# Tau leaping stochastic simulation method in P systems*

Paolo Cazzaniga[a], Dario Pescini[a], Daniela Besozzi[b],
Giancarlo Mauri[a]

[a]Università degli Studi di Milano-Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione
Via Bicocca degli Arcimboldi 8, 20126 Milano, Italy
E-mail: cazzaniga/pescini/mauri@disco.unimib.it

[b]Università degli Studi di Milano
Dipartimento di Informatica e Comunicazione
Via Comelico 39, 20135 Milano, Italy
E-mail: besozzi@dico.unimi.it

**Abstract.** Stochastic simulations based on the tau leaping method are applicable to well stirred chemical reaction systems occurring inside a single fixed volume. In this paper we propose a novel method, based on the tau leaping procedure, for the simulation of complex systems composed by several communicating regions. The new method is here applied to dynamical probabilistic P systems, which are characterized by several features suitable to the purpose of performing stochastic simulations distributed in many membranes. Conclusive remarks and ideas for future research are finally presented.

## 1 Introduction

Stochastic modelling of biological systems is a topic of interest, since stochasticity and discreteness play an important role in cellular processes involving few molecules such as, e.g., signal transduction pathways, and the working of transcription or translation machinery ([11, 20] and references therein). Here we want to introduce a new stochastic approach in the framework of P systems [14], as a novel tool for the modelling of biological systems. In the following we will assume that the reader is familiar with basic notions on P systems; for further information we refer to the P systems Web Page:
http://psystems.disco.unimib.it.

The stochastic simulation algorithm (SSA, in short), introduced by Gillespie in [8], is currently used as the reference procedure for performing stochastic and discrete simulations of various biological systems (see, e.g., [1, 7, 12]). It is essentially an exact numerical simulation method that keeps track of every reaction event occurring in the modelled system, but has as a counterpart the

* Work supported by the Italian Ministry of University (MIUR), under project PRIN-04 “Systems Biology: modellazione, linguaggi e analisi (SYBILLA)”.

fact that the load of computer work is sometimes too high, and thus many realistic problems cannot be efficiently solved by using it.

To speed up the discrete stochastic simulation, Gillespie introduced in [9] the tau leaping method as an approximate simulation strategy. Using Poisson random numbers, it is possible to leap over many reaction events in a way that well approximates the exact stochastic simulation.

The SSA, as well as the tau leaping method, are applicable to well stirred chemical reaction systems contained inside a single fixed volume, at constant temperature.

In this paper, we introduce a new method in order to overcome this limit and to exploit the structure (formed by several volumes) and the communication of P systems, simulating both the behaviour of every volume (or membrane) and the behaviour of the whole system, using a modified tau leaping procedure.

We will refer to the *dynamical probabilistic P systems* (DPPs, in short) introduced in [17, 18]: they are membrane systems where probabilities are associated with the rules, and such values vary during the evolution of the system according to a prescribed strategy. The evolution of the system is achieved using a strategy similar to the SSA algorithm. More details about DPPs and examples of simulated systems can be found in [16–18].

The paper is structured as follows. In Section 2 we recall the state of the art of the stochastic simulations, the tau leaping procedure and we show some results in order to test the accuracy and efficiency of the tau leaping method. In Section 3 we introduce the new tau leaping procedure in the framework of DPPs and we present some results obtained by the simulations. We conclude with some remarks on future extensions of our work.

## 2 Gillespie's stochastic simulation methods

In this section we explain how the tau leaping selection procedure works, we present the pseudo-code of a possible implementation of the algorithm and we show some results in order to prove the accuracy and the efficiency of this method.

### 2.1 Tau leaping

The tau leaping method, first introduced by Gillespie in [9], is used to speed up stochastic simulations where one keeps track of every reaction event (as in SSA [8]), selecting a leap interval where more than one reaction can be fired.

Several improvements of the tau leaping have been proposed by Gillespie and Petzold [10] in order to improve the strategy of selecting the size of the tau leap. Tian and Burrage [19] and Chatterjee *et al.* [5] introduced a *binomial* tau leaping to avoid the possibility of producing negative populations. Also Cao *et al.* [2] modified the original tau leaping procedure to work out the negativity problem.

All these forms of tau leaping are lacking in two parts: first, they seem to violate the leap condition [9] since, during the leap, the estimated change of the propensity function is bound by a fraction $\epsilon$ (that is, a pre-specified error control parameter $0 < \epsilon \leq 1$) over the sum of all propensity functions. In this way, any propensity function that has a relatively small value will be allowed to change by a relatively large amount (the definitions of propensity function and leap condition will be given in the following). Second, the tau leaping selection requires the evaluation of $M^2$ quantities at each step, where $M$ is the number of reactions in the systems.

To avoid these problems, Gillespie *et al.* [3] introduced a new tau selection procedure. This procedure (to which we will refer in this paper) is more accurate than the previous ones since it satisfies more closely the leap condition, bounding in a uniform manner the relative changes in the propensity functions. Moreover, it is faster because the number of auxiliary quantities to compute increases linearly with the number of reactant species.

We consider a well stirred system in thermal equilibrium consisting of $N$ molecular species $\{S_1, \ldots, S_N\}$, which can interact through $M$ chemical reaction channels $\{R_1, \ldots, R_M\}$. The vector $\mathbf{X}(t) \equiv (X_1(t), \ldots, X_N(t))$, where $X_i(t)$ is the number of molecules of the species $S_i$ at time $t$, describes the state of the system at time $t$.

The probability that a reaction will occur in the next infinitesimal time interval $[t, t+dt)$ is given by $a_j(x)dt$, where $a_j$ is called the *propensity function* of the reaction $R_j$ in the state $\mathbf{X}(t) = x$ and is defined as $a_j = h_j c_j$, where $h_j$ is the number of distinct reactant molecules combinations and $c_j$ is the stochastic rate constant associated to the reaction $R_j$. The changes of the populations of the species are ruled by the *state change vector* $\mathbf{v}_j \equiv (v_{1j}, \ldots, v_{Nj})$, for $j = 1, \ldots, M$. The element $v_{ij}$ of $\mathbf{v}_j$ represents the change in the number of species $S_i$ due to reaction $R_j$.

The tau leaping procedure [3] tries to speed up the computation executing several reactions at each step of length $\tau$.

Given the state $\mathbf{X}(t) = x$ of the system, let $K_j(\tau, x, t)$ be the number of times that a reaction $R_j$ will fire in the time interval $[t, t+\tau)$ (where $j$ runs over all reaction channels).

For arbitrary values of $\tau$, it is difficult to compute the values of $K_j(\tau, x, t)$. On the contrary, if $\tau$ is small enough that the change in the state during $[t, t+\tau)$ will be so slight that no propensity function will suffer an appreciable change in its value (this is called *leap condition*), we obtain a good approximation to $K_j(\tau, x, t)$ using $P(a, \tau)$, which is the Poisson random variable with mean and variance $a\tau$.

So, starting from the state $\mathbf{X}(t) = x$ and choosing a value $\tau$ that satisfies the leap condition, we can update the state of the system at time $t + \tau$ according to:

$$\mathbf{X}(t+\tau) = x \; + \; \sum_{j=1}^{M} \mathbf{v}_j P_j(a_j(x), \tau) \tag{1}$$

where $P_j(a_j(x), \tau)$, for each $j = 1, \ldots, M$, denotes an independent sample of the Poisson random variable with mean $a_j(x)\tau$.

The procedure for the selection of tau is accomplished in order to bound the relative changes in the molecular populations, in such a way that the relative changes in the propensity functions will be all bounded, during the $\tau$ interval, by a small value $\epsilon$ $(0 \leq \epsilon \leq 1)$.

Let $\varDelta_\tau X_i$ be the change in the population $X_i$ in the time interval from $t$ to $t + \tau$, given the state $\mathbf{X}(t) = x$, the above requirement can be written as:

$$|\varDelta_\tau X_i| \ \leq \ max\{\epsilon_i x_i, 1\} \ \ \forall \ i \in I_{rs}, \tag{2}$$

where $I_{rs}$ denotes the set of indices of all reactant species.

The values $\epsilon_i = \epsilon_i(\epsilon, x_i)$ are chosen so that the relative changes in the propensity functions will be all bounded, at least approximately, by $\epsilon$. To do that, first determine for each $i \in I_{rs}$, the value of the higher order of reaction in which species $S_i$ appears as a reactant (denoted by $HOR(i)$). Then take

$$\epsilon_i = \frac{\epsilon}{g_i} \tag{3}$$

where $g_i = g_i(x_i)$ is defined as follows:

(i) if $HOR(i) = 1$ then $g_i = 1$;

(ii) if $HOR(i) = 2$ then $g_i = 2$
  - if any second-order reaction requires two $S_i$ molecules, then $g_i = \left(2 + \frac{1}{x_i - 1}\right)$

(iii) if $HOR(i) = 3$ then $g_i = 3$
  - if some third-order reaction requires two $S_i$ molecules, then $g_i = \frac{3}{2}\left(2 + \frac{1}{x_i - 1}\right)$
  - if some third-order reaction requires three $S_i$ molecules, then $g_i = \left(3 + \frac{1}{x_i - 1}\frac{2}{x_i - 2}\right)$

The procedure for computing the largest value of $\tau$ that satisfies condition (2) is the following.

Referring to the basic tau-leaping formula (1), it is possible to consider the quantity defined in (2) to be:

$$\varDelta_\tau X_i = \sum_{j \in J_{ncr}} v_{ij} P_j(a_j(x), \tau) \ \ \forall \ i \in I_{rs}, \tag{4}$$

where $J_{ncr}$ denotes the set of noncritical reactions. That is, a reaction channel with a positive propensity function that is currently within a small number of firings of exhausting one of its reactants is called *critical reaction*, the others reactions are named, instead, *noncritical reactions*. The motivations of this restriction can be found in [3].

As previously said, the Poisson random variables $P_j(a_j(x), \tau)$ on the right-hand side of the equation (4) are statistically independent and have means and variances $a_j\tau$, so the mean and variance of the linear combination can be computed as follows:

$$\langle \Delta_\tau X_i \rangle = \sum_{j \in J_{ncr}} v_{ij}[a_j(x)\tau], \qquad var\{\Delta_\tau X_i\} = \sum_{j \in J_{ncr}} v_{ij}^2[a_j(x)\tau] \tag{5}$$

for all $i \in I_{rs}$. So, following the same reasoning that it was used in the tau selection introduced in [10], it is possible to consider the bound (1) substantially satisfied if it is simultaneously satisfied by the absolute mean and the standard deviation of $\Delta_\tau X_i$:

$$|\Delta_\tau X_i| \leq max\{\epsilon_i x_i, 1\}, \qquad \sqrt{var\{\Delta_\tau X_i\}} \leq max\{\epsilon_i x_i, 1\}, \tag{6}$$

for all $i \in I_{rs}$.
Now, substituting formulas (5) into conditions (6) we obtain the following bounds on $\tau$:

$$\tau \leq \frac{max\{\epsilon_i x_i, 1\}}{|\sum_{j \in J_{ncr}} v_{ij} a_j(x)|}, \qquad \tau \leq \frac{max\{\epsilon_i x_i, 1\}^2}{\sum_{j \in J_{ncr}} v_{ij}^2 a_j(x)} \tag{7}$$

for all $i \in I_{rs}$.
Finally, the procedure to obtain $\tau$ is done by first computing the quantities:

$$\mu_i(x) = \sum_{j \in J_{ncr}} v_{ij} a_j(x), \quad \forall i \in I_{rs} \tag{8}$$

$$\sigma_i^2(x) = \sum_{j \in J_{ncr}} v_{ij}^2 a_j(x), \quad \forall i \in I_{rs}, \tag{9}$$

where we still have the restriction on the noncritical reactions $J_{ncr}$, due to the conditions of the modified non-negative Poisson tau-leaping [3], and then taking:

$$\tau = \min_{i \in I_{rs}} \left\{ \frac{max\{\epsilon x_i/g_i, 1\}}{|\mu_i(x)|}, \frac{max\{\epsilon x_i/g_i, 1\}^2}{\sigma_i^2(x)} \right\}, \tag{10}$$

where $g_i$ is obtained by the equation (3).

The formulas (8) and (9) are computed following the reasoning of [10], which shows that $\mu_j(x)\tau$ and $\sqrt{\sigma_j^2(x)\tau}$ respectively estimate the *mean* and the *standard deviation* of the expected change in $a_j(x)$ during $\tau$, and formula (10) requires that both of those quantities would be bounded to $\epsilon a_0(x)$ for $j = 1, \ldots, M$ satisfying the *leap condition*.

### 2.2 The algorithm

In this section, we introduce the algorithm used to compute the value of $\tau$ as described in [3].

Recalling here that we consider a systems with $N$ molecular species interacting through $M$ chemical reaction channels, the vector $\mathbf{X}(t) = x$ describes the state of the system and the dynamic is ruled by the state change vectors $\mathbf{v}_j$ (where $j$ runs over all reaction channels).

The algorithm works as follows:

1. Locate the set of all critical reactions;
2. Compute the quantities $\mu$ and $\sigma^2$;
3. Select the value of $\tau'$ as indicated in equation (10);
4. If $\tau' < n \cdot 1/a_0$ (where $n$ is usually set to 10), execute an SSA step as described in [8] and go to step 1, otherwise go to the next step;
5. Compute the sum of the propensity functions of all critical reactions $a_0^c(x)$;
6. Generate $\tau'' = 1/a_0^c(x) \cdot 1/rnd$, where $rnd$ is a random value from the uniform unit interval $(0, 1)$;
7. If $\tau' < \tau''$ then $\tau = \tau'$

    - For all critical reactions $R_j$ set the number of firings $k_j = 0$.
    - For all noncritical reactions $R_j$ generate $k_j$ as a sample of the Poisson random variable $P(a_j, \tau)$ with mean $a_j\tau$.

8. Else if $\tau'' < \tau'$ then $\tau = \tau''$

    - Select one critical reaction $R_j$ to be fired during this step and set $k_j = 1$, for all other critical reactions $R_j$ set $k_j = 0$.
    - For all noncritical reactions $R_j$ generate $k_j$ as a sample of the Poisson random variable $P(a_j, \tau)$ with mean $a_j\tau$.

9. Update the state of the system: $x(t + \tau) = x(t) + \sum_j k_j v_j$.

During step 1, the procedure identifies the set of critical reactions, which will be used in steps 5 and 6 to satisfy the requirements needed to avoid the negativity. In step 2 the quantities needed to obtain the largest value of $\tau'$ (step 3) that satisfies the leap condition are computed. If this value (step 4) is less than a multiple of $1/a_0$, then an SSA step is executed because, given the actual state of the system, it is more accurate than a tau-leap step.

Steps 5 and 6 generate a second candidate leap $\tau''$ that estimates the time of the next critical reaction.

If $\tau'$ is smaller than $\tau''$, than some noncritical reactions and no critical reactions will be executed during the leap. Otherwise, several noncritical reactions plus one critical reaction will be executed.

Step 9 updates the state of the system.

### 2.3 Results

In this section we present some results in order to show the accuracy and efficiency of the $\tau$ leaping procedure presented above. We have simulated a simple system of consecutive reactions:

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C \tag{11}$$

using the $\tau$ leaping method and the SSA algorithm. Figure 1 shows the behaviour of the system (11) simulated starting from a population of 1000 individuals of species $A$; the stochastic constants used for the simulation are $k_1 = 0.1$ and $k_2 = 0.025$. For the simulation with tau-leaping method, $\epsilon = 0.03$ was used.

Figure 2 shows the histogram plots of the distribution of $X_1(0.1)$, that is the number of individuals of species $A$ at time 0.1 $s$, obtained from $10^6$ runs of the SSA and $10^6$ runs of the tau leaping method with $\epsilon = 0.03$, simulating the system (11).

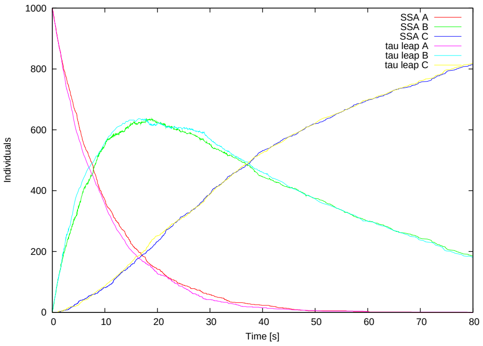

**Fig. 1.** Consecutive reactions system

The similar behaviour shown in Figure 1 and the distances between the SSA and the tau leap histograms of Figure 2 prove the accuracy of the tau leaping procedure. The efficiency is proved by the average number of steps of the simulations done, that is 102 using SSA and 79 with the tau leaping method.

## 3 $\tau$-DPPs

In this section we present a new tau leaping selection method in the framework of DPPs, the pseudo-code for the implementation of the procedure and we show some result to test the exactness of the procedure and of the communication between membranes.

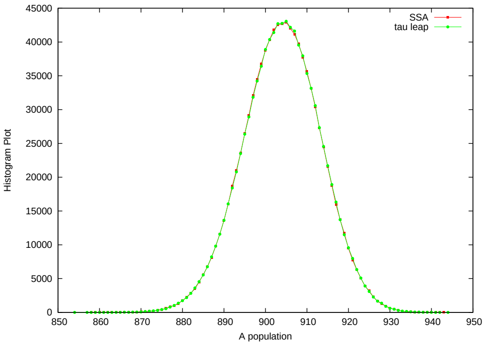

**Fig. 2.** Histogram plot of the distributions of species A

### 3.1 Tau leaping procedure in DPPs

The basic assumption of the tau leaping method introduced by Gillespie in [9], is that the analysed system is contained inside a single fixed volume. The aim of this work is to extend the tau leaping method to systems structured with different volumes. In the following we will refer without distinction to volumes or membranes.

P systems, first introduced by Paun in [14], have a membrane structure suitable to represent that kind of systems. Moreover, P systems manage the communication between membranes like biological systems and also this feature is suitable to our needs.

There are many classes of P systems [15], here we will refer to *dynamical probabilistic P systems* (DPP, in short) introduced by Pescini *et al.* in [18].

The major advantages of DPPs are the stochastic modelling feature and the possibility to probe different kind of parallelism; for instance, by introducing a bounded parallelism it is possible to execute some rules at each step, inside all membranes. This is exactly what happens executing the tau-leaping method for simulating biological systems. The main difference, as previously said, is that tau-leap algorithm works on a single volume whereas DPPs may simulate complex structured systems, where every membrane can have a different set of reaction channels.

Cazzaniga *et al.* introduced in [4] different stochastic approaches, using Gillespie SSA algorithm inside DPPs. The main problem arisen from that study is connected to the communication rules, because synchronization of all evolutive processes, in order to communicate objects between membranes, is forced.

Moreover, DPPs exploiting SSA select one rule in each membrane of the system and compute the time needed to execute it, considering only the internal state of the membrane where the rule will be executed. Therefore, different $\tau$ inside different membranes are obtained. For this reason, different time lines are simulated although one rule per step (in every membrane where there is something to evolve) is executed.

Introducing tau leaping method inside DPPs, we can work out these problems in three ways, because (1) we can choose the same leap of length $\tau$ for all the volumes, naturally synchronizing the membranes. Like DPPs evolving according to SSA algorithm, here we need to synchronize the processes running inside every membrane, because all of these have to evolve executing the same number of steps. The difference is that, when using SSA inside DPPs, the synchronization is forced at the end of each step, because all volumes generate different values of $\tau$, that is, after the same number of steps the time simulated within the membranes is different. On the contrary, with tau leaping method we execute the same number of iterations, synchronizing the processes at the end of each step, but the synchronization here is implicit, because at each iteration the same value of $\tau$ for all membranes is used.

Moreover, (2) we can communicate objects in the right way assuming that these are sent to the other volumes just at the end of each step, obtaining a good approximation of the real behaviour, because inside $\tau$ leap step the order of execution is not important.

Finally, (3) we can manage to keep track of the simulated time of the whole system because, as previously said, every membrane of the system evolves according to a common $\tau$ value, at each step.

The introduction of tau leaping method inside DPPs, requires a new procedure to select the value of $\tau$ inside all membranes of the system.

We recall here that the original tau leaping procedure can evolve, during each step, in three different manner: (i) like the SSA algorithm, executing one reaction during the leap, (ii) executing only noncritical reactions, or (iii) executing noncritical reactions and one critical reaction.

The $\tau$-DPP selection procedure has to consider how every membrane is evolving during the actual step, and then the smallest $\tau$ generated within the membranes is used to update the system.

For instance, if a membrane is in the condition of evolving by executing only non critical reactions, but the $\tau$ chosen inside that membrane is not the smallest one of the system, then, after receiving the minimal $\tau$, that membrane has to continue by sampling the rules to execute from the set of non critical reactions.

Finally, the procedure generates a local $\tau$ and then, if no membranes are evolving like SSA algorithm, the smallest tau ($\tau_{min}$) generated inside the volumes during the current step is chosen. Then the number of firings of the rules is sample as the Poisson random variable $P(a_j, \tau_{min})$ (where $j$ runs over all reaction channels).

If there is at least one membrane evolving in the SSA manner, which generates a value $\tau_{SSA}$, the procedure has to check if $\tau_{min} = \tau_{SSA}$.

This requirement is needed because if $\tau_{SSA}$ is greater than $\tau_{min}$, it is not possible to apply the rule selected inside that membrane, because the execution would be longer than the leap.

Otherwise, $\tau_{min} = \tau_{SSA}$ means that $\tau_{min}$ was generated by the membrane evolving according to the SSA algorithm and the execution of the selected rule is allowed.

### 3.2 The new algorithm

In this section we introduce the procedure to select the $\tau$ leap value between different membranes and how to execute local and communication rules.

The selection of local $\tau$ inside the membranes is done following the procedure presented in Section 2.2, the smallest $\tau$ of the system is then used to select the number of firings of the rules.

The new version of the algorithm, inside every volume, works as follows:

1. Locate the set of all critical reactions;
2. Compute the quantities $\mu$ and $\sigma^2$;
3. Select the value of $\tau'$ as indicated in equation (10);
4. If $\tau' < n \cdot 1/a_0$ (where $n$ is usually set to 10) then extract an SSA $\tau$ as described in [8], set $flag = 1$ and go to step 8, otherwise go to the next step;
5. Compute the sum of the propensity functions of all critical reactions $a_0^c(x)$;
6. Generate $\tau'' = 1/a_0^c(x) \cdot 1/rnd$, where $rnd$ is a random value from the uniform unit interval $[0, 1]$;
7. If $\tau' < \tau''$, then set $\tau = \tau'$ and $flag = 2$, else set $\tau = \tau''$ and $flag = 3$;
8. Receive the smallest tau of the system: $\tau_{min}$;
9. If $flag == 1$ and $\tau == \tau_{min}$, extract one rule to execute;
10. If $flag == 1$ and $\tau > \tau_{min}$, set $\tau = \tau - \tau_{min}$;
11. If $flag == 2$:
    - For all critical reactions $R_j$, set the number of firings $k_j = 0$.
    - For all noncritical reactions $R_j$, generate $k_j$ as a sample of the Poisson random variable $P(a_j, \tau_{min})$ with mean $a_j \tau_{min}$.
12. If $flag == 3$:
    - Select one critical reaction $R_j$ to be fired during this step and set $k_j = 1$, for all other critical reactions $R_j$ set $k_j = 0$.
    - For all noncritical reactions $R_j$, generate $k_j$ as a sample of the Poisson random variable $P(a_j, \tau_{min})$ with mean $a_j \tau_{min}$.
13. Send and Receive objects to and from other membranes (if communication rules were selected);
14. If $flag == 1$, $\tau > \tau_{min}$ and objects are received, go to the next step, otherwise go to step 8;
15. Update the state of the system: $x(t + \tau_{min}) = x(t) + \sum_j k_j v_j$.

The procedure begins like the pure tau leaping method, that is, the same $\tau$ selection is executed.

In this new version of the algorithm, a $flag$ is used during the iterations to remember how the rule selection has to proceed: $flag = 1$ means that the membrane is evolving according to the SSA algorithm, $flag = 2$ means that the membrane has to execute only non critical reactions, and $flag = 3$ means that, during the current step, the membrane will execute non critical reactions and one critical reaction.

After receiving the smallest $\tau$, during step 8, the procedure has to check how the membrane will evolve.

So, during the step 9 (the SSA-like evolution), if $flag = 1$ and the internal $\tau$ is the smallest of the system, a single rule is applied during the actual step. On the other hand (step 10), the value of the local $\tau$ is decreased by $\tau_{min}$ and no rule are executed, because during $\tau_{min}$ is not possible to completely execute the rule selected with the SSA algorithm that would spend $\tau$ to be executed.

If $flag = 2$ or $flag = 3$, the algorithm selects the rules to fire as a sample of the Poisson random variable $P(a_j, \tau_{min})$ with mean $a_j \tau_{min}$.

All the communication rules are applied during step 13, sending and receiving objects to and from other membranes.

Step 14 checks for eventually received object if the membrane is evolving in SSA manner but without executing any rule: if something is received from the outside, a new value of $\tau$ will be computed during the next iteration because, although no reactions will be executed, the state of the membrane changes due to the received objects. Otherwise the algorithm skips to step 8.

### 3.3 Results

To test the new algorithm presented in the previous section, we have implemented the consecutive reactions systems, shown in subsection 2.3, in the framework of DPPs. Here we want to model a system formed by two volumes, putting one rule in each volume. Labelling the membranes with 1 and 2, we put the rule $A \rightarrow (B, in_2)$ inside volume 1 and $B \rightarrow (C, in_1)$ inside volume 2. So, we can test the communication between membranes and check the new tau selection procedure.

Figure 3 shows the results of the tau leaping and the $\tau$-DPPs simulations that have similar behaviour.

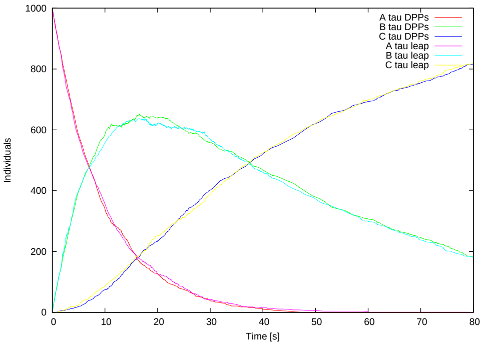

**Fig. 3.** Comparison between Gillespie's tau leaping and DPPs tau leaping

This benchmark shows that our algorithm is correct and reliable. We are now working on more complex models such as the Ras/cAMP/PKA signaling pathway in response to glucose addition and intracellular acidification in *Saccharomyces cerevisiae* [13], and to the Repressilator system [6].

# 4 Conclusions

In this work we have presented the stochastic method based on the tau-leaping procedure for simulating biological systems, implemented within the framework of P systems. In particular, the class of dynamical probabilistic P systems have been considered, to exploit the possibilities of modelling systems with a complex structure (that is, composed by several volumes) and of probing different levels of parallel rule application.

The new $\tau$ selection procedure here introduced works by selecting the smallest $\tau$ taken from the set of $\tau$-s generated by the membranes of the system during the current iteration; then, the evolution step is performed executing several rules that are selected following the procedure presented in Section 3.

The advantage of introducing tau leaping method inside DPPs is that we can choose the same leap of length $\tau$ for all the volumes, we can communicate objects in the right way (assuming that they are sent to the other volumes just at the end of each step, because the execution order does not matter), thus obtaining a good approximation of the system's behaviour. Finally, we can trace the simulated time of the whole system because every membrane evolves according to a common $\tau$ value. Moreover, the time needed to simulate biological systems using this new procedure is shorter than the time needed by the SSA. In Section 3.3 the communication and the new tau leap procedures are tested, comparing the behaviour of a simple system implemented both with the single volume model and with the multi-volume model.

The approach proposed in this paper opens several interesting research lines, directed to the modelling of real cellular processes or biological systems in general, as well as to the algorithmic improvements of the procedure, and the development of relevant (modelling and simulating) features in the area of Membrane Computing.

# References

1. A. Arkin, J. Ross, and H.H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. *Genetics*, 149:1633–1648, 1998.
2. Y. Cao, D. T. Gillespie, and L.R. Petzold. Avoiding negative populations in explicit Poisson tau-leaping. *Journ. Chem. Phys.*, 123:054104, 2005.
3. Y. Cao, D. T. Gillespie, and L.R. Petzold. Efficient step size selection for the tau-leaping simulation method. *Journ. Chem. Phys.*, 124:044109, 2006.
4. P. Cazzaniga, D. Pescini, F.J. Romero-Campero, D. Besozzi, and G. Mauri. Stochastic approaches in P systems for simulating biological systems. In M.A. Gutiérrez-Naranjo, G. Păun, A. Riscos-Núnez, and F.J. Romero-Campero, editors, *Proceedings of the Fourth Brainstorming Week on Membrane Computing Sevilla*, RGNC REPORT 02/2006, pages 145–164. Fénix Editora, 2006. Sevilla, Spain, January 30 - February 3.
5. A. Chatterjee, D.G. Vlachos, and M.A. Katsoulakis. Binomial distribution based tau-leap accelerated stochastic simulation. *Journ. Chem. Phys.*, 122:024112, 2005.

6. J. Garcia-Ojalvo, M. B. Elowitz, and S. H. Strogatz. Modeling a synthetic multicellular clock: Repressilators coupled by quorum sensing. *PNAS*, 101:10955–10960, 2004.
7. M. Gibbons and J. Bruck. Chemical Systems with Many Species and Many Channel. *Journ. Phys. Chem.*, 104:1876–1889, 2000.
8. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journ. Phys. Chem.*, 81:2340–2361, 1977.
9. D. T. Gillespie and L.R. Petzold. Approximate accelerated stochastic simulation of chemically reacting systems. *Journ. Chem. Phys.*, 115:1716–1733, 2001.
10. D. T. Gillespie and L.R. Petzold. Improved leap-size selection for accelerated stochastic simulation. *Journ. Chem. Phys.*, 119:8229–8234, 2003.
11. T.C. Meng, S. Somani, and P. Dhar. Modeling and simulation of biological systems with stochasticity. *In Silico Biology*, 4:0024, 2004.
12. C. J. Morton-Firth. *Stochastic Simulation of Cell signaling Pathways.* PhD thesis, University of Cambridge, Cambridge, UK, 1998.
13. D. Müller, S. Exler, L. Aguilera-Vázquez, E. Guerrero-Martín, and M. Reuss. Cyclic amp mediates the cell cycle dynamics of energy metabolism in Saccharomyces cerevisiae. *Yeast*, 20:351–367, 2003.
14. G. Păun. Computing with membranes. *J. Comput. Syst. Sci.*, 61:108–143, 2000.
15. G. Păun. *Membrane Computing. An Introduction.* Springer–Verlag, 2002. Berlin.
16. D. Pescini, D. Besozzi, and G. Mauri. Investigating local evolutions in dynamical probabilistic P systems. *Proceedings of SYNASC 2005–TAPS 2005, IEEE Computer Press, (2005)*, pages 440–447, 2005.
17. D. Pescini, D. Besozzi, G. Mauri, and C. Zandron. Analysis and simulation of dynamics in probabilistic P systems. In *Proceedings of DNA11 - 11th International Meeting on DNA Computing.*
18. D. Pescini, D. Besozzi, G. Mauri, and C. Zandron. Dynamical probabilistic P systems. *International Journal of Foundations of Computer Science*, 17:183–204, 2006.
19. T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *Journ. Chem. Phys.*, 121:10356–10364, 2004.
20. T.E. Turner, S. Schnell, and K. Burrage. Stochastic approaches for modelling in vivo reactions. *Computational Biology and Chemistry*, 28:165–178, 2004.