: First simulation where the switch goes to the lysogenic state.

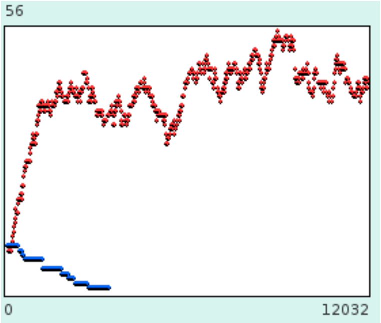


Figure 12: Second simulation where the switch goes to the lysogenic state.

easily add, modify, suppress agents or behaviours during the simulation (see in-virtuo definition.

A interesting use of our software is the extraction of the curves representing the concentration of biological species according to the time. A mathematical analysis of these curves is an interesting perspective because this can allow to describe the system in term of mathematical equations. Thus, we can use the mathematical tools to analyse the dynamic of the studied systems (steady state, cycles, etc).

To facilitate the design of simulations, we developed a graphical interface. Thanks to this interface, the user can create MorphoBlock and behaviours without any line of code. The software is accessible from the Internet [1] in the form of a Java applet.

6 Conclusion

We have presented the paradigm of RigidStructur to reproduce large biological structures in a simple way.

This approach have several drawbacks. The first problem is the grid where rotations are not accurate enough. It could be improved in different ways: first by increasing the size of the core. It will be interesting to add a core made with many site. Another problem is the design of very large simulations. A solution can be the use of the multicore on the next generation of microprocessors. Our agents are very simple reactive agent (without memory nor planning). Nevertheless, the simulations show different types of self-assembled structures. Moreover we can visualise the different stages of the structure formations during the simulations. The simulations show interesting results which can be compared, in an abstract point of view, to biological processes like the switch of the lambda phage.

Acknowledgments: we thanks the ANR MitoSCoP and the Région Bretagne for their financial contribution.

[1] http://netbiodyn.tuxfamily.org