

Delay Stochastic Simulation of Biological Systems: A Purely Delayed Approach

Roberto Barbuti, Giulio Caravagna, Andrea Maggiolo-Schettini, and
Paolo Milazzo

Dipartimento di Informatica, Università di Pisa
Largo Pontecorvo 3, 56127 Pisa, Italy.
{barbuti,caravagn,maggiolo,milazzo}@di.unipi.it

Abstract. Delays in biological systems may be used to model events for which the underlying dynamics cannot be precisely observed. Mathematical modeling of biological systems with delays is usually based on Delay Differential Equations (DDEs), a kind of differential equations in which the derivative of the unknown function at a certain time is given in terms of the values of the function at previous times. In the literature, delay stochastic simulation algorithms have been proposed. These algorithms follow a “delay as duration” approach, which is not suitable for biological systems in which species involved in a delayed interaction can be involved at the same time in other interactions. We show on a DDE model of tumor growth that the delay as duration approach for stochastic simulation is not precise, and we propose a simulation algorithm based on a “purely delayed” interpretation of delays which provides better results on the considered model. Moreover, we give a formal definition of a stochastic simulation algorithm which combines both the delay as duration approach and the purely delayed approach.

1 Introduction

Biological systems can often be modeled at different abstraction levels. A simple event in a model that describes the system at a certain level of detail may correspond to a rather complex network of events in a lower level description. The choice of the abstraction level of a model usually depends on the knowledge of the system and on the efficiency of the analysis tools to be applied to the model.

Delays may appear in models of biological systems at any abstraction level, and are associated with events whose underlying dynamics either cannot be precisely observed or is too complex to be handled efficiently by analysis tools. Roughly, a delay may represent the time necessary for the underlying network of events to produce some result observable in the higher level model.

Mathematical modeling of biological systems with delays is mainly based on delay differential equations (DDEs), a kind of differential equations in which the derivative of the unknown function at a certain time is given in terms of the values of the function at previous times. In particular, this framework is very

general and allows both simple (constant) and complex (variable or distributed) forms of delays to be modeled.

As examples of DDE models of biological systems we mention [3,15,10,14,7]. In [3,15] an epidemiological model is defined that computes the theoretical number of people infected with a contagious illness in a closed population over time; in the model a delay is used to model the length of the infectious period. In [10] a simple predator-prey model with harvesting and time delays is presented; in the model a constant delay is used based on the assumption that the change rate of predators depends on the number of prey and predators at some previous time. Finally, models of tumor growth [14] and of HIV cellular infection [7] have been presented and analyzed by using DDEs.

Models based on DDEs, similarly to their simplest versions based on ordinary differential equations (ODEs), may be studied either analytically (by finding the solution of the equations, equilibria and bifurcation points) or via approximated numerical solutions. However, for complex real models analytical solutions are often difficult or impossible to be computed, whereas their approximated numerical solution is more feasible.

Models based on differential equations, although very useful when dealing with biological systems involving a huge number of components, are not suitable to model systems in which the quantity of some species is small. This is caused by the fact that differential equations represent discrete quantities with continuous variables, and when quantities are close to zero this becomes a too imprecise approximation. In these cases a more precise description of systems behavior can be obtained with stochastic models, where quantities are discrete and stochastic occurrence of events is taken into account.

The most common analysis technique for stochastic models is stochastic simulation that, in the case of models of biological systems without delays, often exploits Gillespie's Stochastic Simulation Algorithm (SSA) of chemical reactions [9], or one of its approximated variants [8,6]. In recent years, the interest for stochastic delayed processes increased [13]. In [2] a Delay Stochastic Simulation Algorithm (DSSA) has been proposed, this algorithm gives an interpretation as durations to delays. The delay associated with a chemical reaction whose reactants are consumed (i.e. are not also products) is interpreted as the duration of the reaction itself. Such an interpretation implies that the products of a chemical reaction with a delay are added to the state of the simulation not at the same time of reactants removal, but after a quantity of time corresponding to the delay. Hence, reactants cannot be involved in other reactions during the time modeled by the delay.

We argue that the interpretation of delay as duration is not always suitable for biological systems. We propose a simple variant of the DSSA in which reactants removal and products insertion are performed together after the delay. This corresponds to a different interpretation of delays, that is the delay is seen as the time needed for preparing an event which happens at the end of the delay. An example of a biological behavior which can be suitably modeled by this interpretation is mitosis. Cell mitosis is characterized by a pre-mitotic phase and

by a mitotic phase (cell division). The pre-mitotic phase prepares the division of the cell, when a cell undergoes the mitotic process, the pre-mitotic phase can be seen as a delay before the real cell division. During the pre-mitotic phase the cell can continue to interact with the environment, for example it can die. The DSSA in [2] cannot model this interactions because the reactants (in this case the cell itself) are removed at the beginning of reaction and the products are added at its end (that is after the delay).

In this paper we start by recalling the definition of DDEs and a DDE model of tumor growth [14]. Then, we give a stochastic model of the considered tumor growth example and simulate it by using the DSSA introduced in [2] and based on an interpretation of delays as durations. Subsequently, we propose a new “purely delayed” interpretation of delays and, consequently, a new variant of the DSSA that we apply to the considered tumor growth example. Although this new DSSA permits to have more precise results than the DSSA in [2] and it has a very easy implementation, there exist some scenarios in which this version of the algorithm does not work properly. Hence, we define a more precise version of it which requires a much more complex implementation. Before drawing our conclusions, we give a formal definition of a stochastic simulation algorithm which combines both the delay as duration approach and the purely delayed approach in its most precise definition.

2 Delay Differential Equations (DDEs)

The mathematical modeling of biological systems is often based on Ordinary Differential Equations (ODEs) describing the dynamics of the considered systems in terms of variation of the quantities of the involved species over time.

Whenever phenomena presenting a delayed effect are described by differential equations, we move from ODEs to *Delay Differential Equations* (DDEs). In DDEs the derivatives at current time depend on some past states of the system. The general form of a DDE for $X(t) \in \mathbb{R}^n$ is

$$\frac{dX}{dt} = f_x(t, X(t), \{X(t') : t' \leq t\}),$$

The simplest form of DDE considers *constant delays*, namely consists of equations of the form

$$\frac{dX}{dt} = f_x(t, X(t), X(t - \sigma_1), \dots, X(t - \sigma_n))$$

with $\sigma_1 > \dots > \sigma_n \geq 0$ and $\sigma_i \in \mathbb{R}$. This form of DDE allows models to describe events having a fixed duration. They have been used to describe biological systems in which events have a non-negligible duration [3,15,10] or in which a sequence of simple events is abstracted as a single complex event associated with a duration [14,7].

In what follows we recall an example of DDE model of a biological system that we shall use to compare delay stochastic simulation approaches.

2.1 A DDE model of tumor growth

Villasana and Radunskaya proposed in [14] a DDE model of tumor growth that includes the immune system response and a phase-specific drug able to alter the natural course of action of the cell cycle of the tumor cells.

The cell cycle is a series of sequential events leading to cell replication via cell division. It consists of four phases: G_1 , S, G_2 and M. The first three phases (G_1 , S, G_2) are called interphase. In these phases, the main event which happens is the replication of DNA. In the last phase (M), called mitosis, the cell segregates the duplicated sets of chromosomes between daughter cells and then divides. The duration of the cell cycle depends on the type of cell (e.g a human normal cell takes approximately 24 hours to perform a cycle).

The model in [14] considers three populations of cells: the immune system, the population of tumor cells during cell cycle interphase, and the population of tumor cells during mitosis. A delay is used to model the duration of the interphase, hence the model includes a delayed event that is the passage of a tumor cell from the population of those in the interphase to the population of those in the mitotic phase. In the model the effect of a phase-specific drug, able to arrest tumor cells during the mitosis, is studied. Such a drug has a negative influence also on the survival of cells of the immune system.

In this paper we study a simplified version of the model (presented in subsection 4.1.2 of [14]), where the effects of the immune response and of the drug are not taken into account. The simplified model, which considers only tumor cells (both in pre-mitotic and mitotic phases), consists of the following DDEs:

$$\begin{aligned} \frac{dT_I}{dt} &= 2a_4T_M - d_2T_I - a_1T_I(t - \sigma) & T_I(t) &= \phi_0(t) \text{ for } t \in [-\sigma, 0] \\ \frac{dT_M}{dt} &= a_1T_I(t - \sigma) - d_3T_M - a_4T_M & T_M(t) &= \phi_1(t) \text{ for } t \in [-\sigma, 0] \end{aligned}$$

Function $T_I(t)$ denotes the population of tumor cells during interphase at time t , and function $T_M(t)$ denotes the tumor population during mitosis at time t . The terms d_2T_I and d_3T_M represent cell deaths, or apoptosis. The constants a_1 and a_4 represent the phase change rates from interphase to mitosis (a_1) and back (a_4). In the following we shall denote with d the rate at which mitotic cells disappear, namely $d = d_3 + a_4$.

We assume that cells reside in the interphase at least σ units of time; then the number of cells that enter mitosis at time t depends on the number of cells that entered the interphase at least σ units of time before. This is modeled by the terms $T_I(t - \sigma)$ in the DDEs. Note that each cell leaving the mitotic phase produces two new cells in the T_I population (term $2a_4T_M$). In the model the growth of the tumor cell population is obtained only through mitosis, and is given by the constants a_1 , a_4 , and σ which regulate the pace of cell division. The delay σ requires the values of T_I and T_M to be given also in the interval $[-\sigma, 0]$: such values are assumed to be constant in the considered interval, and hence equal to the values of T_I and T_M at time 0.

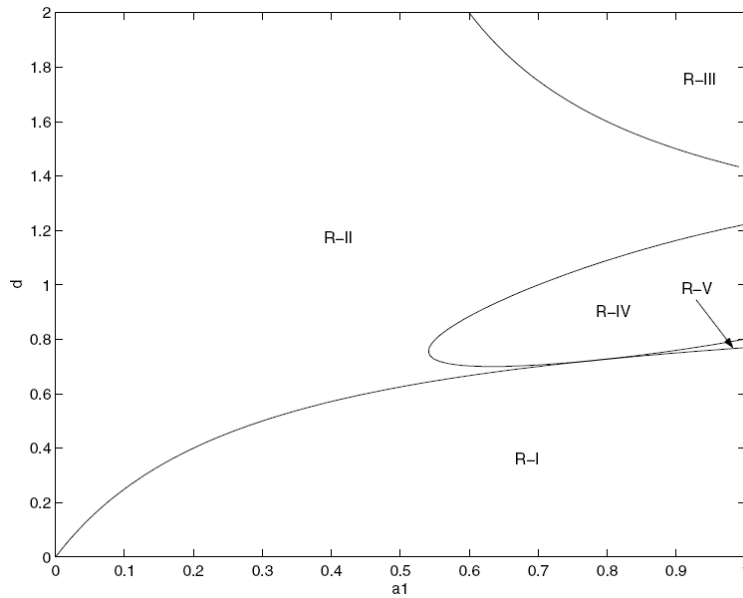


Fig. 1. The regions which describe the different behaviors of the DDE model by varying parameters a_1 and d (picture taken from [14]).

The analytic study of the DDEs constituting the model gives $(0, 0)$ as unique equilibrium. In Figure 1 (taken from [14]) some results are shown of the study of the model by varying a_1, d and σ and by setting the parameters a_4 and d_2 to 0.5 and 0.3, respectively. Figure 1 shows five regions.

When $\sigma = 0$, the region in which the tumor grows is R-I, while in the other regions the tumor decays.

When the delay is present ($\sigma > 0$), the growth region is essentially unaltered, but the decay is split into regions in which the tumor has different behaviors: in regions $R-II \cup R-IV$ the tumor still decays, but in regions $R-III \cup R-V$, when the value of σ is sufficiently large, the equilibrium becomes unstable. This is shown in Figures 2 and 3.

Figure 2 describes the behavior of the model, obtained by numerical solutions, inside the regions R-I, R-II, R-III, and R-IV, when $\sigma = 1$. Actually, we considered the point $(0.6, 0.6)$ in R-I, the point $(0.4, 1.0)$ in R-II, the point $(1.0, 1.8)$ in R-III, the point $(0.8, 0.8)$ in R-IV and an initial state consisting in 10^5 tumor cells in the interphase and 10^5 tumor cells in mitosis. We shall use always these parameters in the rest of the paper. In the figure, we can observe that, while the tumor grows in region R-I, it decays in all the other regions.

Figure 3 describes the behavior of the model when $\sigma = 10$. In regions R-I and R-IV the tumor has the same behavior as before. In region R-II it decays after some oscillations, while in region R-III it expresses an instability around the equilibrium. However, remark that values of T_M and T_I under 0 are not real-

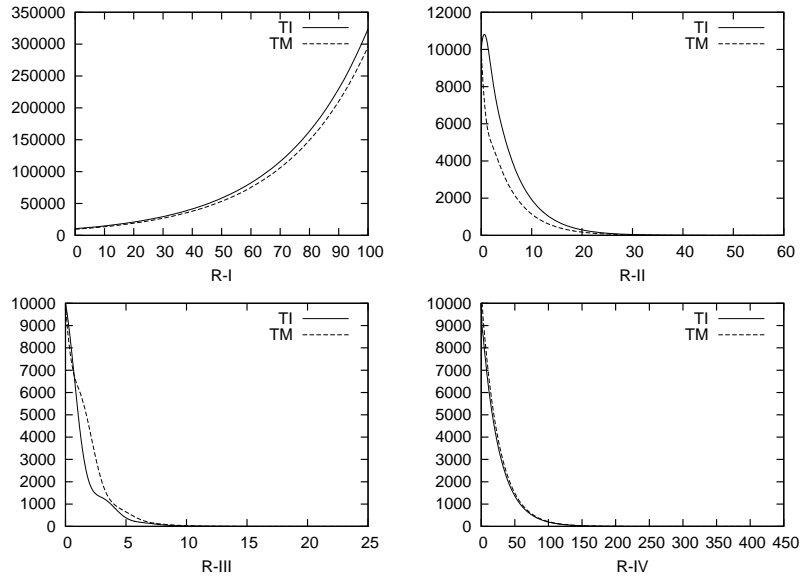


Fig. 2. Results of the numerical solution of the DDE model with $\sigma = 1$ for the regions described in Figure 1. On the x-axis time is given in *days* and on the y-axis is given the *number of cells*.

istic, and, as we will see in the following, they cannot be obtained by stochastic simulations.

3 Delay Stochastic Simulation

In this section we present algorithms for the stochastic simulation of biological systems with delays. Firstly, we introduce a well-known formulation of one of these algorithms, and we analyze the results of the simulations of the stochastic model equivalent to the one presented in the previous section. Secondly, we propose a variant of this algorithm and we compare the results of the simulations done by using this algorithm with those of the simulation done by using the original one.

All the simulations and the algorithms that we are going to present in this section have been implemented in the software tool `DelaySim`. This tool, available at <http://www.di.unipi.it/msvbio>, has been written in Java.

3.1 The Delay as Duration Approach (DDA)

In [2] Barrio *et al.* introduced a *Delay Stochastic Simulation Algorithm* (DSSA) by adding delays to Gillespie's Stochastic Simulation Algorithm (SSA) [9]. The algorithm has been used to explain more carefully than with DDE models the observed sustained oscillations in the expression levels of some proteins.

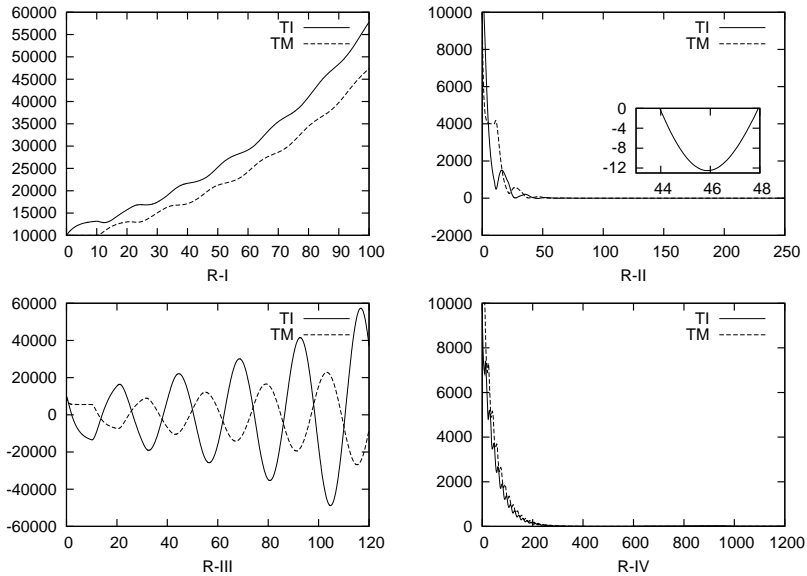


Fig. 3. Results of the approximated numerical simulation of the DDE model with $\sigma = 10$ for the regions described in Figure 1. On the x-axis time is given in *days* and on the y-axis is given the *number of cells*.

In order to recall the definition of the algorithm in [2] we consider a well-stirred system of *molecules* of N chemical *species* $\{S_1, \dots, S_N\}$ interacting through M chemical *reaction channels* $\mathcal{R} = R_1, \dots, R_M$. We assume the volume and the temperature of the system to be constant. We denote the number of molecules of species S_i in the system at time t with $X_i(t)$, and we want to study the evolution of the *state vector* $\mathbf{X}(t) = (X_1(t), \dots, X_N(t))$, by assuming that the system was initially in some state $\mathbf{X}(t_0) = \mathbf{x}_0$.

A reaction channel R_j is characterized mathematically by three quantities. The first is its *state-change vector* $\nu_j = (\nu_{1j}, \dots, \nu_{Nj})$, where ν_{ij} is defined to be the change in the S_i molecular population caused by one R_j reaction; let us denote each state-change vector ν_j as a the composition of the state-change vector for reactants, ν_j^r , and the state-change vector for products, ν_j^p , noting that $\nu_j = \nu_j^r + \nu_j^p$. For instance, given two species A and B , a reaction of the form $A \rightarrow B$ is described by the vector of reactants $(-1, 0)$, by the vector of products $(0, 1)$ and by the state-change vector $(-1, 1)$; differently, a reaction of the form $A \rightarrow A+B$ is described by the vector of reactants $(-1, 0)$, by the vector of products $(1, 1)$, and by the state-change vector $(0, 1)$.

The second characterizing quantity for a reaction channel R_j is its *propensity function* $a_j(\mathbf{x})$; this is defined, accordingly to [9], so that, given $\mathbf{X}(t) = \mathbf{x}$, $a_j(\mathbf{x})dt$ is the probability of reaction R_j to occur in state \mathbf{x} in the time interval $[t, t + dt]$.

As stated in [9], the propensity function can be defined as follows

$$a_j(\mathbf{x}) = k \cdot \prod_{i=1}^N \binom{X_i(t)}{|\nu_{i,j}^r|}$$

where $k \in \mathbb{R}$ denotes the kinetic function of reaction R_j and $|\nu_{i,j}^r|$ denotes the absolute value of the i -th coordinate of vector ν_j^r . This probabilistic definition finds its justification in physical theory.

The other characterizing quantity is a constant delay defined by a real number $\sigma \geq 0$. Following Barrio *et al.*, we classify reactions with delays into two categories: non-consuming reactions, where the reactants are also products (e.g. $A \rightarrow A + B$), and consuming reactions, where some of the reactants are consumed (e.g. $A \rightarrow B$). Throughout the paper, we denote the set of non-consuming reactions with delay by \mathcal{R}_{nc} , the set of consuming reactions with delay by \mathcal{R}_c , and the reactions without delays by \mathcal{R}_{nd} ; notice that $\mathcal{R} = \mathcal{R}_{nc} \cup \mathcal{R}_c \cup \mathcal{R}_{nd}$ and \mathcal{R}_{nc} , \mathcal{R}_c and \mathcal{R}_{nd} are pair-wise disjoint.

By adding delays to the SSA, Barrio *et al.* provide a method to model the firing of a reaction with delay based on the previously given classification. Formally, given a system in state $\mathbf{X}(t) = \mathbf{x}$, let us denote with τ the stochastic time quantity computed as in the SSA representing the putative time for next reaction to fire. Let us assume to choose to fire a non-consuming reaction with delay (a reaction from set \mathcal{R}_{nc}); then the reaction is scheduled at time $t + \sigma + \tau$, where σ is the delay of the reaction. Furthermore, the clock is increased to the value $t + \tau$ and the state does not change. On the contrary, if a consuming reaction with delay (a reaction from set \mathcal{R}_c) is chosen to fire, then its reactants are immediately removed from the state \mathbf{x} , the insertion of the products is scheduled at time $t + \sigma + \tau$, and, finally, the clock is increased to the value $t + \tau$. Reactions from set \mathcal{R}_{nd} (non-delayed reactions) are dealt with exactly as in the SSA. The DSSA by Barrio *et al.* is given in Figure 4.

We discuss now on the scheduling of the reactions with delay. When a non-consuming reaction is chosen, the algorithm does not change state, but simply schedules the firing of the reaction at time $t + \sigma_j + \tau$ (step (B2)). The reaction will complete its firing (reactants and products will be removed and inserted, respectively) when performing steps (A) and (A1).

Differently, as regards consuming reactions, the removal of the reactants is done at time instant t (step (B3)) preceding the time instant of insertion of the products (steps (A) and (A2)), namely the time at which the insertion is scheduled, $t + \sigma_j + \tau$. Notice that the removed reactants cannot have other interactions during the time interval $[t, t + \sigma_j + \tau)$.

As the reactants cannot have other interactions in the time quantity passing between the removal of the reactants and the insertion of the products, then this quantity can be seen as a duration needed for the reactants to exclusively complete the reaction. Since the approach of Barrio *at al.* gives this interpretation of delays we shall call it “delays as duration approach” (DDA).

As regards the handling of the scheduled events (step (A) of the algorithm), if in the time interval $[t; t + \tau)$ there are scheduled reactions, then τ is rejected and

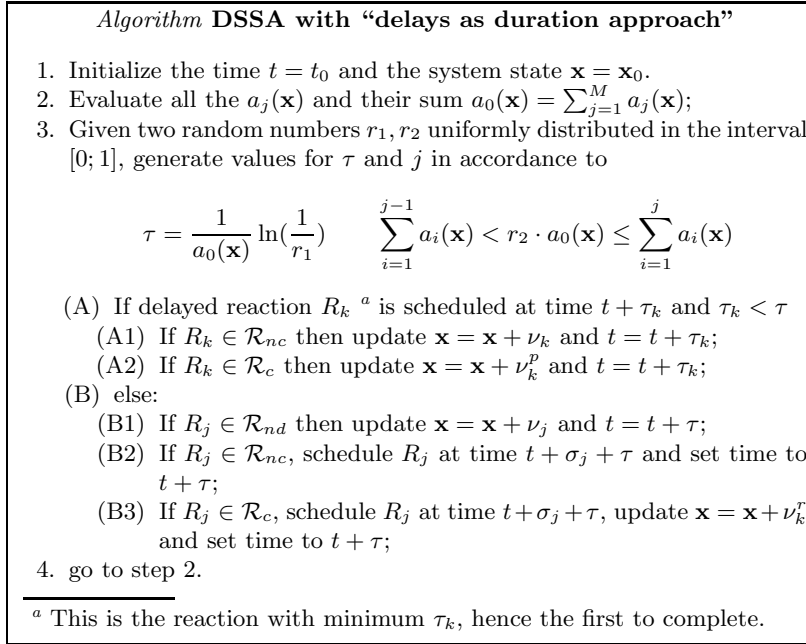
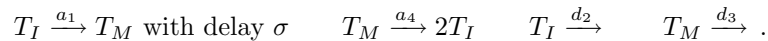


Fig. 4. The DSSA with “delays as duration approach” proposed in [2].

the scheduled reaction is handled. At this step the algorithm implicitly assumes R_k to be the scheduled action with the minimum τ_k . Among all the others which could be chosen the choice of the one with the minimum τ_k is quite intuitive since this will be the first to complete. Since generating random numbers is a costly operation, other authors defined variants of the DSSA that avoid rejecting τ in the handling of scheduled reactions [5,1]. However, the interpretation of the delays used to define these variants is the same as that of Barrio *et al.*

This interpretation of delays may not be precise for all biological systems. In particular, it may be not precise if in the biological system the reactants can have other interactions during the time window modeled by the delay. The tumor growth system we have recalled in Section 2.1 is an example of these systems. In fact, while tumor cells are involved in the phase change from interphase to mitosis (the delayed event) they can also die.

We applied the DSSA by Barrio *at al.* (we refer to the simulations done by applying this DSSA as DDA simulations) to a chemical reaction model corresponding to the DDE model of tumor growth recalled in Section 2.1. The reactions of the model are the following:



We have run 100 simulations for each considered parameter setting. The results of simulations with the same parameters as those considered in Figures 2 and 3

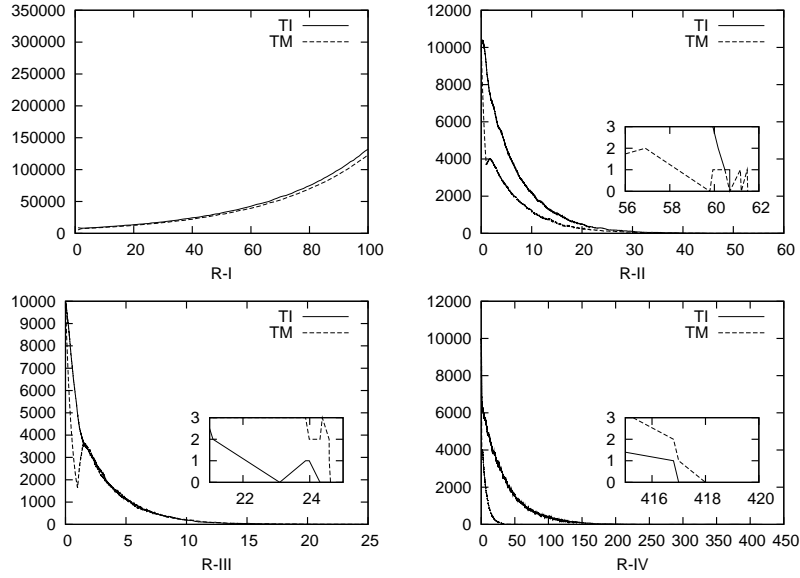


Fig. 5. DDA simulation of the stochastic model with $\sigma = 1$ for the regions described in Figure 1. On the x-axis time is given in *days* and on the y-axis is given the *number of cells*.

are shown in Figures 5 and 6, respectively. Actually, in the figures we show the result of one randomly chosen simulation run for each parameter setting.

Qualitatively, results obtained with DDA simulations are the same as those obtained with numerical simulation of the DDEs: we have exponential tumor growth in region R-I, tumor decay in the other regions and oscillations arise when the delay is increased. However, from the quantitative point of view we have that in the DDA simulations the growth in region R-I and the decay in the other regions are always slower than in the corresponding numerical simulation of the DDEs. In fact, with $\sigma = 1$ by the numerical simulation of the DDEs we have that in region R-I after 100 days both the quantities of tumor cells in interphase and in mitotic phase are around 300000, while in the result of DDA simulations they are around 130000. In the same conditions, but with $\sigma = 10$, in the numerical simulation of the DDEs we have about 47000 tumor cells in mitosis and 57000 tumor cells in interphase, while in the DDA simulations we have about 5000 and 5500 cells, respectively. As regards the other regions, in Table 1 the average tumor eradication times obtained with DDA simulations are compared with those obtained with numerical simulation of the DDEs (in this case with “eradication” we mean that the number of tumor cells of both kinds is under the value 1). Again, we have that in DDA simulations the dynamics is slower than in the numerical simulation of the DDEs. For instance, with $\sigma = 10$, in region R-IV the time needed for eradication in the DDEs is about 41% of the time needed in the DDA (440 against 1072), in region R-II the percentage

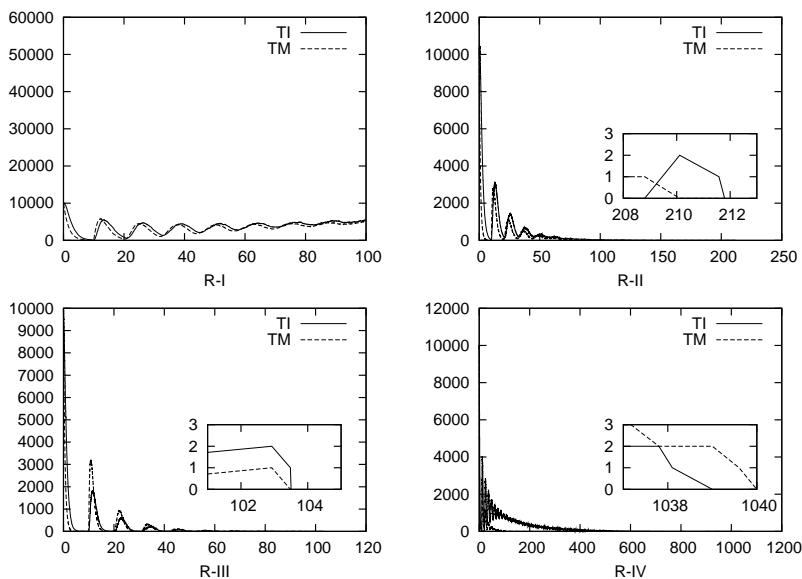


Fig. 6. DDA simulation of the stochastic model with $\sigma = 10$ for the regions described in Figure 1. On the x-axis time is given in *days* and on the y-axis is given the *number of cells*.

is smaller, 26% (59 against 224), and, in region R-III, it reaches 9% (12 against 126). For the same regions with $\sigma = 1$ these differences are smaller but not negligible.

3.2 A Purely Delayed Approach (PDA)

In this section we propose a variant of the DSSA based on a different interpretation of delays, namely a Stochastic Simulation Algorithm which follows a “purely delayed approach” (PDA). With this interpretation we try to overcome the fact that in the DDA the reactants cannot have other interactions. Furthermore, differently from Barrio *et al.*, we use the same interpretation of delays to define the method for firing both non-consuming and consuming reactions. This interpretation of delays was firstly implicitly adopted by Bratsun *et al.* in [4], to model a very simple example of protein degradation.

The approach we propose consists in firing a reaction completely when its associated scheduled events is handled, namely removing its reactants and inserting its products after the delay. The fact that we simply schedule delayed reactions without immediately removing their reactants motivates the terminology of “purely delayed”. Notice that non-consuming reactions are handled in the same way by DDA and PDA.

In this interpretation of delays it may happen that, when handling a scheduled reaction (again assuming to pick the one with the minimum τ_k as in the

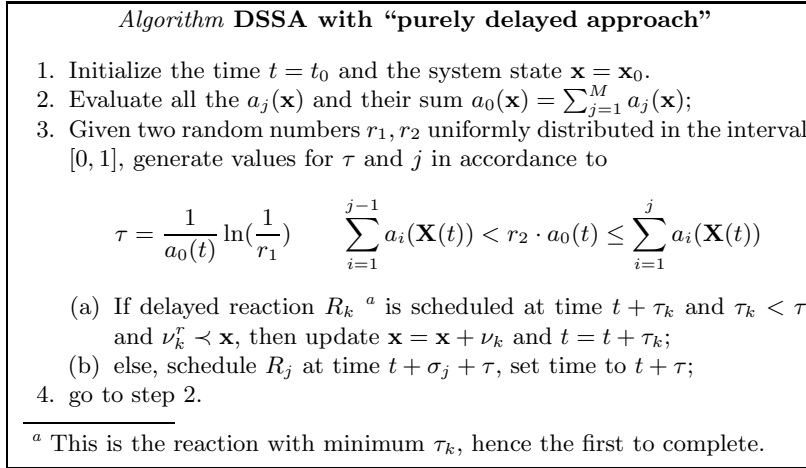


Fig. 7. The DSSA with “purely delayed approach”.

	DDEs	DDA Simulation	PDA Simulation
R-II with $\sigma = 1.0$	50	64	51
R-II with $\sigma = 10.0$	59	224	67
R-III with $\sigma = 1.0$	15	29	17
R-III with $\sigma = 10.0$	12	126	20
R-IV with $\sigma = 1.0$	238	302	214
R-IV with $\sigma = 10.0$	440	1072	248

Table 1. Average eradication times given in *days* for DDE model, DDA and PDA stochastic models. For the stochastic models the entries represent the sample of 100 simulations.

DDA since this is the first to complete), the reactants may not be present in the current state. In fact, they could have been destroyed or transformed by other interactions happened after the scheduling. In this case, the scheduled reaction has to be ignored. To formalize this, we know that a reaction R_j can be applied only if its reactants are all present in the current state of the simulation. Algebraically this corresponds to the fact that $\nu_j^r \prec \mathbf{x}$ where ν_j^r is the state-change vector of the reactants of reaction R_j , the system is described by \mathbf{x} and \prec is the ordering relation defined as $\forall i = 1, \dots, N. -\nu_{ij}^r \leq X_i(t)$. In order to verify that a scheduled reaction can effectively fire, it will be sufficient to check whether this condition holds. The formal definition of the DSSA with PDA is given in Figure 7.

As for the DDA, we have run 100 simulations of the stochastic model of tumor growth for each considered parameter setting. The results of simulations (we refer to these simulations as PDA simulations) with the same parameters as those considered in Figures 2 and 3 are shown in Figures 8 and 9, respectively.

Actually, in the figures we show the result of one randomly chosen simulation run for each parameter setting.

Qualitatively, results obtained with PDA simulations are the same as those obtained with numerical simulation of the DDEs (and with DDA simulations). From the quantitative point of view we have that in the PDA simulations the growth in region R-I with $\sigma = 1$ is almost equal to the corresponding numerical simulation of the DDEs (about 300000 tumor cells in both mitosis and interphase after 100 days, we recall that the DDA had reached values around 130000). On the contrary, with $\sigma = 10$, the difference between DDEs and PDA is higher: we have about 22000 tumor cells in interphase against 57000 for the DDEs and 5500 for the DDA, and 16000 tumor cells in mitosis against 47000 for the DDEs and 5000 for the DDA.

As regards the other regions, in Table 1 the average tumor eradication times obtained with PDA simulations are compared with those obtained with numerical simulation of the DDEs (again, in this case with “eradication” we mean that the number of tumor cells of both kinds is under the value 1). In PDA simulations the dynamics is generally slower than in the numerical simulation of the DDEs but it is faster than the DDA one. With $\sigma = 10$, in region R-IV the time needed for eradication in the PDA is smaller than the one in the DDEs (248 days against 440, DDA is 1072). In region R-II the values are: 67 days for the PDA and 59 days for the DDEs, DDA is 224. In region R-III values are: 20 days for the PDA, 12 days for the DDEs, and 126 days for DDA.

It is important to remark that differences between delay stochastic simulation results and numerical solutions of DDEs are also influenced by the initial conditions. The numerical solution of the DDEs assumes the initial population to be constant and greater than zero in the time interval $[-\sigma, 0]$. This allows delayed event to be enabled in the time interval $[0, \sigma]$. Both variants of the DSSA start to schedule delayed events from time 0, hence delayed reactions can fire only after the time σ . This results, when σ is great enough, in a behavior that is, in general, delayed with respect to that given by the DDEs.

Now, even if for this particular model this PDA definition is enough to justify the introduction of simulation techniques different from the DDA one, we must make some considerations about the PDA algorithm we proposed. In particular, there are some scenarios in which the algorithm does not work properly. We will go through these scenarios via some examples. For instance, consider a system described by the following initial state and chemical reaction:



where the initial state contains one single molecule A and the only reaction is the one transforming a molecule A in a molecule B with a kinetic constant k and a delay $\sigma > 0$. When applying the PDA algorithm to simulate this system it is easy to observe that the algorithm may over-schedule the firing of the reaction. This could happen because the reaction has a delay and, dependently on the value σ and on the random numbers generated by the algorithm, between time t_0 and the first firing of the scheduled reactions, the PDA may schedule an arbitrary amount

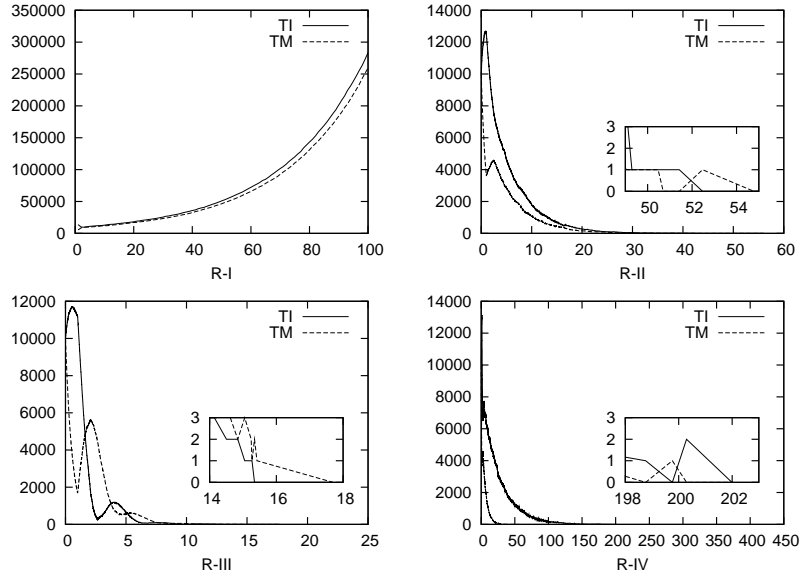


Fig. 8. PDA simulation of the stochastic model with $\sigma = 1$ for the regions described in Figure 1. On the x-axis time is given in *days* and on the y-axis is given the *number of cells*.

of times the reaction. However, after applying the first scheduled reaction at time $t + \varphi$, the system state becomes the vector $\mathbf{X}(t_0 + \varphi) = (0, 1)$ and, consequently, all the other scheduled reactions are not applicable anymore. This situation is not incorrect in fact, as expected, just one molecule A is transformed in a molecule B , but it is computationally unpleasant since it schedules a lot of reactions that will be never performed.

Also, it is possible to define, on top of this scenario, a new model such that the behavior of the PDA becomes incorrect. Imagine, for instance, to have the same scenario enriched with a reaction which produces, by an external unbounded source, molecules of type A . As there is no way of tracking the time since a molecule is in the system, then there is no way of preventing to apply a scheduled rule to a molecule A which is not in the current state of the system by, at least, σ time units. In this enriched scenario it may be the case that molecules A just appearing in the state by the firing of the new reaction, may be used to perform over-scheduled reactions and this is, obviously, incorrect.

Consequently, even if the PDA is, for some biological systems, a better candidate than the DDA algorithm by Barrio *et al.*, it needs to be properly tuned to avoid to simulate incorrect behaviors of the modeled system. The next section will be devoted to the definition of a variant of the PDA in order to face these issues. The more precise variant of the PDA will also be extended to obtain an algorithm that integrates the two approaches.

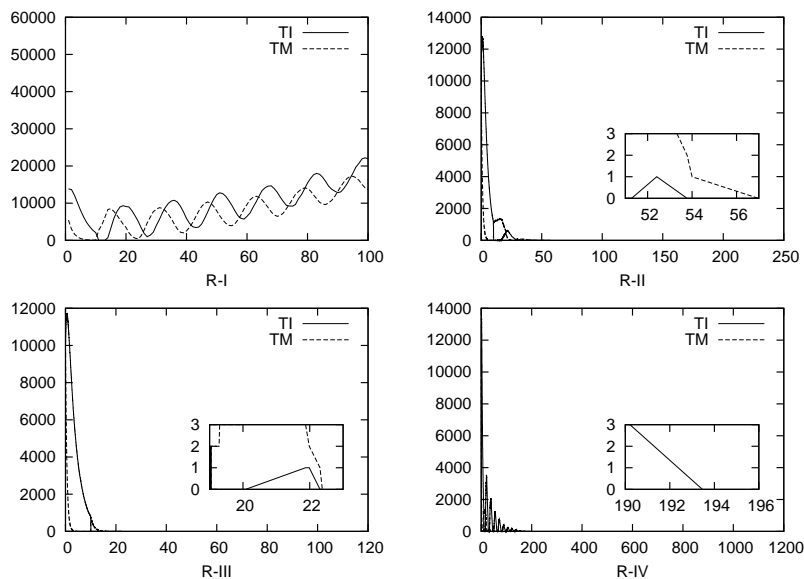


Fig. 9. PDA simulation of the stochastic model with $\sigma = 10$ for the regions described in Figure 1. On the x-axis time is given in *days* and on the y-axis is given the *number of cells*.

3.3 The PDA with markings

In order to get a correct version of the PDA we consider a solution based on a marking of molecules. This variant of the PDA, in the following named “marked Purely Delay Approach” (mPDA), is based on the idea of assigning, to each molecule of the system, a marking which permits the identification of the molecules involved in any scheduled reaction. On one side, this will fix the liabilities of the PDA approach but, as it is intuitive, it will be computationally much more expensive than the PDA.

In order to define the mPDA, we still assume the framework we used to introduce the PDA where the state vector $\mathbf{X}(t)$ describes the time evolution of a set of molecules belonging to n chemical species and \mathcal{R} denotes a set of chemical reaction channels.

In what follows we describe the mPDA, whose definition will be given in Figure 10.

Marking the molecules. The marking of molecules is based on the use of natural numbers as identifiers. In order to get a clear marking policy we classify the molecules of the system. Firstly, the molecules are classified in species, hence a system is described by a set of species $\mathcal{S} = \{\Sigma_1, \dots, \Sigma_n\}$ which defines the type of molecules we are considering. Furthermore, any Σ_i denotes a set of molecules

such that

$$\Sigma_i = \{S_{N_1}^{i,1}, \dots, S_{N_{n_i}}^{i,n_i}\}$$

where $S_N^{i,j}$ is a single molecule belonging to species $\Sigma_i \in \mathcal{S}$, with a unique identifier $j \in \mathbb{N}$ and concurrently performing the reactions in $N \subset \wp(\{z | R_z \in \mathcal{R}\})$, a set of identifiers of the reactions which are present in the current model. Notice that, for any molecule of the model, we carry much more information than the one we had in the PDA. In particular, for any molecule we can exactly know which reactions it is concurrently performing and, in this context, this means that there is an instance of reaction consuming that molecule which is currently scheduled in the event list. As these sets change during the simulation of a system, we may denote by $\mathcal{S}(t)$ and $\Sigma_i(t)$ the set of species and the set of molecules of species Σ_i at time t , respectively.

The marking of molecules requires to discuss the use of vector $\mathbf{X}(t)$ which, as in the PDA, will be used to observe the state changes due to the reactions firing. The construction of the state vector $\mathbf{X}(t)$ is slightly changed, with respect to the PDA, by the introduction of this marking notation. In particular, we define $\mathbf{X}(t)$ as

$$\mathbf{X}(t) = (|\Sigma_1(t)|, \dots, |\Sigma_n(t)|) \quad (1)$$

where $|\Sigma_i(t)|$ denotes the cardinality of the set $\Sigma_i(t)$. Notice that $|\Sigma_i(t)|$ represents the number of molecules of species Σ_i at time t , exactly as the element $X_i(t)$ of $\mathbf{X}(t)$ in the definition of the PDA.

As regards $\mathbf{X}(t_0)$, given an initial input state \mathbf{x}_0 for the mPDA, a proper initial marking for the system to simulate has to be computed. Let us assume $\mathbf{x}_0 = (X_1(t_0), \dots, X_n(t_0))$, the set of species can be defined as $\mathcal{S}(t_0) = \{\Sigma_1(t_0), \dots, \Sigma_n(t_0)\}$ where

$$\forall i = 1, \dots, n. \Sigma_i(t_0) = \{S_\emptyset^{i,1}, \dots, S_\emptyset^{i,X_i(t_0)}\}. \quad (2)$$

This construction creates n sets of species types $\Sigma_i(t_0)$ and, for each of them, it creates $X_i(t_0)$ molecules, each one with a different identifier, which are performing no reactions (at time t_0 the event list is empty, hence their set subscript is the empty set). This guarantees that the molecules are correctly marked and, hence, distinguishable.

In general, at any step of computation, the mPDA algorithm may modify some of the sets $\Sigma_i(t) \in \mathcal{S}(t)$. The discussion about how the mPDA modifies the sets in $\mathcal{S}(t)$, accordingly to the time-evolution of the simulated system, is presented in the forthcoming subsections.

Evaluating the propensity functions. The firing of a reaction is the result of a probabilistic choice based on the propensity function of the reaction, evaluated in the current state of the simulation. In order to explain how the mPDA works, we recall the assumption which is at the basis of the definition of this PDA

algorithm. The molecules can perform multiple reactions in parallel, but each molecule can be involved in each reaction at most once at a time. The first reaction to finish interrupts the others running in parallel and involving the same molecule. In order to avoid over-scheduling phenomena it is to be ensured that propensity function of a reaction depends only on the occurrences of reactants that are not yet involved in the same reaction.

Let us assume that we have to evaluate the propensity function of a reaction $R_z : M \xrightarrow{k, \sigma} P \in \mathcal{R}$ such that R_z transforms a multiset of molecules M in a multiset of molecules P with a kinetic constant $k \in \mathbb{R}$. More precisely, let us assume M to be a multiset of the form $\{(1, n_1), \dots, (w, n_w)\}$, namely reaction R_z transforms, for any $j = 1, \dots, w$, a number of n_j molecules of species Σ_j . Notice that this corresponds to a set representation of the state-change vector for reactants.

As we want to take into consideration only the molecules in the current state of the simulation which are not already involved in any scheduled firing of reaction R_z , then we have to filter those that are candidate for being used, if any. Let us denote by $[\Sigma_i(t), z]$ the set of identifiers of molecules belonging to species $\Sigma_i(t)$ which have to be considered in the evaluation of the propensity function of R_z , namely the set

$$[\Sigma_i(t), z] = \{j \mid S_N^{i,j} \in \Sigma_i(t) \wedge z \notin N\}.$$

Notice that this set is obtained by considering all the molecules in the current system, and by filtering them on the basis of the marking information that the mPDA stores in $\mathcal{S}(t)$. In the PDA this set could not have been defined.

Given $\mathbf{X}(t) = \mathbf{x}$, the propensity function $a_z(\mathbf{x})$ must consider only those molecules required by M which are not already performing reaction R_z , hence it can be defined as follows:

$$a_z(\mathbf{x}) = k \cdot \prod_{(i, n_i) \in M} \binom{||[\Sigma_i(t), z]||}{n_i} \quad (3)$$

where $||[\Sigma_i(t), z]||$ denotes the cardinality of the set $[\Sigma_i(t), z]$. This definition of the function $a_z(\mathbf{x})$ is such that mPDA propensity functions compute, in general, strictly smaller values than the PDA ones. Again, the PDA cannot distinguish the molecules which are performing a reaction from those which are not.

Scheduling a reaction to fire. Whenever the propensity functions have been evaluated, for any $R_z \in \mathcal{R}$, accordingly to the definition (3), the index of the reaction to fire can be chosen with the same policy used in the PDA. However, having a marking of molecules, the mPDA has a further level of choice to determine to which molecules the reaction will be applied.

To clarify this, as an example consider a system with two distinct molecules of the same type and both available for being consumed by a reaction. Whenever the mPDA decides to fire that reaction, it has to choose to which of the two molecules the reaction will be applied. This further choice is required by the

mPDA because it stores individual information about the molecules and, hence, there exist two different destination markings that the system may reach. Notice that, as the PDA abstracted these informations, it did not perform this further choice.

In order to define this further probabilistic choice, assume the mPDA has chosen to schedule the firing of the reaction $R_z : \{(1, n_1), \dots, (w, n_w)\} \xrightarrow{k, \sigma} P$ introduced in the previous section. The mPDA stores in the event list the same information of the PDA, namely the index of the reaction, z , and the time in which it will fire, some $t + \tau + \sigma$ if t is the current time, τ is the putative time for next reaction as computed in the PDA and σ is the delay of the reaction. Together with this information, the mPDA stores in each element of the event list a set of labels E representing the identifiers of the molecules which will be consumed by the reaction, when handled. The set E contains pairs of natural numbers and is such that if $(i, j) \in E$ then the molecule $S_N^{i,j} \in \Sigma_i(t)$ is involved in the reaction. The set E is built by considering the molecules which can effectively perform reaction R_z . Formally, for all $(i, n_i) \in M$, we choose n_i molecules from the set $[\Sigma_i, z]$. Each molecule is chosen with probability $|\Sigma_i(t, z)|^{-1}$, hence the probability of choosing a set E , with a system at time t , denoted by $P(E, t)$, is defined as

$$P(E, t) = \prod_{(i, n_i) \in M} \binom{|\Sigma_i(t, z)|}{n_i}^{-1} \quad (4)$$

The mPDA updates the system clock to a value $t + \tau$, stores the triple $(z, t + \tau + \sigma, E)$ in the event list and changes the marking of the molecules belonging to the set E . The marking is updated to store the information that the molecules in E are performing reaction R_z . This will guarantee that, when evaluating the propensity function for reaction R_z in the next time, the molecules in E will not be counted again, as expected. The updated set $\mathcal{S}(t + \tau)$, built by modification of the set $\mathcal{S}(t)$ satisfies the following proposition

$$\forall i = 1, \dots, n. \Sigma_i(t + \tau) = \{S_N^{i,j} \in \Sigma_i(t) \mid (i, j) \notin E\} \cup \{S_{N \cup \{z\}}^{i,j} \in \Sigma_i(t) \mid S_N^{i,j} \in \Sigma_i(t) \wedge (i, j) \in E\}. \quad (5)$$

Intuitively, any molecule in $\Sigma_i(t)$ that has not been assigned to the firing of reaction R_z is simply copied in $\Sigma_i(t + \tau)$. Differently, all the molecules assigned to this firing of R_z , are copied in $\Sigma_i(t + \tau)$ with the index z added to their set of concurrently running reactions.

Handling a scheduled reaction. When the mPDA decides, with the system at time t , to handle a scheduled reaction R_z it finds, as information in the event list, a triple (z, t', E) where z is the identifier of the reaction to fire, t' is the time to which the clock must be set and E is the set of identifiers of the molecules which will be consumed by the reaction. It is guaranteed, by construction, that the molecules denoted by the set E are still present in the current state of the

simulation. Hence, differently from the PDA, the condition $\nu_z^r \prec \mathbf{x}$ has not to be checked at this time.

The scheduled reaction is applied, as expected, by using the same policy of the PDA, namely the reactants are removed and the products are inserted. However, the mPDA must perform some additional operations to keep the marking of the molecules correct.

First of all, let us assume the set $E = \{(s_1, l_1), \dots, (s_m, l_m)\}$, then all the molecules denoted by these labels in $\mathcal{S}(t)$ must not be present anymore in the set $\mathcal{S}(t')$, built by the mPDA to represent the markings after the application of reaction R_k . In particular, for any $j = 1, \dots, m$, the molecule $S_N^{s_j, l_j}$ must be removed from the proper set in $\mathcal{S}(t')$. To define this, we start by defining the following sets

$$\forall i = 1, \dots, n. \Sigma_i(t') = \Sigma_i(t) \setminus \{S_N^{i,j} \in \Sigma_i(t) \mid (i,j) \in E\} \quad (6)$$

Notice that this corresponds to remove exactly the number of reactants required by the application of the reaction R_z . Consequently, given the state vector $\mathbf{X}(t) = \mathbf{x}$ defined accordingly to (1), this new marking corresponds to a new state $\mathbf{x} - \nu_z^r$.

As regards the interruption of the concurrently running reactions which were assuming to use the reactants just consumed by reaction R_z , the mPDA performs two operations. Firstly, the mPDA interrupts these reactions by removing them from the event list and, secondly, it unlocks all the involved partners molecules, so that they may start again, in the future, the interrupted reactions.

The interruption of the scheduled reactions is trivial. Let us denote with $\mathcal{E}(t)$ the event list of the system at time t , all the reactions to be interrupted are those which contain, at least, one reactant which is consumed by reaction R_z . We denote by $\mathcal{B}(t)$ the set of reactions to be interrupted at time t , namely the set

$$\mathcal{B}(t) = \{(\tilde{w}, \tilde{t}, \tilde{E}) \in \mathcal{E}(t) \mid \tilde{E} \cap E \neq \emptyset\}.$$

Consequently, the mPDA modifies the event list $\mathcal{E}(t)$ creating a new event list $\mathcal{E}(t')$ such that

$$\mathcal{E}(t') = \mathcal{E}(t) \setminus \mathcal{B}(t). \quad (7)$$

Unlocking the partners of the interrupted reactions is less easy. First of all, when considering a generic molecule $S_N^{i,j} \in \Sigma_i(t')$, where $\Sigma_i(t')$ is a set of molecules satisfying (6), it may be the case that it is coupled to some of the events which have been interrupted in $\mathcal{E}(t')$, and these events belong to the set $\mathcal{B}(t)$. Also, it may be the case that the molecule is performing other reactions which have not been interrupted. In general, even if $w \in \{w \mid (w, t, E) \in \mathcal{B}(t)\}$, this does not imply that all the scheduled events referring to reaction R_w have to be interrupted. Clearly, this depends on the one-to-many correspondence between a reaction and all the related scheduled events. Hence, in order to filter the reactions which have been really interrupted for a molecule $S_N^{i,j} \in \Sigma_i(t')$, we define the set

$$\mathcal{D}(t, i, j) = \{w \mid (w, \tilde{t}, \tilde{E} \cup \{(i, j)\}) \in \mathcal{B}(t)\}.$$

The construction of the set $\mathcal{D}(t, i, j)$ is straightforward. All the reactions which have to be interrupted, with respect to molecule $S_N^{i,j}$, are only those relative to events effectively interrupted and such that the molecule was assumed to be consumed by that instance of reaction. This constraint filters any possible collision between the indexes of the reactions relative to the interrupted events and those which are performed with partners whose are not affected by the application of the scheduled reaction R_z .

After this considerations, we can formally define how the interruption of some events affects the marking of the molecules by defining these new sets

$$\forall i = 1, \dots, n. \Sigma'_i(t') = \{S_{N'}^{i,j} \mid S_N^{i,j} \in \Sigma_i(t') \wedge N' = N \setminus \mathcal{D}(t, i, j)\}. \quad (8)$$

Notice that, as this definition does not modify the number of molecules present in the markings, then this marking, with respect to definition (1), still represents the vector state $\mathbf{x} - \nu_z^r$.

Finally, we discuss how the insertion of the products affects the marking of the molecules in the system, with respect to the sets just created. Let us assume that the scheduled reaction R_z creates a multiset of products $P = \{(1, n_1), \dots, (p, n_p)\}$, namely R_z produces, for any $j = 1, \dots, p$, a number n_j of new molecules of species Σ_j .

The creation of new objects to add to the sets $\Sigma'_i(t')$ requires to assign them new fresh identifiers respecting the uniqueness of the markings. As the marking is based on the use of natural numbers, the mPDA has an infinite set of numbers from which to choose the new identifiers. Let us denote, for a species Σ_i , the maximum among all the used identifiers appearing in $\Sigma'_i(t')$ as follows

$$\mu_i = \max\{j \mid S_N^{i,j} \in \Sigma'_i(t')\}.$$

Hence, for the set $\Sigma'_i(t')$, the creation of n_i non colliding identifiers can be obtained by choosing the n_i successors of the number μ_i . By these consideration we can define the following sets

$$\forall i = 1, \dots, n. \Sigma''_i(t') = \Sigma'_i(t') \cup \{S_\emptyset^{i,\mu_i+1}, \dots, S_\emptyset^{i,\mu_i+n_i} \mid (i, n_i) \in P\}. \quad (9)$$

Finally, the complete marking computed by the mPDA after the application of a scheduling rule is defined as

$$\mathcal{S}(t') = \{\Sigma''_1(t'), \dots, \Sigma''_n(t')\}. \quad (10)$$

This new marking is obtained by modifying the one representing, accordingly to definition (1), the state vector $\mathbf{x} - \nu_z^R$. As this marking is built by inserting, for each species, exactly the number of product molecules of reaction R_z , then this new marking corresponds to the state vector $\mathbf{x} - \nu_z^r + \nu_z^p = \mathbf{x} + \nu_z$ which is, as expected, the resulting state of the correct application of reaction R_z .

3.4 A DSSA combining the mPDA and the DDA

In this section we define a stochastic simulation algorithm which combines the delay as duration approach and the purely delayed approach in its most precise

Algorithm DSSA with “marked purely delayed approach”

1. Initialize the time $t = t_0$, build the initial marking w.r.t definition (2) by using the input initial state \mathbf{x}_0 .
2. Evaluate all the $a_j(\mathbf{x})$ w.r.t. definition (3), define $a_0(\mathbf{x}) = \sum_{j=1}^M a_j(\mathbf{x})$;
3. Given two random numbers r_1, r_2 uniformly distributed in the interval $[0, 1]$, generate values for τ and j in accordance to

$$\tau = \frac{1}{a_0(t)} \ln\left(\frac{1}{r_1}\right) \quad \sum_{i=1}^{j-1} a_i(\mathbf{x}) < r_2 \cdot a_0(t) \leq \sum_{i=1}^j a_i(\mathbf{x})$$

- (a) If delayed reaction R_k ^a is scheduled at time $t + \tau_k$ and $\tau_k < \tau$, then:
 - update the event list w.r.t. definition (7);
 - update the marking w.r.t definitions (6), (8), (9) and (10);
 - set time to $t = t + \tau_k$.
 - (b) else:
 - choose, w.r.t. definition (4), the set of reactants E that will be modified by reaction R_j ;
 - update the marking w.r.t. definition (5);
 - schedule the triple $(j, t + \sigma_j + \tau, E)$;
 - set time to $t + \tau$.
4. go to step 2.

^a This is the reaction with minimum τ_k , hence the first to complete.

Fig. 10. The DSSA with “marked purely delayed approach”.

definition. This will allow biological phenomena that cannot be suitably dealt with by only one of the two approaches, to be studied.

The framework in which we define this DSSA, in the following denoted as Full DSSA, is a simple modification of the one in which we defined the mPDA. This requires to redefine the DSSA with DDA in a framework where markings are present. As regards the notation, we introduce two disjoint sets of possible reactions $\mathcal{R} = \mathcal{R}_{\mathcal{D}} \cup \mathcal{R}_{\mathcal{P}}$ where $\mathcal{R}_{\mathcal{D}}$ and $\mathcal{R}_{\mathcal{P}}$ are the sets of reactions that are treated with a DDA approach and a mPDA approach, respectively.

In what follows we describe the Full DSSA, whose definition will be given in Figure 11.

Marking the molecules. Marking the molecules is necessary to use the mPDA inside the Full DSSA. Clearly, the marking defined by the mPDA in Section 3.3, together with definitions (1) and (2), is still valid in the Full DSSA.

Evaluating the propensity functions. We introduced two disjoint sets of reactions, $\mathcal{R}_{\mathcal{D}}$ and $\mathcal{R}_{\mathcal{P}}$, in order to separate reactions whose delays have to be considered as durations from those whose delays are pure. However, it is easy to

notice that, for any reaction in $R_z \in \mathcal{R}$, its propensity function can be correctly defined as in (3). This can be done because the different interpretations of the delays do not require different definitions of the propensity functions, but simply different semantics of the firings of reactions.

Despite this similarity, it is worth making a simple consideration about reactions in \mathcal{R}_D . Those reactions are such that, whenever started, they remove the reactants from the state of the simulation and, when the firing terminates, they add to the state their products. Hence, if a reaction of set \mathcal{R}_D is performed by a molecule $S_N^{i,j} \in \Sigma_i(t)$, then all the reactions concurrently running in N have to be interrupted and, the involved partners, have to be unlocked. By this consideration it is easy to notice that $\forall i \in N. R_i \in \mathcal{R}_P$ and, hence, it holds that $\forall t > t_0. \forall z \in \mathcal{R}_D. [\Sigma_i(t), z] = \Sigma_i(t)$. Summarizing, evaluating the propensity function of a reaction from set \mathcal{R}_D does not require to define the set $[\Sigma_i(t), z]$, an operation whose cost is at most linear in the size of $\Sigma_i(t)$, and, consequently, it is computationally less expensive than the evaluation of a propensity function of a reaction in \mathcal{R}_P .

Scheduling a reaction to fire. Reactions in \mathcal{R}_P are scheduled accordingly to the definitions (4) and (5) whereas, the reactions in the set \mathcal{R}_D are scheduled with a different policy.

Assume that the Full DSSA wants to schedule a reaction $R_w \in \mathcal{R}_D$ at time $t + \tau + \sigma_w$. Firstly, the Full DSSA must choose the reactants to which the reaction is applied. As this choice is independent with respect to the interpretation of the delays, the set E to which the reaction will be applied can be chosen accordingly to definition (4), as in the mPDA.

Now, as the state must be modified by the removal of the reactants, the Full DSSA changes the marking accordingly to definition (6) which corresponds exactly to this operation. Furthermore, as the Full DSSA has, for all molecules in the set E , to interrupt all the reactions that they are concurrently performing, it modifies the event list accordingly to definition (7). Finally, the Full DSSA further modifies the marking accordingly to definition (8) in order to unlock the partners involved in the interrupted reactions.

The scheduling of the reaction is then performed by adding, to the event list $\mathcal{E}(t)$, a pair $(w, t + \tau + \sigma_w)$. Consequently, the event list in the case of the Full DSSA contains some triples referring to scheduled reactions belonging to set \mathcal{R}_P , and some pairs referring to scheduled reactions belonging to set \mathcal{R}_D .

We remark that, in the mPDA, definitions (6), (7) and (8) were introduced when handling a scheduled reaction. The fact that for a reaction in \mathcal{R}_D the Full DSSA uses these definitions at the time of scheduling the reaction is due to the different interpretations of delays.

Handling a scheduled reaction. Scheduled reactions belonging to set \mathcal{R}_P are handled, accordingly to the mPDA, as explained in Section 3.3.

Differently, handling a reaction with a DDA approach is trivial because the major computational effort has been done when it was scheduled. Assume that

Algorithm Full DSSA

1. Initialize the time $t = t_0$, build the initial marking w.r.t definition (2) by using the input initial state \mathbf{x}_0 .
2. Evaluate all the $a_j(\mathbf{x})$ w.r.t. definition (3), define $a_0(\mathbf{x}) = \sum_{j=1}^M a_j(\mathbf{x})$;
3. Given two random numbers r_1, r_2 uniformly distributed in the interval $[0, 1]$, generate values for τ and j in accordance to

$$\tau = \frac{1}{a_0(t)} \ln\left(\frac{1}{r_1}\right) \quad \sum_{i=1}^{j-1} a_i(\mathbf{x}) < r_2 \cdot a_0(t) \leq \sum_{i=1}^j a_i(\mathbf{x})$$

- (a) If delayed reaction R_k ^a is scheduled at time $t + \tau_k$ and $\tau_k < \tau$, then:
 - if $R_k \in \mathcal{R}_{\mathcal{D}}$ update the marking w.r.t definitions (9) and (10);
 - if $R_k \in \mathcal{R}_{\mathcal{P}}$ update the event list w.r.t. definition (7), update the marking w.r.t definitions (6), (8), (9) and (10);
 - set time to $t = t + \tau_k$.
- (b) else:
 - choose, w.r.t. definition (4), the set of reactants E that will be modified by reaction R_j ;
 - if $R_j \in \mathcal{R}_{\mathcal{D}}$ update the event list w.r.t. definition (7), update the marking w.r.t. definitions (6) and (8), schedule the pair $(j, t + \sigma_j + \tau)$;
 - if $R_j \in \mathcal{R}_{\mathcal{P}}$ update the marking w.r.t. definition (5) and schedule the triple $(j, t + \sigma_j + \tau, E)$;
 - set time to $t + \tau$.
4. go to step 2.

^a This is the reaction with minimum τ_k , hence the first to complete.

Fig. 11. The Full DSSA with both “delay as duration approach” and “marked purely delayed approach”.

the Full DSSA wants to handle a scheduled reaction described by the pair (w, t') where $R_w \in \mathcal{R}_{\mathcal{D}}$. In order to insert the product molecules of R_w by modifying the current marking the Full DSSA modifies $\mathcal{S}(t)$ by applying definitions (9) and (10).

4 Discussion

In the previous sections we showed two different approaches to the firing of delayed reactions. The two approaches can be conveniently used for dealing with two different classes of delayed reactions. The delay as duration approach suitably deals with reactions in which reactants cannot participate, whenever scheduled, in other reactions. On the other hand, the purely delayed approach can be conveniently used in cases in which reactants can be involved in other reactions during the delay time.

In the example we have shown, cells in the interphase, which wait for entering the mitotic phase, can be involved in another reaction, namely their death. Thus in this example the purely delayed approach seems to be more appropriate for capturing the behavior of this real system.

However, the algorithm we have defined, implemented and applied to the considered cell-growth model is rather naive, and may be incorrect in several scenarios. Consequently, we have defined a more precise algorithm, the mPDA, based on the purely delayed approach, which exploits a technique of marking of the molecules. The marking technique makes the mPDA computationally costly. As future work, we plan to study simplified versions of the mPDA to be proved correct by means of abstract interpretation techniques.

Furthermore, as there are biological systems in which, due to the heterogeneity of reactions, both the approaches should be used, we combined, in a new framework, both the duration as delay approach and the purely delayed approach with markings.

In the future, we plan to define formal languages for the definition of models with delays. These languages should be such that the time evolution of the described models is in accordance with the algorithms we proposed. As far as we know, similar notions of delay have been presented in the framework of Petri nets with time information. In particular, in Timed nets [12] a notion of delay similar to a duration appears; differently, in Time nets [11] the notion of delay corresponds to our purely delayed approach.

References

1. D.F. Anderson (2007): *A Modified Next Reaction Method for Simulating Chemical Systems with Time Dependent Propensities and Delays*. *J. Ch. Phys.* 127(21), 214107.
2. M. Barrio, K. Burrage, A. Leier, T. Tian (2006) : *Oscillatory Regulation of Hes1: Discrete Stochastic Delay Modelling and Simulation*. *PLoS Computational Biology*, 2(9).
3. E. Beretta, T. Hara, W. Ma, Y. Takeuchi (2002): *Permanence of an SIR Epidemic Model with Distributed Time Delays*. *Tohoku Mathematical Journal* 54(2), 581–591.
4. D. Bratsun, D. Volfson, L.S. Tsimring, J. Hasty (2005): *Delay-induced Stochastic Oscillations in Gene Regulation*. *PNAS* 102(41), 14593–14598.
5. X. Cai (2007): *Exact Stochastic Simulation of Coupled Chemical Reactions with Delays*. *J. Ch. Phys.*, 126, 124108.
6. Y. Cao, D. Gillespie, L. Petzold (2005): *The Slow-scale Stochastic Simulation Algorithm*. *J. Ch. Phys.* 122, 014116.
7. R.V. Culshaw, S. Ruan (2000) : *A Delay-differential Equation Model of HIV Infection of CD4+ T-cells*. *Mathematical Biosciences* 165, 27–39.
8. D. Gillespie (2001): *Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems*. *J. Phys. Ch.* 115, 1716.
9. D. Gillespie: *Exact Stochastic Simulation of Coupled Chemical Reactions*. *J. Phys. Ch.* 81, 2340.
10. A. Martin, S. Ruan (2001): *Predator-prey Models with Delay and Prey Harvesting*. *J. Math. Biol.* 43(3), 247–267.

11. L. Popova–Zeugmann, M. Heiner, I. Koch (2005): *Time Petri Nets for Modelling and Analysis of Biochemical Networks*. *Fundamenta Informaticae* 67, 149–162.
12. C. Ramchandani (1974): *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*. *Massachusetts Inst. Technol. Res. Rep.*, MAC-TR 120.
13. R. Schlicht, S. Winkler (2008): *A Delay Stochastic Process with Applications in Molecular Biology*. *J. Math. Biol.* 57, 613–648.
14. M. Villasana, A. Radunskaya (2003): *A Delay Differential Equation Model for Tumor Growth*. *J. Math. Biol.* 47, 270–294.
15. F. Zhanga, Z. Lia, F. Zhangc (2008): *Global Stability of an SIR Epidemic Model with Constant Infectious Period*. *Applied Mathematics and Computation* 199(1), 285–291.