Generalized Continuous Time Bayesian Networks and their GSPN Semantics

Daniele Codetta-Raiteri, Luigi Portinale
Dipartimento di Informatica, Università del Piemonte Orientale "A. Avogadro"
{dcr, portinal}@di.unipmn.it

Abstract

We present an extension to Continuous Time Bayesian Networks (*CTBN*) called Generalized *CTBN* (*GCTBN*). The formalism allows one to model continuous time delayed variables (with exponentially distributed transition rates), as well as non delayed or "immediate" variables, which act as standard chance nodes in a Bayesian Network. The usefulness of this kind of model is discussed through an example concerning the reliability of a simple component-based system. The interpretation of *GCTBN* is proposed in terms of Generalized Stochastic Petri Nets (*GSPN*); the purpose is twofold: to provide a well-defined semantics for *GCTBN*in terms of the underlying stochastic process, and to provide an actual mean to perform inference (both prediction and smoothing) on *GCTBN*.

1 Introduction

The goal of this paper is to propose a generalization of Continuous Time Bayesian Networks (*CTBN*) (Nodelman et al., 2005) by allowing the presence of nodes with no explicit temporal evolution, called "immediate nodes". The resulting framework is called Generalized *CTBN* (*GCTBN*) and allows the modeling of processes having both a continuous-time temporal component and an immediate component capturing the logical/probabilistic interactions among modeled variables.

The possibilities offered by this generalization, can be exploited in several applications. For example, in system reliability analysis, it is very practical to distinguish between system components (having a temporal evolution) and specific modules or subsystems, whose behavior has to be modeled for the analysis. For instance, in Fault Tree Analysis (Dugan et al., 1992), basic events represent the system components with their failure rates, while non-basic events are logical gates identifying modules of the system under examination. In Dynamic Fault Trees (Dugan et al., 1992), logical gates identifying sub-modules, can be combined with dynamic gates, modeling time-dependent dependencies (usually assuming continuous time) among components or sub-modules. Also in this case, it is very important to distinguish, at the modeling level, between delayed and immediate entities. Of course, similar considerations apply in other tasks as well, as in medical diagnosis, financial forecasting, biological process modeling, etc.

The paper is organized as follows: Sec. 2 provides basic notions about the formalisms involved in the *GCTBN* definition and analysis; in Sec. 3, the *GCTBN* formalism is defined; in Sec. 4, a reliability case study is introduced, together with the corresponding *GCTBN* modeling; in Sec. 5, a semantic model, based on the formalism of Generalized Stochastic Petri Nets (*GSPN*) (Ajmone et al., 1995) is defined, and the corresponding model for the case study is discussed; in Sec. 6, we provide the algorithms to perform inference on a *GCTBN*, by means of analysis on the corresponding *GSPN*.

2 Preliminary notions

CTBN. Probabilistic graphical models for reasoning about processes that evolve over time, allow for a *factorization* (Lauritzen and Richardson, 2002) of the state space of the process, resulting in better modeling and inference features. Such models are usually based on graph structures, grounded on the theory of Bayesian Networks (*BN*). When time is taken into account, the main choice concerns whether to consider it as a discrete or a continuous dimension. In the second case, Continuous Time

Bayesian Networks (*CTBN*) have been firstly proposed in (Nodelman et al., 2002; Nodelman et al., 2005) and then refined in (Saria et al., 2007).

Standard inference tasks in temporal probabilistic models are *prediction* and *smoothing*. *Prediction* consists in computing the probability of a future state, given past evidence (a special case occurs when the last evidence time point and the query time are the same and is called *Filtering* or *Monitoring*). *Smoothing* is the task of estimating a past state, given all the evidence (observations) up to now. Such tasks can be accomplished by inference procedures usually based on specific adaptation of standard *BN* algorithms. In case of a *CTBN*, exact inference may often be impractical, so approximations through message-passing algorithms on cluster graphs (Nodelman et al., 2005; Saria et al., 2007), or through sampling (El-Hay et al., 2008; Fan and Shelton, 2008), have been proposed.

CTMC. A Continuous Time Markov Chain (Ajmone et al., 1995) enumerates the possible system states (nodes) and state transitions (arcs). A transition is not immediate, but may occur after a random period of time ruled by the negative exponential distribution according to the transition rate. Besides transition rates, a *CTMC* is characterized by the initial probability distribution of its states. There are two main analyses that can be performed with a *CTMC*: *steady state* and *transient* analysis. In the first case, the equilibrium distribution (at infinite time) of the states is computed, while in the second case, such a distribution is computed at a given time point.

GSPN are a particular form of Petri Nets, so they are composed by places, transitions and arcs (Fig. 2). A place can contain a discrete number of tokens (place *marking*), and the current state of the system is represented by the net marking given by the number of tokens in each place of the net. Transitions are used to model the system state transitions; a transition is enabled to fire when a certain net marking holds, and when the transition fires, a certain amount of tokens is moved from a set of places to another one, changing the net marking, so the system state.

Directed arcs are used to connect places to transitions and vice-versa, with the aim of moving tokens when transitions fire. In *GSPN*, inhibitor arcs are also present and connect a place to a transition with the aim of disabling the transition if the place is not empty. A cardinality can be associated with an arc in order to specify the number of tokens to be moved, in case of directed arcs, or the number of tokens necessary to disable the transition, in case of inhibitor arcs.

In *GSPN*, transitions can be immediate or timed. Immediate transitions fire as soon as they are enabled. In case of concurrent immediate transitions, their firing can be ruled by means of weights or priorities (π). Timed transitions fire if enabled, after a random period of time ruled by the negative exponential distribution according to the firing rate. Vanishing markings are those enabling immediate transitions; tangible markings are those enabling timed transitions.

The stochastic process associated with a *GSPN* is a homogeneous continuous time semi-Markov process (Ajmone et al., 1995) that can be analyzed by removing from the set of possible markings (states), the vanishing markings (since the system does not spend time in such states), and by analyzing the resulting *CTMC*. In this way, the analysis of a *GSPN* can provide several measures, and in particular the transient or steady state probability distribution of the number of tokens in each place.

A *GSPN* model can be edited and analyzed (or simulated) by means of *GreatSPN* (Chiola et al., 1995); in particular, this tool allows to set marking dependent firing rates. This means that the value of the firing rate of a timed transition, can change according to a set of conditions concerning the current marking of specific places. This possibility is exploited to simplify the generation of the *GSPN* model from the *GCTBN* (Portinale and Codetta, 2009). However, a timed transition characterized by a marking dependent firing rate, is equivalent to a set of timed transitions, each characterized by a certain constant firing rate and enabled by the corresponding condition about the marking of places. The two solutions determine the same underlying *CTMC*.

3 Generalized *CTBN*

Following (Nodelman et al., 2002), a *CTBN* is defined as follows:

Let $X = X_1, \dots X_n$ be a set of discrete variables, a *CTBN* over X consists of two components. The first one is an initial distribution P^0_X over X (possibly specified as a standard *BN* over X). The second component is a continuous-time transition model specified as (1) a directed graph G whose nodes are $X_1, \dots X_n$ (and with $Pa(X_i)$ denoting the parents of X_i in G); (2) a conditional intensity matrix (*CIM*) $Q_{X_i|Pa(X_i)}$ for every $X_i \in X$.

A *GCTBN* is defined as follows:
Given a set of discrete variables $X = \{X_1, \dots X_n\}$ partitioned into the sets D (delayed variables) and I (immediate variables) (i.e. $X = D \cup I$ and $D \cap I = \emptyset$), a *GCTBN* is a pair $N = \langle P^0_D, G \rangle$ where
-P^0_D is an initial probability distribution over D;
-G is a directed graph whose nodes are $X_1, \dots X_n$ (and with $Pa(X_i)$ denoting the parents of X_i in G) such that
1. there is no directed cycle in G composed only by nodes in the set I;
2. for each node $I_j \in I$ a conditional probability table (*CPT*) $P[I_j|Pa(I_j)]$ is defined (as in standard *BN*);
3. for each node $D_k \in D$ a *CIM* $Q_{D_k|Pa(D_k)}$ is defined (as in standard *CTBN*).

Delayed nodes are, as in case of a *CTBN*, nodes representing discrete variables with a continuous time evolution: the transition from a value to another one, is ruled by exponential transition rates defined in the *CIM* associated with the node. A delayed node is characterized also by the initial probability distribution of its possible values. So, a delayed node implicitly incorporates a *CTMC* (Sec. 2). If a delayed node is a root node (has no parent nodes) the transition rates are constant, otherwise the rates are conditioned by the values of the parent nodes. Such nodes, in the case of *GCTBN*, may be either delayed or immediate.

Immediate nodes are introduced in order to capture variables whose evolution is not ruled by transition rates associated with their values, but is conditionally and immediately determined, at a given time point, by the values of other variables in the model. Immediate nodes are then treated as usual chance nodes in a *BN* and have a standard *CPT* associated with them. In case of an immediate root node, its *CPT* actually specifies a prior probability distribution.

In a *GCTBN*, an immediate node I_j directly depends on its parent nodes ($Pa(I_j)$). However, if the set $Pa(I_j)$ contains immediate nodes, then the change of such nodes is ruled in turns by the change of their parents, eventually being delayed variables. So, what really determines a change in I_j is not $Pa(I_j)$, but instead the set of the "Closest" Delayed Ancestors of I_j ($CDA(I_j)$). Such set contains any delayed variable D_k such that a path from D_k to I_j exists and contains no intermediate delayed nodes.

The initial distribution P^0_D is specified only on delayed variables, since this is sufficient to obtain the joint initial distribution over the set X of all the variables of the *GCTBN* as follows: $P^0_X = P^0_D \prod_{I_j \in I} P[I_j|Pa(I_j)]$.

A few words are worth to be spent for the structure of the graph modeling the *GCTBN*. While it is in general possible to have cycles in the graph (as in *CTBN*) due to the temporal nature of some nodes, such cycles cannot be composed only by immediate nodes. Indeed, if this would be the case, we would introduce static circular dependencies among model variables.

The evolution of a system modeled through a *GCTBN* occurs as follows: the initial state is given by the assignment of the initial values of the variables, according to P^0_X (immediate root nodes, if any, keep their initial value during the model evolution). Given the current system state (represented by the joint assignment of the model variables, both delayed and immediate), a value transition of a delayed variable D_k will occur, after an exponentially distributed delay, by producing a new state called a "van-

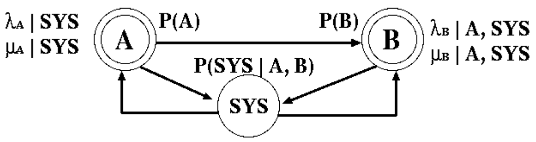

Figure 1: *GCTBN* model of the case study.

ishing state"; given the new vanishing state, a new assignment is determined to any immediate variable I_j such that D_k belongs to $CDA(I_j)$. The assignment to I_j is consistent with the *CPT* of I_j. The resulting state, called a "tangible state", is the new actual state of the system, from which the evolution can proceed, with a new transition of value, by a delayed variable. As we noticed in Sec. 2, the same state classification can be recognized in *GSPN*.

4 A motivating case study

To highlight usefulness and features of a *GCTBN* model, we now consider a simple case study in the field of reliability analysis. It consists of a small system composed by the main component A and its spare component B. Initially A is active while B is dormant; in case of failure of A, B is activated in order to replace A. However, the activation of B may fail with probability 0.01. If B fails before A, B can not replace A. The system is considered as failed if A is failed and B is dormant or failed. We suppose that only while the system is failed, the components A and B undergo repair. As soon as the repair of one of the components is completed, the component re-starts in working state and consequently the system becomes operative again; this implies that the repair of the other component is suspended.

The time to failure of the components is a random variable ruled by the negative exponential distribution: in the case of A, the failure rate is λ_A =1.0E-06 h^{-1}. The failure rate of B, λ_B, changes according to its current state: if B is dormant, λ_B is equal to 5.0E-07 h^{-1}; if instead B is active, λ_B is equal to 1.0E-06 h^{-1}. The time to repair a component is still ruled by the negative exponential distribution: A and B have the same repair rate $\mu_A = \mu_B = 0.01\ h^{-1}$.

a)

1 → 2		2 → 1	
SYS	λ_A	SYS	μ_A
1	1.0E-06 h^{-1}	1	0 h^{-1}
2	1.0E-06 h^{-1}	2	0.01 h^{-1}

b)

1 → 2			2 → 1		
A	SYS	λ_B	A	SYS	μ_B
1	1	5.0E-07 h^{-1}	1	1	0 h^{-1}
1	2	–	1	2	–
2	1	1.0E-06 h^{-1}	2	1	0 h^{-1}
2	2	5.0E-07 h^{-1}	2	2	0.01 h^{-1}

c)

A	B	SYS	Prob.	A	B	SYS	Prob.
1	1	1	1	2	1	1	0.99
1	1	2	0	2	1	2	0.01
1	2	1	1	2	2	1	0
1	2	2	0	2	2	2	1

Table 1: a) *CIM* of A. b) *CIM* of B. c) *CPT* of SYS.

The *GCTBN* model. The case study described above is represented by the *GCTBN* model in Fig. 1 where the variables A, B, SYS represent the state of the components and of the whole system respectively. All the variables are binary because each entity can be in the working state (1) or in the failed state (2); for the component B, the working state comprises both the dormancy and the activation.

The variable A influences the variable B because the failure rate of the component B depends on the state of A. Both the variables A and B influence SYS because the state of the whole system depends on the state of the components A and B. The arcs connecting the variable SYS to A and B respectively, concern the repair of the components A and B only while the system is failed.

A and B are delayed variables (Sec. 3) and implicitly incorporate a *CTMC* composed by two states: 1 and 2. Since both components are initially supposed to work, the initial probability distribution is set equal to 1 for states $A = 1$ and $B = 1$. In the *CIM* of A (Tab. 1.a), we can notice that the rate μ_A is not null only if the value of SYS is 2. The rate λ_A instead, is constant. In the *CIM* of B (Tab. 1.b), λ_B is increased only when A is equal to 2 and SYS is equal to 1 (this implies that B is active). As in the case of the variable A, the rate μ_B is not null only if the value of SYS is 2. The combination $A = 1, SYS = 2$ is impossible, so the corresponding entries are not significant.

The variable SYS is immediate (Sec. 3) and is characterized by the *CPT* appearing in Tab. 1.c. In particular, SYS is surely equal to 1 if A is equal to 1, and surely equal to 2 if both A and B are equal to 2. In the case of A equal to 2 and B equal to 1, SYS assumes the value 1 with probability 0.99 (this implies the activation of the spare component B), or the value 2 with probability 0.01 (this implies that B fails to activate).

The introduction of the immediate variable SYS is actually an important modeling feature, since it allows one to directly capture the static (or immediate) interactions between A and B, resulting in a probabilistic choice about the whole system status. Without the use of an immediate variable, it is hard to factorize the model using variables A and B (see (Portinale and Codetta, 2009) for an example, where an ordinary *CTBN* may fail in modeling all possible state transitions).

5 A Petri Net semantics for *GCTBN*

Combining in a single model entities explicitly evolving over time with entities whose determination is "immediate", has been already proposed in frameworks other than *CTBN*. In case of continuous time, a model having such features can be found in the framework of Petri Nets, namely *GSPN* (Sec. 2). A *GCTBN* model can be expressed in terms of a *GSPN*, by means of a set of translation rules (see (Portinale and Codetta, 2009) for details). This translation is twofold: (1) it provides a well-defined semantics for a *GCTBN* model, in terms of the underlying stochastic process it represents; (2) it provides an actual mean to perform inference on the *GCTBN* model, by exploiting well-studied analysis techniques for *GSPN*.

In fact, solution techniques for *GSPN* have received a lot of attention, especially with respect to the possibility of representing in a compact way the underlying *CTMC* and in solving it efficiently (Miner, 2007). Once a *GCTBN* has been compiled into a *GSPN*, such techniques can be employed to compute inference measures on the original *GCTBN* model (Sec. 6).

Case study. According to the conversion rules described in (Portinale and Codetta, 2009), the *GCTBN* of the case study in Fig. 1 can be converted into the *GSPN* model shown in Fig. 2 where the places A, B and SYS represent the variables of the *GCTBN* model. The value of a *GCTBN* variable is mapped into the marking (number of tokens) of the corresponding place in the *GSPN*. Let us consider the place B in the *GSPN*: the marking of the place B can be equal to 1 or 2, the same values that the variable B in the *GCTBN* can assume. B is a delayed variable and its initialization is modeled in the *GSPN* by the immediate transitions B_init_1 and B_init_2. Their effect is to set the initial marking of the place B to 1 or 2 respectively. Their weights correspond to the initial probability distribution of the variable B.

The change of the marking of the place B is determined by the timed transitions B_1_2 and B_2_1. The transition B_1_2 is enabled to fire when the place B contains one token; the firing sets the marking of B to 2. The transition B_2_1 instead, can fire when the marking of the place B is equal to 2, and turns it to 1.

The dependency of the transition rate of a variable on the values of the other variables in the *GCTBN* model, becomes in the *GSPN* model, the dependency of the firing rate of a timed transition on the markings of the other places. For instance, in the *GCTBN* model, the variable B depends on A and SYS; in the *GSPN* model, λ_B becomes the firing rate of the timed transition B_1_2, its value depends on the marking of the places A and SYS, and assumes the same values reported in Tab. 1.b. The firing rate of the timed transition B_2_1 instead, is μ_B reported in Tab. 1.b, still depending on the marking of the places A and SYS.

In the *GCTBN*, the variable SYS is immediate and depends on A and B. Therefore in the *GSPN*, each time the marking of the place A or B is modified, the marking of SYS has to be immediately updated: each time the transition A_1_2, A_2_1, B_1_2 or B_2_1 fires, one token appears in the place $emptySYS$; this determines the firing of the immediate transition $reset\_SYS\_1$ or $reset\_SYS\_2$ removing

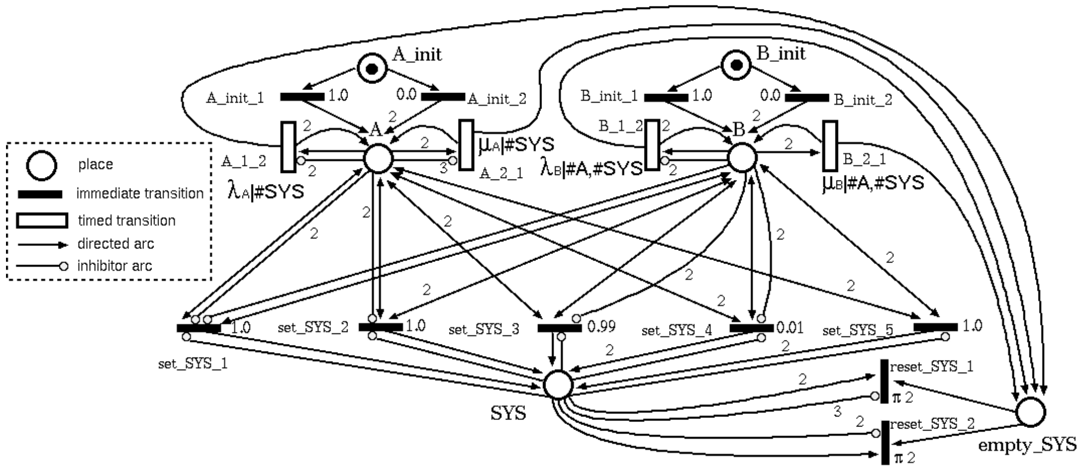

Figure 2: *GSPN* model obtained from the *GCTBN* in Fig. 1.

any token in SYS. Then, the marking of such place is set by one of the immediate transitions set_SYS_1, set_SYS_2, set_SYS_3, set_SYS_4, set_SYS_5. Each of them corresponds to one entry having not null probability in the *CPT* of Tab. 1.c.

Infinite rates. Another advantage of the *GSPN* semantics for *GCTBN* is that we can also model immediate changes on delayed variables; since delayed variables are characterized by changing rates (conditioned on their parents' values), this can be theoretically modeled by "infinite" rate values, but this is unmanageable with standard stochastic analysis. On the other hand, modeling the changes of delayed variables through transitions of a *GSPN* allows one to use immediate transitions to represent the above situation. A practical example can again be found in reliability applications, when the failure of a system component triggers the instantaneous failure of a dependent component (this is usually called a *functional dependency* (Dugan et al., 1992)). Fig. 3 is an example of a functional dependency between a component F (the trigger) and a component A: the failure of F immediately induces a failure of A. Fig. 3 shows the *GCTBN* model and the corresponding *GSPN*.

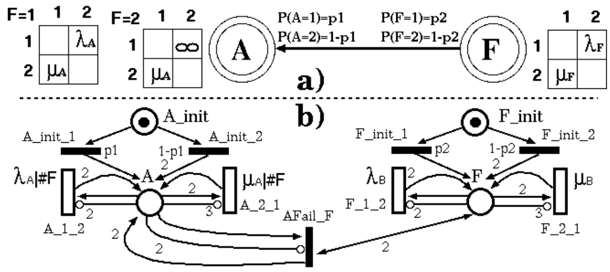

Figure 3: a) *GCTBN* modeling a functional dependency. b) The corresponding *GSPN* model.

6 Inference algorithms

In the present work, we take advantage of the correspondence between *GCTBN* and *GSPN*, in order to propose inference algorithms based on *GSPN* solution algorithms. For instance, computing the probability of a given *GCTBN* variable assignment $X = x_i$ at time t, will correspond to compute the probability of having i tokens at time t in the place modeling X in the *GSPN*.

Standard inference tasks are prediction and smoothing. The prediction task consists in computing $P(Q_t|e_{t_1}, \ldots, e_{t_k})$ which is the posterior probability at time t of a set of queried variables $Q \subseteq (D \cup I)$, given a stream of observations (evidence) $e_{t_1}, \ldots, e_{t_k}$ from time t_1 to time t_k with $t_1 < \ldots t_k < t$. Every evidence e_{t_j} consists of a

```
Procedure Prediction
INPUT: a set of queried variables Q, a query time t, a
set of temporally labeled evidences e_{t_1}, ... e_{t_k} with
t_1 < ... t_k < t
OUTPUT: P(Q_t | e_{t_1}, ... e_{t_k})
- let t_0 = 0;
for i = 1 to k {
- solve the GSPN transient at time (t_i - t_{i-1});
- compute from transient, p_i(j) = Pr{X_j | e_{t_i}} for X_j ∈ D ∪
R;
- update the weights of the immediate init transitions of
X_j according to p_i(j); }
- solve the GSPN transient at time (t - t_k);
- compute from transient, r = Pr{Q};
- output r;
```

Figure 4: The prediction inference procedure.

(possibly different) set of instantiated variables. Prediction can then be implemented by repeatedly solving the transient (Sec. 2) of the corresponding *GSPN* at the observation and query times. Of course, any observation will condition the evolution of the model, so the suitable conditioning operations must be performed before a new *GSPN* resolution.

The smoothing task consists in computing $P(Q_t|e_{t_1},\dots,e_{t_k})$ which is the probability at time t of a set of queried variables $Q \subseteq (D \cup I)$, given a stream of observations (evidence) $e_{t_1},\dots,e_{t_k}$ from time t_1 to time t_k with $t < t_1 < \dots t_k$. The issue is how to condition on variables observed at a time instant that follows the current one. The idea is then to try to reformulate the problem in such a way that it can be reduced to a prediction-like task. The approach is then based on the application of the Bayes rule as follows:
$P(Q_t|e_{t_1},\dots e_{t_k}) = \alpha P(Q_t)P(e_{t_1},\dots e_{t_k}|Q_t) =$
$= \alpha P(Q_t)P(e_{t_1}|Q_t)\dots P(e_{t_k}|e_{t_1},\dots e_{t_{k-1}},Q_t)$
In this way, every factor in the above formula is conditioned on the past and can be implemented as in prediction. However, this solution requires the computation of a normalization factor (α).

The pseudo-code for the prediction and smoothing procedure is shown in Fig. 4 and Fig. 5 respectively, and explained in details in (Portinale and Codetta, 2009).

```
Procedure Smoothing
INPUT: a set of queried variables Q, a query time t, a
set of temporally labeled evidences e_{t_1}, ... e_{t_k}
with t < t_1 < ... t_k
OUTPUT: P(Q_t | e_{t_1}, ... e_{t_k});
{ - Let N be the cardinality of possible assignments q_i(1 ≤
i ≤ N) of Q;
- A: array[N];
for i = 1 to N  A[i]=Smooth(q_i); //possibly in parallel
- output normalize(A); }

Procedure Smooth(q) {
- t_0 = t;
- solve the GSPN transient at time t;
- compute from transient, r = Pr{Q = q};
- ev = q;
for i = 1 to k {
- compute from transient, p_{i-1}(j) = Pr{X_j | ev} for X_j ∈ D ∪
R;
- update the weights of the immediate init transitions of
X_j according to p_{i-1}(j);
- solve the GSPN transient at time (t_i - t_{i-1});
- compute from transient, p_i(e) = Pr{e_{t_i}}
- r = r · p_i(e);
- ev = e_{t_i}; }
- output r; }
```

Figure 5: The smoothing inference procedure.

Case study. Consider again the case study of Fig. 1. We can easily compute the unreliability of the whole system, by asking for the probability P($SYS = 2$) over time. This reduces to compute the probability of having 2 tokens into place SYS, on the corresponding *GSPN*. This is done, by solving the transient (Sec. 2) at the required time instants. Results for our example are reported in Tab. 2. Since the modeled system is repairable, it makes sense to ask for the steady state distribution, in order to understand whether the system is reliable in the long run. By solving the *GSPN* for steady state (Sec. 2), we can indeed compute that the probabilities of component A and B being faulty in the long run are 0.496681 and 0.500026 respectively, while the probability of the whole system being faulty ($P(SYS = 2)$) is 0.000051, meaning that a good reliability is assured.

Concerning prediction, let us consider to observe the system working ($SYS = 1$) at time $t_1 = 10^5h$ and the system failed ($SYS = 2$) at time $t_2 = 2 \cdot 10^5h$. As described in details in (Portinale and Codetta, 2009), by applying the procedure outlined in Fig. 4, we can compute the probability of component A being working at time $t = 5 \cdot 10^5h$, conditioned by the observation stream. The result is 0.521855.

Concerning smoothing inference, let us suppose to have observed the system working at time $t_1 = 3\cdot10^5h$ and failed at time $t_2 = 5\cdot10^5h$. We ask for the probability of component A being failed at time $t = 2\cdot10^2h$, conditioned by the above evidence. By applying the procedure out-

Time (h)	Unreliability	Time (h)	Unreliability
200000	$1.4E-05$	400000	$2.3E-05$
300000	$1.9E-05$	500000	$2.7E-05$

Table 2: Unreliability results.

lined in Fig. 5, as described in details in (Portinale and Codetta, 2009), we obtain that the required probability is equal to 0.308548.

7 Conclusions and future works

The presented formalism of *GCTBN* allows one to mix in the same model continuous time delayed variables with standard "immediate" chance variables, as well as to model immediate changes on delayed variables. The usefulness of this kind of model has been discussed through some examples from reliability analysis.

The semantics of the proposed *GCTBN* formalism has been provided in terms of *GSPN*, a well-known formalism with well established analysis techniques. In particular, adopting *GSPN* solution algorithms as the basis for *GCTBN* inference, allows one to take advantage of specialized methodologies for solving the underlying stochastic process, that are currently able to deal with extremely large models; in particular, such techniques (based on symbolic data structures) allow for one order of magnitude of increase in the size of the models to be solved exactly, with respect to standard methods, meaning that models with an order of 10^{10} tangible states can actually be solved (Miner, 2007).

However, the analysis of a *GCTBN* by means of the underlying *GSPN* is only one possibility that does not take explicit advantage of the structure of the graph as in *CTBN* algorithms (Nodelman et al., 2005; Saria et al., 2007). Our future works will concentrate on the possibility of adopting cluster-based or stochastic simulation approximations, even on *GCTBN* models, and in comparing their performance and quality with respect to *GSPN*-based solution techniques. Finally, since symbolic representations have been proved very useful for the analysis of *GSPN* models, it would also be of significant interest to study the relationships between such representations and the inference procedures on probabilistic graphical models in general, since this could in principle open the possibility of new classes of algorithms for *BN*-based formalisms.

References

M. Ajmone, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. 1995. *Modelling with Generalized Stochastic Petri Nets.* J. Wiley.

G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaudo. 1995. GreatSPN 1.7: Graphical Editor and Analyzer for Timed and Stochastic Petri Nets. *Performance Evaluation*, 24(1&2):47–68.

J.B. Dugan, S.J. Bavuso, and M.A. Boyd. 1992. Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on Reliability*, 41:363–377.

T. El-Hay, N. Friedman, and R. Kupferman. 2008. Gibbs sampling in factorized continuous time Markov processes. In *Proc. 24rd UAI'08.*

Y. Fan and C. Shelton. 2008. Sampling for approximate inference in continuous time Bayesian networks. In *Proc. 10th Int. Symp. on AI and Mathematics.*

S.L. Lauritzen and T.S. Richardson. 2002. Chain graph models and their causal interpretations. *Journal Of The Royal Statistical Society Series B*, 64(3):321–348.

A.S. Miner. 2007. Decision diagrams for the exact solution of Markov models. *Proceedings in Applied Mathematics and Mechanics (PAMM)*, 7(1).

U. Nodelman, C.R. Shelton, and D. Koller. 2002. Continuous Time Bayesian Networks. In *Proc. 18th UAI'02*, pages 378–387.

U. Nodelman, C.R. Shelton, and D. Koller. 2005. Expectation propagation for continuous time Bayesian networks. In *Proc. 21st UAI'05*, pages 431–440.

L. Portinale and D. Codetta. 2009. A GSPN semantics for continuous time Bayesian networks with immediate nodes. Technical Report TR-INF-2009-03-03-UNIPMN, Dip. di Informatica, Univ. del Piemonte Orientale. http://www.di.unipmn.it.

S. Saria, U. Nodelman, and D. Koller. 2007. Reasoning at the right time granularity. In *Proc. 23rd UAI'07*, pages 421–430.