Towards a Hybrid Metabolic Algorithm

Luca Bianco and Federico Fontana

University of Verona
Department of Computer Science
15 strada Le Grazie – 37134 Verona, Italy
bianco@sci.univr.it, federico.fontana@univr.it

Abstract. During these years stochastic algorithms have deserved much attention from the computational biology research communities. In this paper we derive a hybrid version of the formerly known Metabolic Algorithm that is enriched with stochastic features, whose impact on the dynamics of the system is as prominent when the amount of metabolite becomes smaller. This hybrid procedure represents a first attempt to let the Metabolic Algorithm deal with low concentrations of substances according to a non-deterministic policy.

1 Introduction

The simulation of a metabolic process relies on several, sometimes well-established methods and algorithms that either compute its evolution deterministically, as it happens with methods discretizing a Reaction Rate Equation, or stochastically, as it happens with algorithms solving or approximating the Chemical Master Equation [1, 2]. The latter algorithms are quite accurate but computationally expensive, given the individual handling they must do of each molecule, and provided that only several repeated realizations of the same simulation in principle provide sufficient information about the expected behavior of a system. For this reason, approximated versions of these algorithms have been proposed in order to diminish the computational burden of the stochastic approach [3].

Less is known about the possibility to gain efficiency, without loosing too much in accuracy, by using *hybrid* algorithms capable of mixing the deterministic and the stochastic paradigms together. Such an approach is inherently hard to deal with due to the theoretical and technical difficulties that arise when a system, whose kinetic rates range among different scales, is split into a multiple observation-level model accounting for different computation strategies depending on the level of observation [2]. Nevertheless, such an approach has been already pursued for simulating complex biochemical systems [4, 5] and also in the processing of proteomic data[1] [6].

Our work here is still far from adding substantial knowledge about this possibility. Despite this we will show how a hybrid strategy to biochemical system

[1] An interesting introduction to the use of hybrid algorithms in computational biology can be found online at *http://www.bioinfo.de/isb/2004/04/0024/main.html*.

modeling can be implemented within MP systems, by extending their deterministic evolution mechanism in order to include randomness that is always present in biochemical systems.

Metabolic P (MP) systems [7, 8] are rooted in the theory and formalism of P systems [9, 10], to which they couple a deterministic computational strategy called P Metabolic (MP) Algorithm [11, 12, 8]. P systems have been envisioned to find solutions for several problems [13]. In the meantime they have been a fertile ground for the birth of stochastic algorithms for the representation of the evolution of biochemical systems.

The P system community has approached the question of stochastic evolution in several ways. One strategy is known as Dynamical Probabilistic P systems, and employs maximal parallelism both at the rule and object level: there, the probability of tossing a rule is dynamically calculated by starting from the multiset of objects that are present in the system during a transition, as well as from the kinetic constant associated to the rule. Another is called Multi-compartmental Gillespie's algorithm and its aim is to extend the Gillespie algorithm to a multi-compartment environment as it happens for P systems having more than one membrane [1, 14, 15].

By exploiting the versatility of the MP algorithm, we can straightforwardly integrate a stochastic strategy for choosing the strength of the rules governing the system evolution, in a way that the smaller the amount of a substance is, the stronger the effects of randomness. In practice this is made by altering the deterministic character of the reaction maps proportionally to their magnitude [8]. In the end of the paper this hybrid algorithm is tested upon traditional case studies such as the BZ reaction and the Lotka-Volterra dynamics [16, 8, 17, 18].

In the following of the paper we assume the reader to be friendly enough with the notation and the formalism of MP systems, whose leading ideas are that:

- the system evolves by allocating to each evolutionary rule object amounts that play the role of reactants, as well as by obtaining from such rules object amounts that play the role of products;
- nonlinear functions of the state of the system (that is identified by the amount of every object), called *reaction maps*, are computed at the beginning of each system transition for assigning object amounts to the rules.

For further details on MP systems and the MP algorithm (shortly MPA) we refer the reader to [7, 8].

2 The Algorithm

The idea of the *hybrid algorithm* is to switch from a completely deterministic (MPA) to a completely stochastic approach (s-MPA) depending on the size of the population dealt with by each rule. A threshold τ is used to control the switch between the two strategies and in this way the whole system becomes a stochastic-deterministic hybrid system (h-MPA). If the population is small compared to the threshold, the stochastic strategy should be preferred, otherwise

the deterministic strategy is able to provide an acceptable approximation of the dynamics and is thereby preferable.

In principle, for each rule a deterministic or stochastic strategy has to be chosen according to the size of the population it deals with. Let us suppose to have a rule $r : XY \rightarrow ...$, and to fix a threshold τ. If

$$\min(q(X), q(Y)) < \tau$$

the strategy of application of r is chosen to be stochastic, else it is deterministic—$q(Z)$ denotes the amount of the species Z present into the system. Note that this minimum gives the bottleneck of the reaction, but it is different from the limiter of the standard metabolic algorithm because here we do not take into account the strength of the rules. A population, then, undergoes a stochastic dynamics if its size is smaller than the threshold.

Of course, due to cooperation, a population can undergo a stochastic dynamics even if it is bigger than the threshold, but it is involved in reactions dealing with at least one reactant whose total amount in the system is below the threshold.

What do we mean by stochastic strategy? The idea is to keep a "population perspective" of the dynamics, as in the spirit of the metabolic algorithm. Accordingly, a stochastic strategy for the simulation of a rule r with the system in state $\mathbf{s}$ consists in:

i) evaluating the reaction map $F_r(\mathbf{s})$ as in the deterministic MPA;
ii) picking up a random number from a probability distribution depending on $F_r(\mathbf{s})$, let us call this number v;
iii) applying the rule as in MPA by using v instead of $F_r(\mathbf{s})$ as a reaction map,

where a generic state $\mathbf{s}$ can be thought as a vector of concentrations of all the elements of the system (that are assumed ordered) and it is denoted as $s\text{-}MPA(\mathbf{s})$ (we denote with $MPA(\mathbf{s})$ the purely deterministic evolution of state $\mathbf{s}$).

This pseudo-algorithm gives an intuitive idea of the process, but it does not simulate properly the application of rule r (for example, we need to specify stochastic reaction maps of all reactants of rule r). In Subsection 2.3 a full description of the hybrid algorithm is given. Before entering the description of the algorithm, another preliminary question need to be addressed.

How do we quantify the dependency of the probability distribution on F_r? The idea is to respect somehow the shape of the (deterministic) reaction map in the random choice of the stochastic reaction map. This is because reaction maps should take into account the features of the interactions between elements of the system and with this respect it seems reasonable to consider them as "independent of the scale" and thereby valid also for small populations of objects. Nevertheless, the generality of the approach allows the modeller to specify a different shape for the reaction maps employed in the stochastic part of the algorithm, but since this is not limiting, here we will not exploit this capability.

One possible implementation of the random step (ii) is to generate a random number v by using a pseudo random sequence generator (PRSG) with a gaussian

distribution of mean $F_r(\mathbf{s})$ and a suitable variance, allowing a certain degree of variability in the dynamics.

2.1 PRSG

Standard Matlab (but not only it) provides a primitive for reckoning a (pseudo) random number chosen from a normal distribution with mean zero and variance one. If we denote with v_{nor} such a random number when obtained from a normal distribution, then

$$v_{rnd} = m + \sigma \cdot v_{nor}$$

is a (pseudo)random number chosen from a gaussian distribution with mean m and variance σ^2.

This expression does not necessarily produce positive values. As reaction maps cannot assume negative values, our PRSG skips eventual negative values. This strategy introduces a distortion of the gaussian paradigm, and in the future we will look for a more coherent random generation of numbers.

In all experiments we have used the PRSG to obtain a stochastic reaction map by starting from a deterministic one F_r evaluated in a certain configuration $\mathbf{s}$ of the system, thereby we have used $m = F_r(\mathbf{s})$ as mean value and (see two histograms of random sequences of 100000 numbers depicted in Figure 1) with $\sigma^2 = 0.5 \cdot F_r(\mathbf{s})^2$ as variance.

Figure 1 suggests that $\sigma^2 = 0.5 \cdot F_r(\mathbf{s})^2$ is a possible choice for the variance, giving enough variability in the distribution of the random number.

2.2 Deterministic, stochastic or both?

A sudden switch from the deterministic strategy (when the algorithm deals with populations larger than the threshold) to the stochastic one seems to be unrealistic. For this reason the developed strategy configures itself as a continuous switch between the two strategies, with various degrees of determinism (or stochasticity as well).

The idea in this case is to use a sigmoid function, whose input is the threshold and a population value and whose output is the degree of determinism of the system (i.e., a real value d in the unitary interval $[0, 1]$ determining the rate of change reckoned by means of the deterministic algorithm). Of course, the value $1 - d$ is the degree of stochasticity of the system. Given a threshold τ and an input value x specifying the population size of the species considered, the value of the determinism degree d produced by the sigmoid function can be computed as

$$d = \frac{1}{1 + e^{(10/\tau)(\tau - x)}} \tag{1}$$

and, as previously said, it gives the degree of determinism (or stochasticity) of the system. Note that the choice of this sigmoid function is empirical: several other functions can be employed (for example, the steepness of the sigmoid can

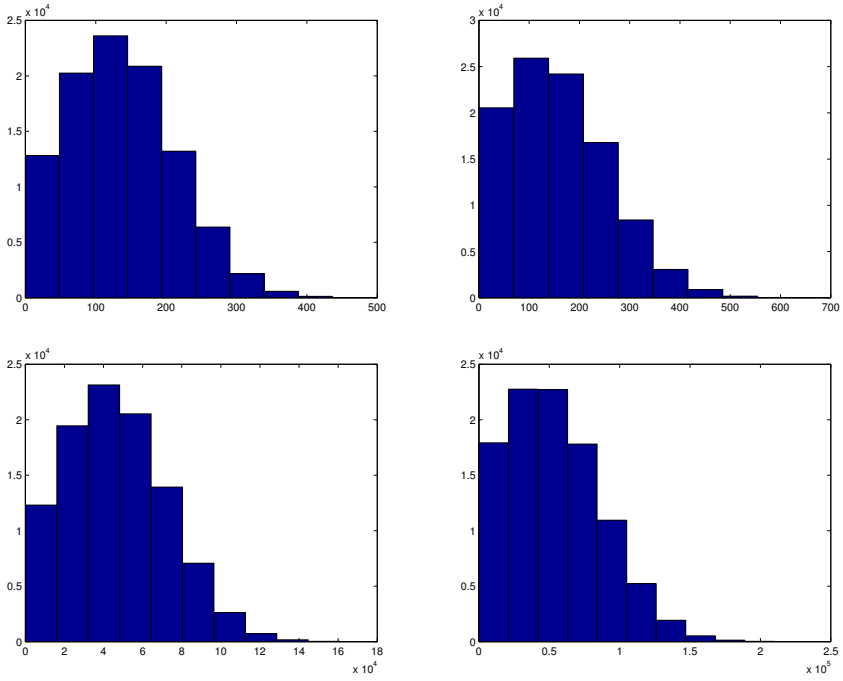

Fig. 1. Histograms of random sequences of 100000 samples with: mean 123 and variance $0.5 \cdot 123^2$ (upper left), mean 123, variance 123^2 (upper right), mean 42000, variance $0.5 \cdot 42000^2$ (lower left) and mean 42000, variance 42000^2 (lower right).

be increased by using $b > e$, such as 8, instead of the Napier's base e of the exponential, or it can be decreased by using $b < e$, such as 2).

In Figure 2 two examples of the sigmoid function (1) are represented; on the left the threshold τ is set to 5000, whereas on the right it is set to 450. We can see that when the population size equals the threshold τ we have a strategy that is half deterministic and half stochastic, for this reason it may be better to consider the following sigmoid function:

$$d = \frac{1}{1 + 4^{\frac{20}{\tau'}(\tau' - x)}} \tag{2}$$

where $\tau' = 0.9\tau$, that is, the 90% of τ.

This newly defined threshold function is shown in Figure 3 (right). It is possible to appreciate that when the population size equals the threshold τ the strategy is deterministic at more than 95%. Although further investigation on the threshold function is needed, in the experiments described in the next section the modified sigmoid has been used.

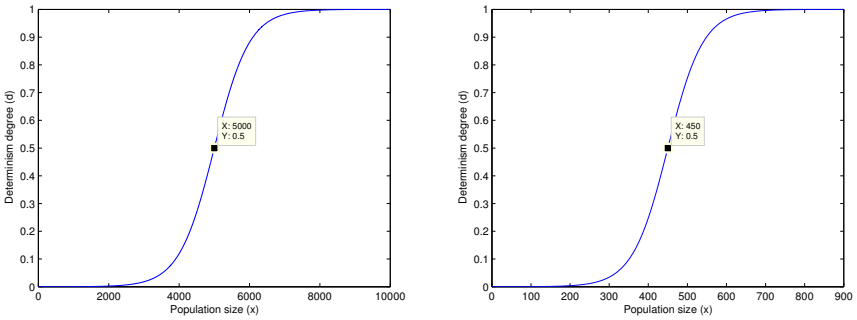

Fig. 2. Sigmoid function (1): population range 0–100000 $\tau = 5000$ (left); population range 0–900 $\tau = 450$ (right).

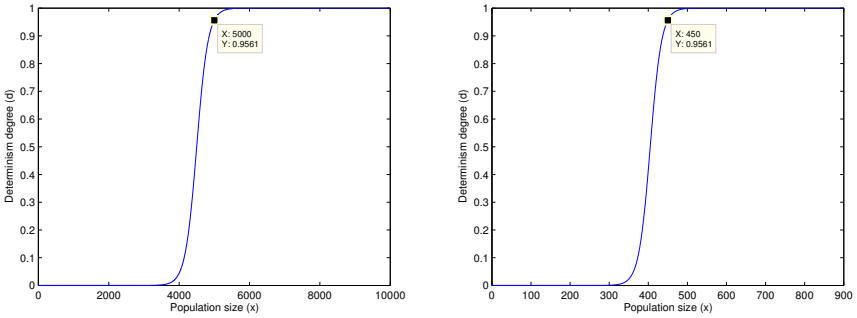

Fig. 3. Sigmoid function (2): population range 0–100000 $\tau' = 5000$ (left); population range 0–900 $\tau' = 450$ (right).

2.3 h-MPA

The idea of the algorithm is to apply each rule in a deterministic way whenever the population involved in it is bigger than a predetermined threshold; whereas, a rule is applied in a stochastic way whenever the population it deals with is below the threshold. For each rule, the total (i.e., not weighted by reaction maps) amount of the bottleneck is reckoned (the bottleneck of a reaction is the reactant whose total amount in the system is the lowest one when compared with the amounts of all other reactants of the rule) and a sigma function is calculated on it in order to obtain the determinism degree of the rule. Then, according to this determinism degree, the rule is applied partially in a deterministic way and partially in a stochastic way.

Let us assume a system specified by means of n rules $r_1, ..., r_n$, defined over the alphabet $\mathcal{A}$, initially in a state $\mathbf{s_0}$, and let τ be the deterministic degree threshold discussed previously. If we denote with $d_1(\mathbf{s}_i), ..., d_n(\mathbf{s}_i)$ the determinism degrees of each of the n rules (calculated as we will see in a while), we can

express the dynamics of the system as the sequence $\mathbf{s}_0, \mathbf{s}_1, ...$ where a transition from a generic state $\mathbf{s}_i$ to the next one can be calculated by computing both the stochastic and deterministic variation for all the rules and then weighting them according to the corresponding determinism degree. In particular, given a rule r_j with $1 \leq j \leq n$, its variation induced on the state $\mathbf{s}_i$, $\delta_{r_j}(\mathbf{s}_i)$, can be calculated as:

$$\delta_{r_j}(\mathbf{s}_i) = d_j(\mathbf{s}_i) \cdot MPA_j(\mathbf{s}_i) + (1 - d_j(\mathbf{s}_i)) \cdot s\text{-}MPA_j(\mathbf{s}_i)$$

where $MPA_j(\mathbf{s}_i)$ is the deterministic application of rule r_j to the state $\mathbf{s}_i$ while $s\text{-}MPA_j(\mathbf{s}_i)$ is the stochastic application of rule r_j to the state $\mathbf{s}_i$.

A transition from a state $\mathbf{s}_i$ to the next one $\mathbf{s}_{i+1}$ by means of the *hybrid* metabolic algorithm can be described by the following meta-code:

Step 0a: *Deterministic reaction maps computation.* The set of deterministic reaction maps is calculated in the current state:

$$F^D_{r_j}(\mathbf{s}_i) \quad \forall j = 1, ..., n\,.$$

Let us denote with $\mathcal{F}^D(\mathbf{s}_i)$ the set of all deterministic reaction maps in the state $\mathbf{s}_i$.

Step 0b: *Stochastic reaction maps computation.* The set of stochastic reaction maps is calculated in the current state:

$$F^S_{r_j}(\mathbf{s}_i) = RND(F^D_{r_j}(\mathbf{s}_i), 0.5 \cdot (F^D_{r_j}(\mathbf{s}_i))^2) \quad \forall j = 1, ..., n$$

where $RND(a, b)$ denotes a gaussian distributed random number, computed as seen before, with mean a and variance b, for $a, b \in \mathbb{R}$. Let us denote with $\mathcal{F}^S(\mathbf{s}_i)$ the set of all stochastic reaction maps in the state $\mathbf{s}_i$.

Step 1: *Single rule variations.* Deterministic and stochastic variations of each rule to each object of the system are computed.

For each couple (r_j, X) with $j = 1, ..., n$ and $X \in \mathcal{A}$, assuming each rule to have the form $r_j = \alpha_j \rightarrow \beta_j$:

o) If $X \notin \alpha_j$ AND $X \notin \beta_j$ then set both the deterministic and stochastic variation induced by r_j on X respectively to:

$$\delta^D_{r_j,X}(\mathbf{s}_i) = 0$$

$$\delta^S_{r_j,X}(\mathbf{s}_i) = 0$$

and goto the step o) of the next couple (if any), otherwise goto step i).

i) Calculate the rate d_j as:

1. Find the bottleneck of reaction r_j:[2]

$$X = \min_{Y \in \alpha_j} q(Y).$$

[2] Other choices for the bottleneck calculation are also reasonable, as they take into account the size of the population involved in each rule instead of the global size of populations, the choice made here simplifies the algorithm.

2. The total amount of the bottleneck is calculated

$$x = q(X).$$

3. The deterministic rate is computed

$$d_j = \frac{1}{1 + 4^{\frac{20}{0.9\tau}(0.9\tau - x)}}.$$

ii) Calculate the deterministic variation induced by r_j on X, $\delta^D_{r_j,X}(\mathbf{s}_i)$ as in the standard metabolic algorithm with reaction maps taken from $\mathcal{F}^D(\mathbf{s}_i)$.
iii) Calculate the stochastic variation induced by r_j on X, $\delta^S_{r_j,X}(\mathbf{s}_i)$ as in the standard metabolic algorithm with reaction maps taken from $\mathcal{F}^S(\mathbf{s}_i)$.

Step 2: *Global variations and system update.* The global deterministic $\Delta^D_X(\mathbf{s}_i)$ and stochastic $\Delta^S_X(\mathbf{s}_i)$ variations are calculated by the weighted sum of all single rules contributions, $\forall X \in \mathcal{A}$:

$$\Delta^D_X(\mathbf{s}_i) = \sum_{j=1}^{n} \delta^D_{r_j,X}(\mathbf{s}_i) \cdot d_j$$

$$\Delta^S_X(\mathbf{s}_i) = \sum_{j=1}^{n} \delta^S_{r_j,X}(\mathbf{s}_i) \cdot (1 - d_j)$$

$$X_{i+1} := X_i + \Delta^D_X(\mathbf{s}_i) + \Delta^S_X(\mathbf{s}_i)$$

where X_i denotes the amount of species X in configuration $\mathbf{s}_i$.

Note that in the case of systems dealing with populations instead of concentrations a rounding policy has to be devised.

3 Case Studies

The case studies presented in this section have been implemented using Matlab, and resulted in numerical simulations whose computation tipically took some seconds.

The first case study is the BZ reaction, that is discussed in two distinct variants, while the second case study is the Lotka-Volterra predator-prey system.

3.1 BZ (model 1)

The first Brusselator model is composed by the following set of rewriting rules and deterministic reaction maps:

$$\begin{array}{lcccl}
r1: & \lambda & \longrightarrow & X & F_{r1} = 10 \\
r2: & XXY & \longrightarrow & XXX & F_{r2} = 900\, q(X)^2 q(Y) \\
r3: & X & \longrightarrow & Y & F_{r3} = 200\, q(X) \\
r4: & X & \longrightarrow & \lambda & F_{r4} = 5\, q(X).
\end{array}$$

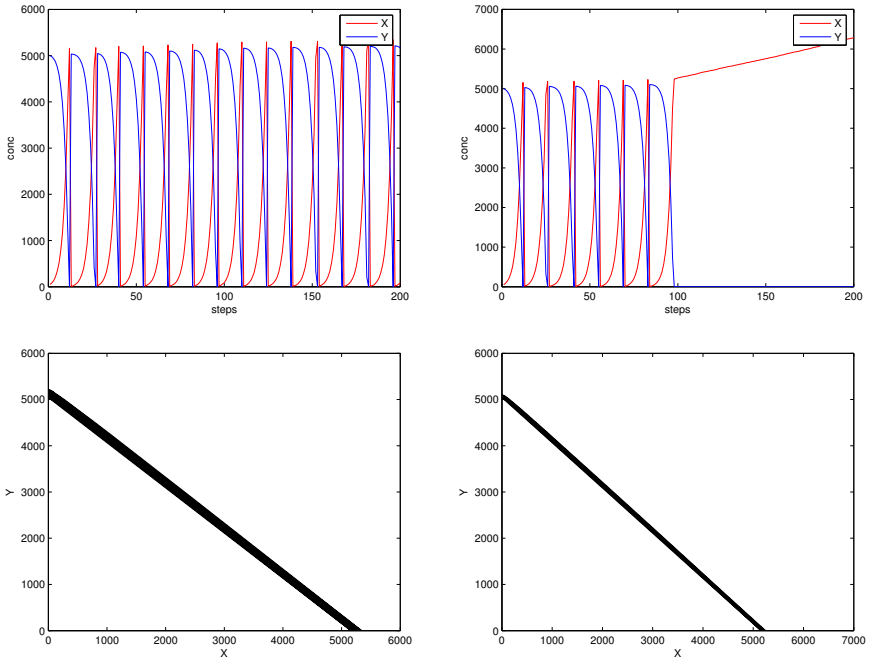

Fig. 4. Two realizations of a hybrid simulation of the Brusselator ($\tau = 2000$, deterministic rate of $r1$ set to 0.5). Corresponding phase space representations are depicted on the lower part of the figure.

The initial amount of objects X and Y are set respectively to 50 and 5000 and both procedures deal with integer objects (the contribution of all rules are floored to the nearest small integer). The deterministic rate that multiplies the variation obtained by the rule is a real value in general, hence for this reason we can have real values in the dynamics.

Note that, due to the fact that λ is not a population, we need a strategy to deal with rule $r1$, that is, we can decide to have a either a completely deterministic feeding of the system, a completely stochastic one, or every degree of determinism.

The algorithm can provide several behaviours depending on the value of the determinism threshold τ. It can show a completely deterministic dynamics if the threshold of determinism is set to 0, or conversely a completely stochastic one, in the case of the threshold $\tau \to \infty$ (or at least larger than the maximum population size in the whole simulation) [8, 11]. Hybrid solutions are obtained for intermediate thresholds (see Figure 4, where $\tau = 2000$ and, on the right, we can observe that the dynamics shows a slight stochasticity in the first instants and then the oscillation is suppressed).

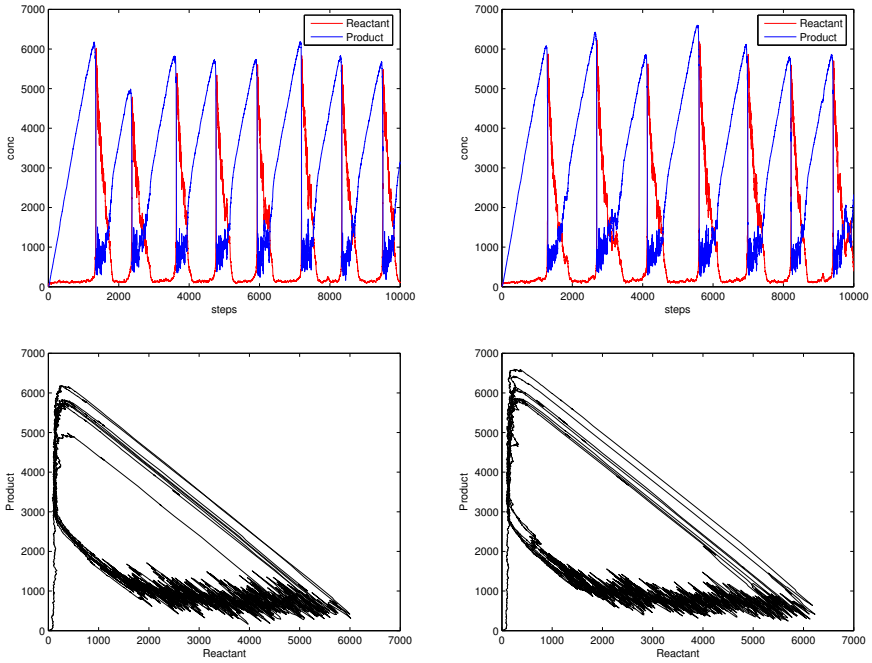

Fig. 5. Two realizations of completely stochastic simulations of the Brusselator ($\tau = 10^{99}$). Corresponding phase plots are shown in the lower figures.

3.2 BZ (model 2)

The second Brusselator model deals with the following set of rewriting rules and deterministic reaction maps:

$$\begin{aligned}
r1 &: A \longrightarrow AX & F_{r1} &= 10^{-3} \\
r2 &: X \longrightarrow Y & F_{r2} &= 50 \\
r3 &: Y \longrightarrow X & F_{r3} &= 2.5 \cdot 10^{-5}\, q(X)^2 \\
r4 &: X \longrightarrow \lambda & F_{r4} &= 5 \\
r5 &: X \longrightarrow X & F_{r5} &= 1000 \\
r6 &: Y \longrightarrow Y & F_{r6} &= 1000 \\
r7 &: A \longrightarrow A & F_{r7} &= 1000.
\end{aligned}$$

Initial amounts for X and Y are respectively set to 1000 and 2000, while the constant feeding element A has an amount set to $5 \cdot 10^6$. The rate parameters have been taken from [16]. Moreover, rounding has not been used.

Two completely stochastic realizations are depicted in Figure 5, where the phase space is shown in the lower plots. Two hybrid realizations using different thresholds are depicted in Figure 6.

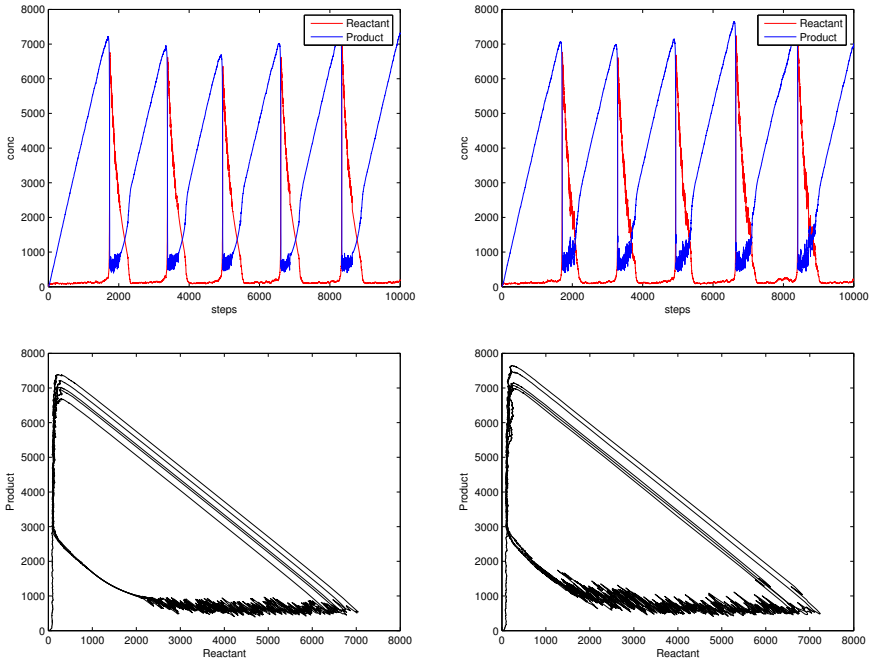

Fig. 6. Two hybrid simulation of the Brusselator ($\tau = 1000$ and $\tau = 3000$ respectively for the left hand part and right hand part of the figure). Corresponding phase plots are shown in the lower figures.

3.3 Lotka-Volterra

The Lotka-Volterra metabolic rewriting system is composed by the following set of rewriting rules and deterministic reaction maps [18]:

$$\begin{aligned} r1: \; X &\xrightarrow{k1} XX & F_{r1} &= 3 \cdot 10^{-3} \\ r2: XY &\xrightarrow{k2} YY & F_{r2} &= 4 \cdot 10^{-6} \cdot \max(X(k), Y(k)) \\ r3: \; Y &\xrightarrow{k3} \lambda & F_{r3} &= 3 \cdot 10^{-3}\, q(X). \end{aligned}$$

Moreover, the initial populations of both predators (Y) and preys (X) are set to 900 and, for each species, an inertia equal to 1 is also considered. Note that no rounding in the population dynamics is performed in this case. As usual, we can have completely deterministic dynamics [18] as well as a completely stochastic dynamics (see Figure 7). Hybrid behaviours can be obtained for intermediate values of τ (see Figure 8). An interesting case has arisen in a simulation using $\tau = 800$: in this simulation the randomness has led the system to rapidly fall toward the steady-state. Of course, this behaviour is not directly related to the choice of the threshold but, rather, to the random choices performed in the simulation.

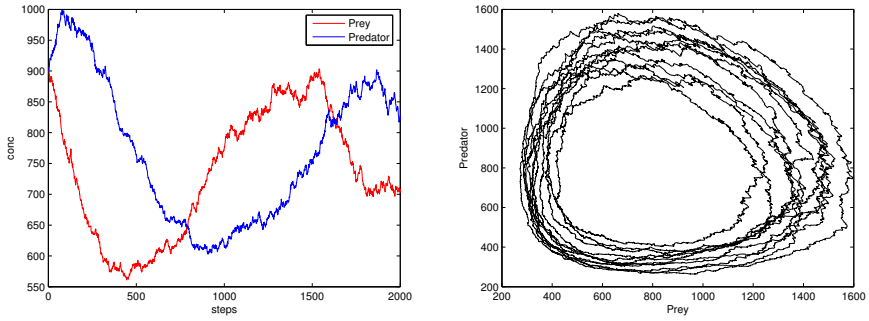

Fig. 7. Completely stochastic simulation of the LV system ($\tau = 10^{20}$), both evolution (left) and phase (right).

4 Conclusion

Although both deterministic and stochastic models for the simulation of biochemical systems have come to a good maturity, few has been done in the direction of hybrid algorithms. We have shown that this issue potentially leads to interesting dynamic representations, especially if coupled with the inherently versatile modeling formalism provided by MP systems.

Besides this, much still has to be done to make this strategy really competitive. Possible improvements in the short run may lead to a more suitable definition of the sigmoid function, and to a better tuned PRSG. In the medium and long run, alternative formalization of the h-MPA can be envisioned. In particular, a procedure accounting for two reaction maps for every rule, the former related to the deterministic, the latter to the stochastic behavior, may lead to rich realizations of a system evolution.

References

1. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J. of Computational Physics **22** (1976) 403
2. Turner, T.E., Schnell, S., Burrage, K.: Stochastic approaches for modelling in vivo reactions. Computational Biology and Chemistry **2004** (2004) 165–178
3. Cao, Y., Gillespie, D., Petzold, L.: Avoiding negative populations in explicit Poisson tau-leaping. J. of Chemical Physics **2005** (2005)
4. Haseltine, E.L., Rawlings, J.B.: Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. J. of Chemical Physics **117** (2002) 1357–1372
5. Salis, H., Kaznessis, Y.: Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. J. of Chemical Physics **122** (2005) 1–13

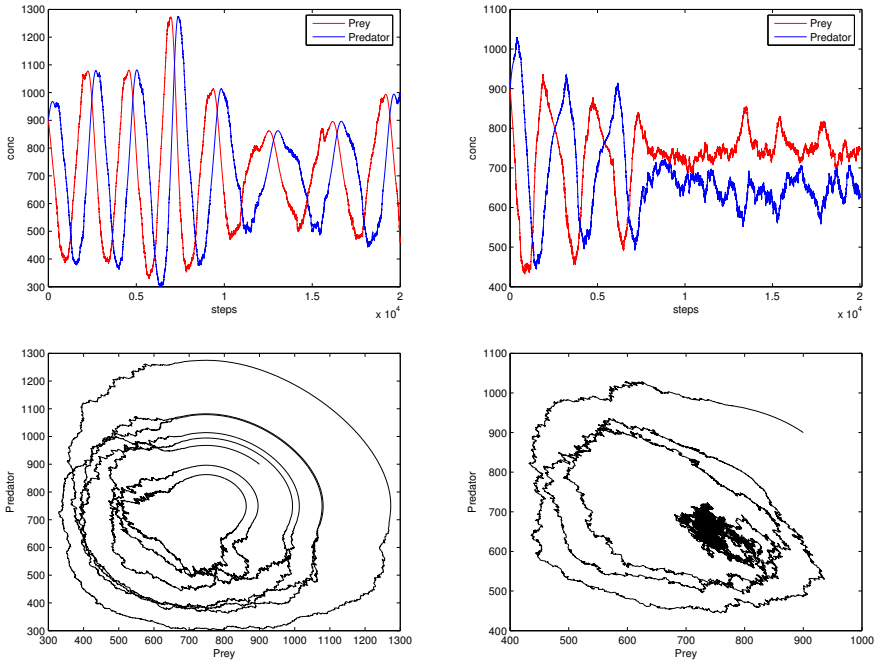

Fig. 8. Hybrid simulation of the LV system: $\tau = 700$ (left); $\tau = 800$ (right). Corresponding phase planes shown in the lower figures.

6. Zhang, X.: A hybrid algorithm for determining protein structure. IEEE Intelligent Systems **9** (1994) 66–74
7. Manca, V.: Topics and problems in metabolic P systems. In Păun, G., Pérez-Jiménez, M.J., eds.: Proc. of the Fourth Brainstorming Week on Membrane Computing (BWMC4), Sevilla, Spain, Fenix Editora (2006)
8. Bianco, L., Fontana, F., Manca, V.: P systems with reaction maps. International Journal of Foundations of Computer Science **17** (2006) 27–48
9. Păun, G.: Membrane Computing. An Introduction. Springer, Berlin (2002)
10. Păun, G., Rozenberg, G.: A guide to membrane computing. Theoretical Computer Science **287** (2002) 73–100
11. Manca, V., Bianco, L., Fontana, F.: Evolutions and oscillations of P systems: Applications to biological phenomena. In Mauri, G., Păun, G., Pérez-Jiménez, M.J., Rozenberg, G., Salomaa, A., eds.: Membrane Computing, 5th International Workshop, WMC 2004. Volume 3365 of Lecture Notes in Computer Science. Springer (2005) 63–84
12. Bianco, L., Fontana, F., Franco, G., Manca, V.: P systems for biological dynamics. In Ciobanu, G., Păun, G., Pérez-Jiménez, M.J., eds.: Applications of Membrane Computing. Springer (2006) 81–126
13. Ciobanu, G., Păun, G., Pérez-Jiménez, M.J., eds.: Applications of Membrane Computing. Springer, Berlin (2006)
14. Pescini, D., Besozzi, D., Mauri, G., Zandron, C.: Dynamical probabilistic P systems. International Journal of Foundations of Computer Science **17** (2006) 183–194

15. Pérez-Jiménez, M.J., Romero-Campero, F.J.: P systems: a new computational modelling tool for systems biology. Transactions in Computational Systems Biology (2006) In press.
16. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. of Physical Chemistry **81** (1977) 2340–2361
17. Illner, R., Bohun, C.S., McCollum, S., van Roode, T.: Mathematical Modelling. American Mathematical Society, Providence, RI (2005)
18. Bianco, L., Fontana, F., Manca, V.: Reaction-driven membrane systems. In Wang, L., Chen, K., Ong, Y.S., eds.: Advances in Natural Computation, First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part II. Volume 3611 of Lecture Notes in Computer Science., Springer (2005) 1155–1158