Acta Cybernetica **20** (2012) 347–384.

Improving the Construction of the DBM Over Approximation of the State Space of Real-time Preemptive Systems*

Abdelli Abdelkrim†

Abstract

We present in this paper an algorithm allowing an efficient computation of the tightest DBM over-approximation of the state space of preemptive systems modeled by using Time Petri Nets with inhibitor arcs. First of all, we propose an algorithm that reduces the effort of computing the tightest DBM over-approximated graph. For this effect, each class of this graph is expressed as a pair $(M, \widetilde{D})$, where M is a marking and $\widetilde{D}$ is the system of all DBM inequalities even the redundant ones. We thereby make it possible to compute the system $\widetilde{D}$ straightforwardly in its normal form, without requiring to compute the intermediary polyhedra. Hence, we succeed to remove the errors reported in the implementation of other DBM approximations. Then we show that by relaxing a bit in the precision of the DBM approximation, we can achieve to construct more compact graphs while reducing still more the cost of their computation. We provide for this abstraction a suitable equivalence relation that contract yet more the graphs. The experimental results comparing the defined constructions with other approaches are reported.

Keywords: Preemptive Systems, Time Petri Nets, Stopwatch Inhibitor arcs, State class graph, DBM over-approximation

1 Introduction

Nowadays, real-time systems are becoming more and more complex and are often critical. Therefore, their verification has to be performed thoroughly in order to prove the correctness of their behaviors. These systems consist of several tasks that are interacting and sharing one or more resources (e.g processors, memory). Hence, the problem is to determine, for instance, whether these actions can be scheduled in such a way that their constraints are satisfied.

Furthermore, the correctness proofs of such systems are demanding much theory regarding their increasing complexity. We may need, for instance, to consider formal

*This work was supported by the Algerian national Project PNR number 8/u160/3067
†LSI Laboratory- Computer Science Department- USTHB University-Algiers Algeria

DOI: 10.14232/actacyb.20.3.2012.1

models requiring the specification of time preemption; concept where execution of a task may be stopped for a while and later resumed at the same point. This notion of suspension requires to extend the semantics of timed clocks in order to handle such behaviors. For this effect, Cassez *et al* have introduced the *stopwatch* mechanism [10] and hence many models have been defined, as for instance, hybrid automata (LHA) [2] and stopwatch automata (SWA) [10]. Time Petri nets (TPN) have also been considered in several works including *Preemptive-TPN* [7], *Stopwatch-TPN* [5], *Inhibitor-TPN* [15], and *Scheduling-TPN* [13].

The verification of qualitative and quantitative properties of such a system on its formal description involves the investigation of a part of or the whole set of its reachable states that determines its state space. As the state space is generally infinite due to dense time semantics, we need therefore to compute finite abstractions of it, that preserve properties of interest. In these abstractions, states are grouped together, in order to obtain a finite number of these groups. These groups of states are, for instance, regions and zones for timed automata, or state classes[4] for time Petri nets. Hence, the states pertaining to each group can be described by a system of linear inequalities, noted D, whose set of solutions determines the state space of the group. Hence, if the model does not use any stopwatch, then D is of a particular form, called DBM *(Difference Bound Matrix) [8].* However, when using stopwatches, the system D becomes more complex and does not fit anymore into a DBM. In actual fact, D takes a general polyhedral form whose *canonical* form [1] is given as a conjunction of two subsystems $D = \widetilde{D} \wedge \widehat{D}$, where $\widetilde{D}$ is a DBM system and $\widehat{D}$ is a polyhedral system that cannot be encoded with $DBMs$.

The major shortcoming of manipulating polyhedra is the performance loss in terms of computation speed and memory usage. Indeed, the complexity of solving a general polyhedral system is exponential in the worst case, while it is polynomial for a DBM system. Furthermore, the reachability is proved to be undecidable for both SWA and LHA [10] [2][11], as well as for TPN extended with stopwatches [5] even when the net is bounded. As a consequence, the finiteness of the graph cannot be guaranteed.

In order to speed up the state space computation, an idea is to leave out the subsystem $\widehat{D}$, to keep only the system $\widetilde{D}$ approximating thus the space of D to the DBM containing it, see [7][15] for details. The obvious consequence of the over-approximation is that we add states in the computed group that are not reachable indeed. However, this could prevent the graph computation to terminate, by making the number of computed markings unbounded. Conversely, this can also make the computation of the approximated graph terminate by cutting off the polyhedral inequalities that prevent the convergence. It is noteworthy that since the resulted graph encompasses the exact one, only a subset of properties of interest are preserved.

In order to perform efficiently the exact analysis of the over-approximated graph, *Bucci et al* [7] have proposed to use the DBM over-approximation as a pre-computing before cleaning up the graph from its additional sequences that have been added due to over-approximation. This is done by constraining each sequence reachable in the over-approximated graph by a linear system that reproduces the

original timing constraints of the model. Hence, if there is no solution that makes the sequence be firable according to the time constraints of the system, then the sequence has been introduced by the over-approximation and can be cleared up, otherwise the solution set makes it possible to determine the feasible timings of this sequence.

Furthermore, in order to settle a compromise between both techniques, a hybrid approach has been proposed by *Roux et al* [14]. The latter puts forward a sufficient condition that determines the cases where the subsystem $\widehat{D}$ becomes redundant in D. Hence, the combination of both DBM and polyhedral representations makes it possible to build the exact state class graph faster and with lower expenses in terms of memory usage comparatively to the polyhedra based approach [13]. More recently, *Berthomieu et al* have proposed an over-approximation method based on a quantization of the polyhedral system D [5]. The latter approach ends in the exact computation of the graph in almost all cases faster than the hybrid approach [14]. Nevertheless, this technique is more costly in terms of computation time and memory usage comparatively to DBM over-approximation although it yields much precise graphs.

We consider in this paper real time preemptive systems modeled by using $ITPN$ (*Time Petri Nets with inhibitor arcs*) [15]. This model extends TPN with inhibitor arcs to control the progression and the suspension of stopwatches.

First of all, we propose a new algorithm to compute the tightest DBM over-approximation of the state class graph of preemptive systems. For this effect, we express each class, noted $\widetilde{E}$, of the approximated graph as a pair $(M, \widetilde{D})$ where M is a marking and $\widetilde{D}$ is the system of all DBM constraints, even the redundant ones. We show that by maintaining a complete representation of $\widetilde{D}$ and avoiding to compute its minimal form, we achieve to define an efficient algorithm that computes a normalized class in a square time in the number of enabled transitions. Besides, we prove that the systems $\widetilde{D}$ and $\widehat{D}$ are equivalent; this ensures that $\widetilde{D}$ is the tightest DBM over-approximation that one can derive from D. Unlike the other approaches [15][7], our algorithm avoids the computation of the intermediary polyhedra D. We thereby avoid its computation and its manipulation and remove all the costs induced by the derivation of the normal form, even the minimal form of $\widetilde{D}$. This allows to improve significantly the calculation of a class, and to remove the drawbacks that stand in the implementation of other DBM over-approximations.

In the second part of the paper, we propose another abstraction of the state space of an $ITPN$. We show that by relaxing a bit in the precision of the constraints of the system $\widetilde{D}$, we can compute smaller graphs with a minimal cost. However, although this abstraction is less precise than the former, it preserves all the firing sequences of the model and may be sufficient to model check the properties of the $ITPN$. To improve once more this construction, we provide a suitable equivalence relation that contracts the size of the resulted graphs while reducing the effort of their computation. For this effect, we show that for specific transitions, the computation of their firing distances can be useless, and we prove that the equality between these distances is not required in the class equivalence test. This result is

important since it makes it possible to gather in a same node unequal classes that are indeed bisimilar. This contraction leads to an efficient construction of DBM approximated graphs that can be in certain cases more appropriate to use to model check the linear properties of the model. Moreover, the experiments show that both constructions are faster in all cases while providing, in general, smaller graphs than other fellow approaches [15][7].

The remainder of this paper is organized as follows: In *Section 2* we present the syntax and the formal semantics of the $ITPN$ model. Then, in *Section* 3 we introduce formally our approach. In *Section* 4 we define a new construction of an abstraction of the state space of an $ITPN$. In *Section* 5 we give some experimental results that compare the performances of our algorithms with those of other approaches.

2 Time Petri Net with Inhibitor Arcs

Time Petri nets with inhibitor arcs ($ITPN$) [15] extend time Petri nets [9] with *Stopwatch inhibitor arcs*. Formally, an $ITPN$ is defined as follows:

Definition 1. *An ITPN is given by the tuple* (P, T, B, F, M^0, I, IH) *where:* P *and* T *are respectively two nonempty sets of places and transitions;* B *is the backward function* [1] *:* $B : P \times T \longrightarrow \mathbb{N} = \{0, 1, 2, ..\}$; F *is the forward function* $F : P \times T \longrightarrow \mathbb{N}$; M^0 *is the initial marking mapping* $M^0 : P \longrightarrow \mathbb{N}$ *;* I *is the delay mapping* $I : T \longrightarrow \mathbb{Q}^+ \times \mathbb{Q}^+ \cup \{\infty\}$ *, where* $\mathbb{Q}^+$ *is set of non negative rational. We write* $I(t) = [tmin(t), tmax(t)]$ *such that* $0 \leq tmin(t) \leq tmax(t)$; $IH : P \times T \longrightarrow \mathbb{N}$ *is the inhibitor arc function; there is an inhibitor arc connecting the place* p *to the transition* t*, if* $IH(p, t) \neq 0$.

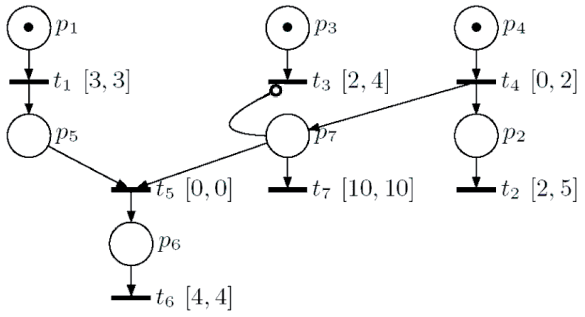

Figure 1: An $ITPN$ model

For instance, let us consider the $ITPN$ model shown in *Figure 1,* already presented in [15]. Therein, the *inhibitor arc* is the arc ended by a circle that connects

[1] $\mathbb{N}$ denotes the set of positive integers. In the graphical representation, we represent only arcs of non null valuation, and those valued 1 are implicit.

the place p_7 to the transition t_3. Initially, the place p_3 is marked but not the place p_7; hence t_3 is enabled but not inhibited. Therefore, t_3 is progressing, as t_4 which is also enabled for the initial marking. However, the firing of the transition t_4 consumes the token in the place p_4 and produces a new one in p_2 and another one in p_7. Therefore, the inhibitor arc connected to t_3 is activated and hence the clock of t_3 is suspended (t_3 is thus inhibited); this time suspension lasts as long as p_7 remains marked. For more details, the formal semantics of the $ITPN$ model is introduced in the next section.

Let $RT := (P, T, B, F, M^0, I, IH)$ be an *ITPN.*

- We call a *marking* the mapping, noted M, which associates with each place a number of tokens: $M : P \to \mathbb{N}$.

- A transition t is said to be *enabled* for the marking M, if $\forall p \in P, B(p,t) \leq M(p)$; the number of tokens in each input place of t is greater or equal to the valuation of the arc connecting this place to the transition t. Thereafter, we denote by $Te(M)$ the set of transitions *enabled* for the marking M.

- A transition t is said to be *inhibited* for a marking M, if it is *enabled* and if there exists an inhibitor arc connected to t, such that the marking satisfies its valuation $(t \in Te(M)) \wedge \exists p \in P, 0 < IH(p,t) \leq M(p)$. We denote by $Ti(M)$ the set of transitions that are *inhibited* for the marking M.

- A transition t is said to be *activated* for a marking M, if it is enabled and not inhibited, $(t \in Te(M)) \wedge \ (t \notin Ti(M))$; we denote by $Ta(M)$ the set of transitions that are *activated* for the marking M.

- Let M be a marking ; two transitions t_i and t_j enabled for M are said to be *conflicting* for M, if $\exists p \in P, \quad B(p,t_i) + B(p,t_j) > M(p)$.

- We note $Conf(M)$ the relation built on $Te(M)^2$ such that $(t_1, t_2) \in Conf(M)$, iff t_1 and t_2 are in conflict for the marking M.

For instance, let us consider again the $ITPN$ of *Figure 1*; its initial marking is equal to $M^0 : \{p_1, p_3, p_4\} \to 1; \{p_2, p_5, p_6, p_7\} \to 0$; the sets of enabled, inhibited, and activated transitions for M^0 are respectively $Te(M^0) = \{t_1\}$, $Ti(M^0) = \varnothing$, and $Ta(M^0) = Te(M^0)$.

Remark 1. We assume in the sequel a monoserver semantics, which means that for a given marking a transition can be enabled at most once .

We define the semantics of an $ITPN$ as follows:

Definition 2. *The semantics of an $ITPN$ is defined as a LTS (labeled transition system), $ST = (\Gamma, e^0, \to)$, such that:*

- Γ *is the set of reachable states: Each state, noted* e*, pertaining to* Γ *is a pair* (M, V) *where* M *is a marking and* V *is a valuation function that associates with each enabled transition* t *of* $Te(M)$ *a time interval that gives the range of relative times within which* t *can be fired. Formally we have :* $\forall t \in Te(M), \quad V(t) := [x(t), y(t)]$

- $e^0 = (M^0, V^0)$ *is the initial state, such that:* $\forall t \in Te(M^0), \quad V^0(t) := I(t) := [tmin(t), tmax(t)]$.

- $\rightarrow \in \Gamma \times (T \times \mathbb{Q}^+) \times \Gamma$ *is a transition relation, such that* $((M, V), (t_f, \underline{t_f}), (M^\uparrow, V^\uparrow)) \in \rightarrow$, *iff:*

  (i) $t_f \in Ta(M)$.

  (ii) $x(t_f) \leq \underline{t_f} \leq \underset{\forall t \in Ta(M)}{MIN} \{y(t)\}$.

  and we have:

  $\forall p \in P,\ M^\uparrow(p) := M(p) - B(p, t_f) + F(p, t_f)$.

  $\forall t \in Te(M^\uparrow)$

  if $t \notin New(M^\uparrow)$*:*

  $$[x^\uparrow(t),\ y^\uparrow(t)] := [MAX(0,\ x(t) - \underline{t_f}),\ \ y(t) - \underline{t_f}] \quad t \in Ta(M)$$
  $$[x^\uparrow(t),\ y^\uparrow(t)] := [x(t),\ \ y(t)] \quad t \in Ti(M)$$

  if $t \in New(M^\uparrow)$

  $$[x^\uparrow(t),\ y^\uparrow(t)] := I(t) = [tmin(t),\ tmax(t)]$$

  - *where* $New(M^\uparrow)$ *denotes the set of transitions newly enabled for the marking* $M^\uparrow$*. These transitions are those enabled for*$M^\uparrow$ *and not for* M*, or those enabled for* $M^\uparrow$ *and* M *but are conflicting with* t_f *for the marking* M*. Otherwise, an enabled transition which does not belong to* $New(M^\uparrow)$ *is said to be persistent.*

If t is an enabled transition for a state e, we note $\underline{t}$ the clock associated with t that takes its values in $\mathbb{Q}^+$. $\underline{t}$ measures the residual time of the transition t relatively to the instant where the state e is reached. The time progresses only for activated transitions, whereas it is suspended for inhibited transitions. Therefore, a transition t_f can be fired at relative time $\underline{t_f}$ from a reachable state e, if (i) t_f is *activated* for the marking M, and if (ii) the time can progress within the firing interval of t_f while satisfying the time constraints of other activated transitions. After firing t_f the reachable state, noted $e^\uparrow$, is obtained:

- by consuming a number of tokens in each input place p of t_f (given by the value $B(p, t_f)$), and by producing a number of tokens in each output place p of t_f (given by the value $F(p, t_f)$);

- by shifting the interval of a persistent activated transition with the value of the firing time of t_f. However, the intervals of persistent inhibited transitions remain unchanged. Finally, newly enabled transitions are assigned their static firing intervals.

Similarly as for TPN, the behavior of an $ITPN$ can be defined as a timed sequence of pairs (t_f, δ), where t_f is a transition of the net and $\delta \in \mathbb{Q}^+$. Therefore, the timed sequence $S^* = ((t_f^1, \delta^1), (t_f^2, \delta^2), .., (t_f^n, \delta^n))$ denotes that t_f^1 is fired after $\underline{t_f^1}$= δ^1 time units, then t_f^2 is fired at relative time $\underline{t_f^2}$=δ^2 and so on, such that t_f^n is fired at relative time $\underline{t_f^n}$=δ^n after an absolute time $\sum_{i=1}^{n} \delta^i$. Moreover, we often express the behavior of the net as an *untimed sequence,* denoted by S, obtained from a timed sequence S^t by removing the firing times: If $S^* = ((t_f^1, \delta^1), (t_f^2, \delta^2), ..., (t_f^n, \delta^n))$, then $S = (t_f^1, t_f^2, .., t_f^n)$. Furthermore, a marking M is said to be *reachable* in ST if there exists an untimed sequence S in ST, going from the initial marking M^0 towards M. As the set of time values is assumed to be dense, the model ST is infinite. In order to analyze this model, we need to compute an abstraction of it that saves the most interesting properties. The *symbolic graph construction* [12] preserves the untimed sequences of ST, and makes it possible to compute a finite graph in almost all cases. However, this contraction might be infinite too when the number of reachable markings is unbounded. As this last property is undecidable for $ITPN$[15], there is no guarantee to compute a finite graph. We show hereafter how to compute the state class graph of the $ITPN$ that preserves the linear properties of the model.

3 $ITPN$ state space construction

For a TPN[9], the state class graph method [4] computes a symbolic graph that preserves mainly the linear properties of the model. Similarly, this construction can be applied to an $ITPN$. This consists in regrouping in a same class all the states reachable after firing the same untimed sequence of transitions; all the states of a same class have the same marking M. Hence, a class is defined by the pair (M, D) where M is the common marking of all the states of the class, and D is a set of inequalities encoding the firing space of the class. D is of a general form, normal form of which is expressed as a conjunction of two subsystems $\overrightarrow{D} \wedge \widehat{D}$. Actually, $\overrightarrow{D}$ contains only DBM constraints and $\widehat{D}$ contains all other constraints than DBM. In the sequel, we refer to $\overrightarrow{D}$ as the tightest DBM system that over-approximates the system D. More formally, a class is defined as follows:

Definition 3. *Let $ST = (\Gamma, e^0, \rightarrow)$ be the LTS associated with an $ITPN$. A class of states of an $ITPN$, noted E, is the set of all the states pertaining to Γ that are reachable after firing the same untimed sequence $S = (t_f^1, .., t_f^n)$ from the initial state e^0. A class E is defined by (M, D), where M is the marking reachable after firing S, and D is the firing space encoded as a set of inequalities. For $Te(M) = \{t_1, .., t_s\}$, we have : $D = \widehat{D} \wedge \overrightarrow{D}$*

$$\overrightarrow{D} := \begin{cases} \bigwedge_{i \neq j} & (\underline{t_j} - \underline{t_i} \ \leq \ d_{ij}) \\ \bigwedge_{i \leq s} & (d_{i\bullet} \leq \ \underline{t_i} \ \leq d_{\bullet i}) \end{cases}$$
$$\textit{with } (t_j, t_i) \in Te(M)^2 \ \ d_{ij} \in \mathbb{Q} \cup \{\infty\}, \ d_{\bullet i} \in \mathbb{Q}^+ \cup \{\infty\}, d_{i\bullet} \in \mathbb{Q}^+$$

$$\widehat{D} := \bigwedge_{k=1..p} \quad (\alpha_{1k}\underline{t_1} + .. + \alpha_{sk}\underline{t_s} \ \leq \ d_k)$$
$$\textit{with } d_k \in \mathbb{Q} \cup \{\infty\}, \ (\alpha_{1k}, .., \alpha_{sk}) \in \mathbb{Z}^s, p \in \mathbb{N} \ \ \textit{and}^2$$
$$\forall k, \exists (i,j), (\alpha_{ik}, \alpha_{jk}) \notin \{(0,0), (0,1), (0,-1), (1,-1)\}$$

We denote by the element $\{\bullet\}$ the instant at which the class E is reached. Therefore, the value of the clock $\underline{t_i}$ expresses the time relative to the instant $\bullet$, at which the transition t_i can be fired. To each valuation ψ satisfying the system D, corresponds a unique state $e = (M, V)$ reachable in ST after firing the sequence S.

In case of a TPN, the system D is reduced to the subsystem $\overrightarrow{D}$. The inequalities of the latter have a particular form, called DBM (*Difference Bound Matrix*)[8]. The coefficients, $d_{\bullet i}, d_{i\bullet}$ and d_{ij} are respectively, the minimum residual time to fire the transition t_i, the maximum residual time to fire the transition t_i, and the maximal firing distance of the transition t_i relatively to t_j. It should be noticed that the value of $d_{\bullet i}$ and $d_{i\bullet}$ are always positive or null, whereas the value of d_{ij} can be negative, thus denoting that there exists no state e reachable in E, such that t_i can be firable from e.

For a TPN, the firing space of a class can be always encoded as a DBM system. This form makes it possible to apply an efficient algorithm to compute the reachable class from a class E. The overall complexity of this algorithm is $O(m^3)$, where m is the number of enabled transitions in E. However, for a TPN augmented with stopwatches, the state space of a class may require polyhedra to be encoded, manipulation of which induces an exponential complexity in the worst case. The exact state class graph of an $ITPN$ is computed by enumerating and exploring all the classes reachable from the initial class E^0. However, as the number of reachable classes may be unbounded, the termination of the algorithm is undecidable. Formally, the exact state class graph of an $ITPN$ can be defined as follows [5]:

Definition 4. *The exact state class graph of an $ITPN$, denoted by GR, is the tuple $(CE, E^0, \longmapsto)$ where:*

- *CE is the set of classes reachable in GR;*
- *$E^0 = (M^0, D^0)$ is the initial class;*
  $D^0 = \left\{ \ \forall t_i \in Te(M^0), \quad tmin(t_i) \leq \underline{t_i} \leq tmax(t_i) \ \right\}$
- *$\longmapsto$ is the transition relation between classes defined on $CE \times T \times CE$, such that $((M, D), t_f, (M^{\uparrow}, D^{\uparrow})) \in \longmapsto$, iff:*

  a) *t_f is activated and the system D augmented with the firing constraints of t_f that we write $D_a = D \wedge (\forall t \in Ta(M), \quad \underline{t_f} \leq \underline{t})$ holds.*

[2] $\mathbb{Z}$ denotes the set of relative integers.

b) $\forall p \in P,\ M^{\uparrow}(p) := M(p) - B(p, t_f) + F(p, t_f)$.

c) *The system $D^{\uparrow}$ is computed from D, as follows:*

1. *In the system D_a, replace each variable $\underline{t}$ (related to an enabled transition that is not inhibited for M), by: $\underline{t} := \underline{t_f} + \underline{t'}$, thus denoting the time progression.*
2. *Eliminate then by substitution the variable $\underline{t_f}$ as well as all the variables relative to transitions disabled by the firing of t_f;*
3. *Add to the system thus computed, the time constraints relative to each newly enabled transition for $M^{\uparrow}$:*
   $\forall t_i \in New(M^{\uparrow}), \quad tmin(t_i) \leq \underline{t_i} \leq tmax(t_i)$

The last definition shows how the exact state class graph of an $ITPN$ is built. Being given a class $E = (M, D)$ and a transition t_f activated for M, the computation of a class $E^{\uparrow} = (M^{\uparrow}, D^{\uparrow})$ reachable from E by firing t_f consists in computing the reachable marking $M^{\uparrow}$ and the firing space induced by the new system $D^{\uparrow}$. The class E can fire the transition t_f, if there exists a valuation that satisfies D (a state of E), such that t_f can be fired before all the other activated transitions. The firing of t_f produces a new class $E^{\uparrow} = (M^{\uparrow}, D^{\uparrow})$ which gathers all the states reachable from those of E. The system $D^{\uparrow}$ that encodes the space of $E^{\uparrow}$ is computed from the system D augmented with the firing constraints of t_f . The substitution of variables relative to activated transitions allows shifting the time origin to the instant at which the new class $E^{\uparrow}$ is reached. Then, a new system is computed wherein the variables of transitions disabled following the firing of t_f are removed. Finally, the time constraints relative to newly enabled transitions are added.

The complexity of the firing test and the step 2 of the algorithm, depends on the form of the system D. If D includes polyhedral constraints, then the complexity of the algorithm is exponential, otherwise it is polynomial. The initial system D^0 is always in DBM form, and polyhedral constraints may appear in reachable classes only when inhibited and activated transitions are both persistently enabled in a firing sequence [7].

Knowing how to compute the successors of a class, the state class graph computation is based on a depth-first or breadth-first strategy. Then the state class graph is given as the quotient of GR by a suitable equivalence relation. This equivalence relation may be equality: two classes (M, D) and (M, D') given in their minimal form are equal if $D = D'$, or inclusion; in other terms, if $\rceil D \lceil$ denotes the set of solutions for the system D, then we have : $\rceil D \lceil \subseteq \rceil D' \lceil$. It should be noticed that the equality preserves mainly the untimed language of the model, whereas the inclusion preserves the set of reachable markings. In order to speed up the class' equivalence test, the reachable systems are computed in their minimal form; this implies that all redundant inequalities are removed. Moreover, it is proved, as for DBM systems [4], that the minimal form of a polyhedral system is unique [1]. This property is very important as it permits to detect equivalent classes by comparing their minimal form.

The algorithm given in Definition 4 can be applied to a TPN with the particularity that the system D is always encoded in $DBMs$. Besides, Berthomieu *et al* proved that the number of equivalent DBM systems computed for a TPN is finite [3]. This implies that the resulted graph is necessarily finite, if the number of reachable markings is bounded.

For TPN augmented with stopwatches, the DBM over-approximation technique has been proposed as an alternative solution to analyze preemptive real time systems [15][7]. This approach consists in cutting off the inequalities of the subsystem $\widehat{D}$ when they are generated in D. It thereby keeps only those of the subsystem $\overrightarrow{D}$ to represent an over-approximation of the space of D. This solution makes it possible to build an approximated graph with lesser expenses in terms of computation time and memory usage. In addition, the DBM over-approximation ensures that the number of computed DBM systems is always finite, whereas that of polyhedral systems may be infinite. Therefore, we can compute a finite approximated graph when the computation of the exact graph does not terminate. However, the DBM over-approximation may compute an infinity of unreachable markings while the exact construction is indeed bounded. For a better understanding of how this technique works, we apply the state class graph method to the $ITPN$ example of *Figure 1.* Let $E' = (M', D')$ be the class reachable in the exact graph after firing the sequence $S = (t_4, t_1, t_5)$ from the initial class $E^0 = (M^0, D^0)$.

$$E^0 = \left(\begin{array}{l} M^0 : p_1, p_3, p_4 \rightarrow 1 \\ D^0 : \left\{ \begin{array}{l} 3 \leq \underline{t_1} \leq 3 \\ 2 \leq \underline{t_3} \leq 4 \\ 0 \leq \underline{t_4} \leq 2 \end{array} \right. \end{array} \right) \qquad E' = \left(\begin{array}{l} M' : p_1, p_4, p_6 \rightarrow 1 \\ D' : \left\{ \begin{array}{ll} 0 \leq \underline{t_2} \leq 4 & 4 \leq \underline{t_6} \leq 4 \\ 0 \leq \underline{t_3} \leq 4 & 1 \leq \underline{t_2} + \underline{t_3} \leq 6 \end{array} \right. \end{array} \right)$$

We notice that the transition t_6 is not firable from E' since t_2 or t_3 should be fired before. Put in other way, the firing of t_6 requires that the system $D' \wedge (\underline{t_6} \leq \underline{t_3}) \wedge (\underline{t_6} \leq \underline{t_2})$ admits at least one solution; we should check whether $(\underline{t_2} = \underline{t_3} = \underline{t_6} = 4) \wedge (\underline{t_2} + \underline{t_3} \leq 6)$ holds, or not. As this last inequality is not satisfied, therefore t_6 cannot fire. The system D' contains a polyhedral constraint that cannot be reduced to a DBM. The DBM over-approximation consists in cutting off the polyhedron $1 \leq \underline{t_2} + \underline{t_3} \leq 6$ to leave only DBM constraints to represent the state space of E'. However, by doing this, t_6 becomes firable since $\underline{t_6} = 4$ holds. Therefore, the system $\widetilde{D'} = \overrightarrow{D'}$ denotes an over-approximation of the system D'. In other words, we add new states in the class E' that are not reachable indeed. Nevertheless, this construction makes it possible to preserve a subset of properties.

The computation of the tightest DBM over-approximation of a class E can be obtained by applying different algorithms [15][7]. To reduce the memory usage and to ease the equivalence test, the previous algorithms compute the DBM system of each class in its minimal form. These approaches proceed first to compute the polyhedra, then eliminate the non DBM constraints while normalizing the remaining ones. Then the process terminates by computing the final system in its minimal form.

However, the implementation of the algorithm defined [15] in ROMEO [17] has revealed a loss in the precision of the DBM approximation. This is due to some

improvements in the computation of normalized systems. On the other side, the computation of the graphs in ORIS [16] (which implements the approach defined [7]), reports very slow times although the resulted graphs are correct.

We show in the sequel that by maintaining all the DBM constraints, even the redundant ones, we succeed to compute the tightest DBM approximated class in $o(m^2)$. In concrete terms, we show that by avoiding to calculate the minimal form, we succeed to define an algorithm that computes straightforwardly a normalized DBM system. We thereby eliminate the computation and the manipulation of the intermediary polyhedra that stand in the other algorithms. Moreover, we improve greatly the implementation of the graph construction and remove the bugs reported in *ROMEO*.

Formally, a DBM over-approximated class of an $ITPN$ can be defined as follows:

Definition 5. ***(Approximated Class)****. A DBM over-approximated class of an $ITPN$, noted $\widetilde{E}$, is the pair $(M, \widetilde{D})$ such that : M is a marking and $\widetilde{D}$ is the full system of all DBM normalized inequalities, involving all variables of transitions enabled for M:*

$$\widetilde{D} = \begin{cases} \bigwedge_{\forall (t_i,t_j)\in Te(M)^2} \quad (\underline{t_j} - \underline{t_i} \ \leq \ \widetilde{d_{ij}}) \\ \bigwedge_{\forall t_i \in Te(M)} \quad (\widetilde{d_{i\bullet}} \leq \ \underline{t_i} \ \leq \widetilde{d_{\bullet i}}) \end{cases}$$

with $(t_j \neq t_i)$, $\widetilde{d_{ij}} \in \mathbb{Q} \cup \{\infty\}$, $\widetilde{d_{\bullet i}} \in \mathbb{Q}^+ \cup \{\infty\}$, $\widetilde{d_{i\bullet}} \in \mathbb{Q}^+$*:*

such that each inequality is in the normal form:
$\forall x, y, z \in Te(M)$, $\left(\widetilde{d_{xy}} \leq \widetilde{d_{xz}} + \widetilde{d_{zy}}\right) \wedge \left(\widetilde{d_{xy}} \leq \widetilde{d_{\bullet y}} - \widetilde{d_{x\bullet}}\right)$.

The space of a DBM over-approximated class is encoded by the system $\widetilde{D}$. Besides, we assume that the system $\widetilde{D}$ is given in its *normal* form. As for the minimal form[3], it is proved that this form is unique for a DBM system [7] ; all equivalent systems have the same normal form.

In the sequel, we encode the system $\widetilde{D}$ as a square matrix where each line and corresponding column, are indexed by an element of $Te(M) \cup \{\bullet\}$. In concrete terms, we have:

$\forall (t_i, t_j) \in Te(M)^2 \wedge (t_i \neq t_j)$, $\widetilde{D}[\bullet, t_i] := \widetilde{d_{\bullet i}}$; $\widetilde{D}[t_i, \bullet] := -\widetilde{d_{i\bullet}}$;
$\widetilde{D}[t_i, t_j] := \widetilde{d_{ij}}$; $\widetilde{D}[t_i, t_i] := 0$; $\widetilde{D}[\bullet, \bullet] := 0$.

These matrix notations are used to represent the coefficients of the system $\widetilde{D}$. For example, the matrix shown in *Table 1* encodes the system $\widetilde{D^0} = D^0$ associated with the initial class of the exact graph of the $ITPN$ of *Figure 1.* It is noteworthy that the approximated class $\widetilde{E^0} = (M^0, \widetilde{D^0})$ is in the normal form, and represents an exact over-approximation of the initial class $E^0 = (M^0, D^0)$ of the graph GR. The minimal form of the system $\widetilde{D^0}$ is given by: $(\underline{t_1} = 3) \wedge (2 \leq \ \underline{t_3} \ \leq 4) \wedge (0 \leq \underline{t_4} \leq 2)$.

[3]The minimal form of a DBM system is obtained from its normal form by cutting off all redundant inequalities.

Table 1: The matrix representation of the system $\widetilde{D^0}$.

$\widetilde{D^0}$	$\bullet$	t_1	t_3	t_4
$\bullet$	0	3	4	2
t_1	-3	0	1	-1
t_3	-2	1	0	0
t_4	0	3	4	0

Taking on the previous definition, if $E = (M, D)$ is a class reachable in GR, then the class $\widetilde{E} = (M, \widetilde{D})$ is an over-approximation of E, if the space of states of E is included in that of $\widetilde{E}$, and we have: $\rceil D \lceil \subseteq \left\rceil \widetilde{D} \right\lceil$. Hence, by substituting $\widetilde{E}$ for E in the graph GR, it results that the class $\widetilde{E}$ may derive additional sequences that are not firable from E in GR. We thereby obtain an over-approximation of the graph GR, that we build as defined next:

Definition 6. *The graph of DBM over-approximated classes of an ITPN, denoted by $\widetilde{GR}$, is the tuple $(\widetilde{CE}, \widetilde{E^0}, \rightsquigarrow)$, such that :*

- *$\widetilde{CE}$ is the set of approximated classes reachable in $\widetilde{GR}$;*
- *$\widetilde{E^0} = (M^0, \widetilde{D^0}) \in \widetilde{CE}$ is the initial class, such that:*

  $$\widetilde{D^0} := \begin{cases} \forall t_i \in Te(M^0), & tmin(t_i) \leq \underline{t_i} \leq tmax(t_i) \\ \forall t_i \neq t_j \in Te(M^0), & \underline{t_j} - \underline{t_i} \leq tmax(t_j) - tmin(t_i) \end{cases}$$

- *$\rightsquigarrow$ is a transition relation between approximated classes defined on $\widetilde{CE} \times T \times \widetilde{CE}$, such that$((M, \widetilde{D}), t_f, (M^{\uparrow}, \widetilde{D^{\uparrow}})) \in \rightsquigarrow$, iff :*
  - $(t_f \in Ta(M)) \wedge (\widetilde{\beta}[t_f] \geq 0)$ *such that:* $\forall x \in Te(M) \cup \{\bullet\}, \quad \widetilde{\beta}[x] = \underset{\forall t \in Ta(M)}{MIN} \left\{ \widetilde{D}[x, t] \right\}$.
  - $\forall p \in P,\ M^{\uparrow}(p) := M(p) - B(p, t_f) + F(p, t_f)$.
  - *The coefficients of the DBM inequalities of the system $\widetilde{D^{\uparrow}}$ are computed from those of $\widetilde{D}$ by applying the following algorithm:*

    $\forall t \in Te(M^{\uparrow})$

    $\widetilde{D^{\uparrow}}[t, t] := 0; \qquad \widetilde{D^{\uparrow}}[\bullet, \bullet] := 0;$

    If t is persistent

    If $t \in Ti(M)$ (t is inhibited for M)

    $$\widetilde{D^{\uparrow}}[t, \bullet] := MIN \begin{pmatrix} \widetilde{D}[t, \bullet] \\ \widetilde{D}[t_f, \bullet] + \widetilde{\beta}[t] \end{pmatrix} \quad \widetilde{D^{\uparrow}}[\bullet, t] := MIN \begin{pmatrix} \widetilde{D}[\bullet, t] \\ \widetilde{D}[t_f, t] + \widetilde{\beta}[\bullet] \end{pmatrix}$$

If $t \notin Ti(M)$ *(*t *is not inhibited for* M*)*

$\widetilde{D^{\uparrow}}[\bullet, t] := \widetilde{D}[t_f, t]$; $\quad \widetilde{D^{\uparrow}}[t, \bullet] := \widetilde{\beta}[t]$.

If t *is newly enabled.*

$\widetilde{D^{\uparrow}}[\bullet, t] := tmax(t)$; $\quad \widetilde{D^{\uparrow}}[t, \bullet] := -tmin(t)$.

$\forall (t_1, t_2) \in (Te(M^{\uparrow}))^2 \wedge (t_1 \neq t_2)$

If t_1 *or* t_2 *are newly enabled.*

$\widetilde{D^{\uparrow}}[t_1, t_2] := \widetilde{D^{\uparrow}}[\bullet, t_2] + \widetilde{D^{\uparrow}}[t_1, \bullet]$.

If t_1 *and* t_2 *are persistent.*

If $(t_1, t_2) \notin (Ti(M))^2$ *(*t_1 *and* t_2 *are not inhibited for* M*)*

$\widetilde{D^{\uparrow}}[t_1, t_2] := MIN(\widetilde{D}[t_1, t_2], \quad \widetilde{D^{\uparrow}}[\bullet, t_2] + \widetilde{D^{\uparrow}}[t_1, \bullet])$.

If $(t_1, t_2) \in (Ti(M))^2$ *(*$t_1 t_2$ *are inhibited for* M*)*

$\widetilde{D^{\uparrow}}[t_1, t_2] := MIN(\widetilde{D}[t_1, t_2], \quad \widetilde{D^{\uparrow}}[\bullet, t_2] + \widetilde{D^{\uparrow}}[t_1, \bullet])$.

If $(t_1 \in Ti(M)) \wedge (t_2 \notin Ti(M))$ *(Only* t_1 *is inhibited for* M*).*

$\widetilde{D^{\uparrow}}[t_1, t_2] := MIN(\widetilde{D}[t_1, t_2] + \widetilde{D}[t_f, \bullet], \quad \widetilde{D^{\uparrow}}[\bullet, t_2] + \widetilde{D^{\uparrow}}[t_1, \bullet])$.

If $(t_1 \notin Ti(M)) \wedge (t_2 \in Ti(M))$ *(Only* t_2 *is inhibited for* M*)*

$\widetilde{D^{\uparrow}}[t_1, t_2] := MIN(\widetilde{D}[t_1, t_2] + \widetilde{\beta}[\bullet], \quad \widetilde{D^{\uparrow}}[\bullet, t_2] + \widetilde{D^{\uparrow}}[t_1, \bullet])$.

If t is an activated transition, then $\widetilde{\beta}[t]$ denotes the minimal time distance between its firing time and any other firable transition. Further, $\widetilde{\beta}[\bullet]$ represents the maximal dwelling time in the class $\widetilde{E}$. Therefore, an activated transition t_f is *not firable* from $\widetilde{E}$, if $\widetilde{\beta}[t_f] < 0$. In other words, it does not exist any state reachable in $\widetilde{E}$ such that the valuation of the clock associated with t_f can overtake the minimal bound $tmin(t_f)$. For a better understanding, the *Figure 2.a.* depicts the computation of the coefficients $\widetilde{\beta}[t_a]$, $\widetilde{D^{\uparrow}}[\bullet, t_a]$ and $\widetilde{D^{\uparrow}}[t_a, \bullet]$ for $t_a \in Ta(M)$.

Moreover, we notice that the maximal residual time of an inhibited transition t_h can decrease after firing t_f. Besides, the minimal residual time of t_h can increase. To clarify this point, let us consider the $ITPN$ of *Figure 1.* Initially t_3 is activated with $\widetilde{D}[t_3, \bullet] = -2$, and the model can fire the transition t_4 between $[0, 2]$. After this firing, the place p_7 becomes marked, and t_3 is inhibited for the first time; we have $\widetilde{D}[t_3, \bullet] = 0$. Then, to fire the newly enabled transition t_2, it needs to let time progress at least with $tmin(t_2) = 2$, while the absolute time must not surpass $tmax(t_1) = 3$. This last constraint restricts the state space of the class reachable after firing t_2 only to states that have fire initially t_4 during $[0, 1]$. As a result, the minimal residual time of t_3 increases after the firing of t_2 (see *Figure 2.b)*

In other respects, the firing distance $\widetilde{D}[t_a, t_h]$ between an activated transition t_a and an inhibited transition t_h can only increase after firing t_f, with the maximal

dwelling time[4] in $\widetilde{E}$. Also, the distance $\widetilde{D}[t_h, t_a]$ can only decrease after firing t_f with the the minimal dwelling time[5] in $\widetilde{E}$.

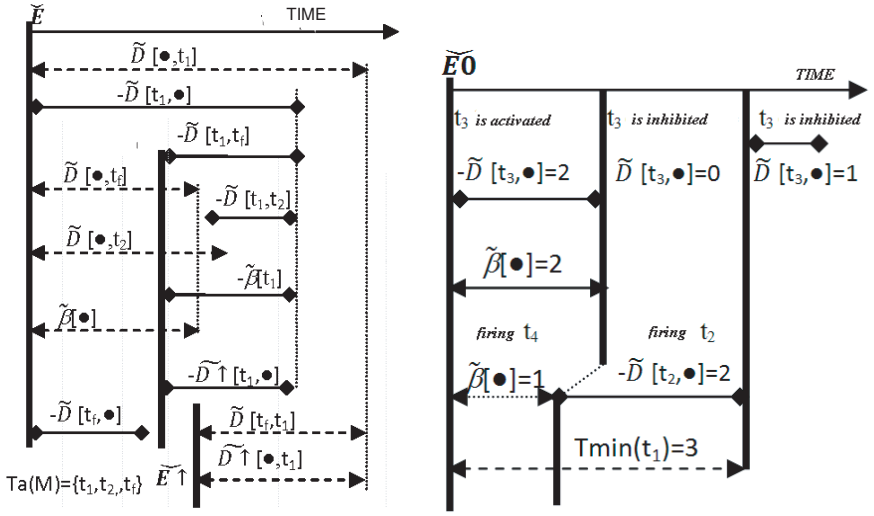

Figure 2: Computing the *DBM* coefficients.

It is noteworthy that if $\widetilde{E}$ is an over-approximation of the exact class E, then all the transitions firable from E are also firable from $\widetilde{E}$. However, a transition which is not firable from E can, on the other hand, be firable[6] from $\widetilde{E}$. Actually, as the class $\widetilde{E}$ contains all the states of E, we can find at least one state e of $\widetilde{E}$ non reachable in E, such that e can fire t_f. We prove hereafter that the algorithm given in Definition 6 computes in all cases the tightest *DBM* over-approximation of the exact graph defined in Definition 4.

Theorem 1. *The graph* $\widetilde{GR} = (\widetilde{CE}, (M^0, \widetilde{D^0}), \rightsquigarrow)$ *is the tightest DBM over-approximation that we can compute from the graph* $GR = (CE, (M^0, D^0), \longmapsto)$.

Proof. We should prove that:

1. $\rceil D^0 \lceil \subseteq \left\rceil \overrightarrow{D^0} \right\lceil = \left\rceil \widetilde{D^0} \right\lceil$.

2. Let be $S = (t_f^1, .., t_f^n)$; if $(M^0, D^0) \overset{t_f^1}{\longmapsto} .. \overset{t_f^n}{\longmapsto} E = (M, D)$ and $(M^0, \widetilde{D^0}) \overset{t_f^1}{\rightsquigarrow} .. \overset{t_f^n}{\rightsquigarrow} \widetilde{E} = (M, \widetilde{D})$, such that $\rceil D \lceil \subseteq \left\rceil \overrightarrow{D} \right\lceil = \left\rceil \widetilde{D} \right\lceil$; we have, if $E \overset{t_f}{\longmapsto} E^{\uparrow} =$

[4]This time denotes the maximal time that has elapsed for t_a and during which t_h has remained suspended.

[5]This time denotes the minimal time that has elapsed for t_a and during which t_h has remained suspended.

[6]Conversely, if t_f is not firable from $\widetilde{E}$, then it is not firable from E.

$(M^{\uparrow}, D^{\uparrow})$, then $\widetilde{E} \overset{t_f}{\rightsquigarrow} \widetilde{E^{\uparrow}} = (M^{\uparrow}, \widetilde{D^{\uparrow}})$ and $\rceil D^{\uparrow} \lceil \subseteq \left\rceil \overrightarrow{D^{\uparrow}} \right\lceil = \left\rceil \widetilde{D^{\uparrow}} \right\lceil$.

The clause (1) holds since the system D^0 is in DBM ; we have by definition : $\rceil D^0 \lceil = \left\rceil \overrightarrow{D^0} \right\lceil = \left\rceil \widetilde{D^0} \right\lceil$. Let us prove now the clause (2). For this effect, we write: t_f the transition to fire, t_h an inhibited transition of $Ti(M)$, and t_a an activated transition of $Ta(M) - \{t_f\}$. The system D stands as $D = \overrightarrow{D} \wedge \widehat{D}$, and wherein we suppose that the system $\overrightarrow{D}$ is the full system of all DBM normalized inequalities, given as follows:

$$\begin{cases} C_1 : \begin{cases} \forall t_{h_1} \neq t_{h_2} & \underline{t_{h_2}}\text{-}\underline{t_{h_1}} \leq \overrightarrow{D}[t_{h_1}, t_{h_2}] \\ \forall t_h \quad -\overrightarrow{D}[t_h, \bullet] & \leq \underline{t_h} \leq \overrightarrow{D}[\bullet, t_h] \end{cases} \\ C_2 : \begin{cases} \forall t_{a_1} \neq t_{a_2} & \underline{t_{a_2}}\text{-}\underline{t_{a_1}} \leq \overrightarrow{D}[t_{a_1}, t_{a_2}] \\ \forall t_a \quad -\overrightarrow{D}[t_a, \bullet] & \leq \underline{t_a} \leq \overrightarrow{D}[\bullet, t_a] \end{cases} \\ C_3 : \begin{cases} \forall t_a, t_h & \underline{t_a}\text{-}\underline{t_h} \leq \overrightarrow{D}[t_h, t_a] \\ \forall t_a, t_h & \underline{t_h}\text{-}\underline{t_a} \leq \overrightarrow{D}[t_a, t_h] \end{cases} \end{cases} \quad C_4 : \begin{cases} \forall t_a & \underline{t_a}\text{-}\underline{t_f} \leq \overrightarrow{D}[t_f, t_a] \\ \forall t_a & \underline{t_f}\text{-}\underline{t_a} \leq \overrightarrow{D}[t_a, t_f] \end{cases}$$

$$C_5 : \begin{cases} \forall t_h & \underline{t_h}\text{-}\underline{t_f} \leq \overrightarrow{D}[t_f, t_h] \\ \forall t_h & \underline{t_f}\text{-}\underline{t_h} \leq \overrightarrow{D}[t_h, t_f] \end{cases} \quad C_6 : -\overrightarrow{D}[t_f, \bullet] \leq \underline{t_f} \leq \overrightarrow{D}[\bullet, t_f]$$

Besides, $\widetilde{D}$ is the tightest DBM over-approximation of D; hence the next property holds $(P) : \left\rceil \widetilde{D} \right\lceil = \left\rceil \overrightarrow{D} \right\lceil$.

Let us consider now the firing of the transition t_f from E to reach the class $E^{\uparrow} = (M^{\uparrow}, D^{\uparrow})$. The calculation of the system $D^{\uparrow}$ is performed by application of the algorithm given in Definition 4. We extend D to the firing constraints of t_f , $C_7 : \forall t_a \quad \underline{t_a}\text{-}\underline{t_f} \geq 0$.

Therefore, if t_f is firable from E, then the system $\rceil D \wedge C_7 \lceil \neq \emptyset$, and we have the coefficients $\overrightarrow{D}[t_f, t_a] \geq 0$ and by using the property (P), we deduce that $\widetilde{D}[t_f, t_a] \geq 0$, hence $\widetilde{\beta}[t_f] \geq 0$. Consequently, t_f is also firable from the class $\widetilde{E}$. However, it remains to prove that $\left\rceil \overrightarrow{D^{\uparrow}} \right\lceil = \left\rceil \widetilde{D^{\uparrow}} \right\lceil$.

The computation of the system $D^{\uparrow}$ is performed by replacing each variable $\underline{t_a}$ associated with an activated transition $t_a \in Ta(M) - \{t_f\}$ by $\underline{t_a^{'}} + \underline{t_f}$. To ease the sketch of the proof, we suppose that all transitions of $Ti(M^{\uparrow}) \cup Ta(M^{\uparrow}) - \{t_f\}$ are persistent after firing t_f. Further, we limit the proof to the manipulation of DBM constraints, since we aim at computing the tightest DBM over-approximation that can be derived from D subsequently to the firing of t_f. It should be noticed that the manipulation of the constraints $\widehat{D}$ produces only new polyhedral constraints that cannot be reduced to $DBMs$ [7]. So, after substitution, the constraints of the subsystem $\overrightarrow{D} \wedge C_7$ are as follows:

$$\begin{cases} C_1 \wedge C_5 \\ C'_2 : \begin{cases} \forall t_{a_1} \neq t_{a_2} & \underline{t'_{a_2}}\text{-}\underline{t'_{a_1}} \leq \overrightarrow{D}[t_{a_1}, t_{a_2}] \\ \forall t_a \quad -\overrightarrow{D}[t_a, \bullet] & \leq \underline{t'_a} + \underline{t_f} \leq \overrightarrow{D}[\bullet, t_a] \end{cases} \\ C'_3 : \begin{cases} \forall t_a, t_h & \underline{t'_a} + \underline{t_f}\text{-}\underline{t_h} \leq \overrightarrow{D}[t_h, t_a] \\ \forall t_a, t_h & \underline{t_h}\text{-}\underline{t'_a}\text{-}\underline{t_f} \leq \overrightarrow{D}[t_a, t_h] \end{cases} \end{cases} \quad C'_7 : -\underline{t'_a} \leq 0$$

$$C'_4 : \begin{cases} \forall t_a & \underline{t'_a} \leq \overrightarrow{D}[t_f, t_a] \\ \forall t_a & -\underline{t'_a} \leq \overrightarrow{D}[t_a, t_f] \end{cases} \quad C_6 : -\overrightarrow{D}[t_f, \bullet] \leq \underline{t_f} \leq \overrightarrow{D}[\bullet, t_f]$$

By operating an intersection of the constraints C'_7 and C'_2, we obtain the system:

$$\left\{\begin{array}{ll} C_1 \wedge C_2' \wedge C_3' \wedge C_4' \wedge C_5 \wedge C_6 \wedge C_7' & \\ C_8 : \forall t_{a_1} \neq t_{a_2} & \underline{t'_{a_2}} \text{-} \underline{t'_{a_1}} \leq \overrightarrow{D}[t_{a_1}, t_{a_2}] \end{array}\right. \quad C_{12} : \left\{\begin{array}{ll} \forall t_{a_1} & \text{-}\underline{t'_{a_1}} \leq \underset{\forall t_{a_1} \neq t_{a_2}}{MIN} \left\{\overrightarrow{D}[t_{a_1}, t_{a_2}]\right\} \leq 0 \\ & \underline{t_f} \leq MIN_{\forall t_a} \left\{\overrightarrow{D}[\bullet, t_a]\right\} \end{array}\right.$$

Then by intersection and using the property $\overrightarrow{D}[t_a, t_a] = 0$, the constraints of C_{12},C_6 C_4' change into C_6' and $C_4^{"}$; we obtain:

$$\left\{\begin{array}{l} C_1 \wedge C_2' \wedge C_3' \wedge C_5 \wedge C_7' \wedge C_8 \\ C_6' : -\overrightarrow{D}[t_f, \bullet] \leq \underline{t_f} \leq \underset{\forall t \in Ta(M)}{MIN} \left\{\overrightarrow{D}[\bullet, t]\right\} \\ C_4^{"} : \forall t_a \quad - \underset{\forall t \in Ta(M)}{MIN} \overrightarrow{D}[t_a, t] \leq \underline{t'_a} \leq \overrightarrow{D}[t_f, t_a] \end{array}\right.$$

We write: $\left\{\begin{array}{ll} (F_1) : \forall t_a & \overrightarrow{D^{\uparrow}}[\bullet, t_a] := \overrightarrow{D}[t_f, t_a] \\ (F_2) : & \overrightarrow{D^{\uparrow}}[t_a, \bullet] := \underset{\forall t \in Ta(M)}{MIN} \left\{\overrightarrow{D}[t_a, t]\right\}. \end{array}\right.$

Then by using C_8 and $C_4^{"}$, we obtain:

$$\left\{\begin{array}{l} C_1 \wedge C_2' \wedge C_3' \wedge C_4^{"} \wedge C_5 \wedge C_6' \wedge C_7' \\ C_8' : \forall t_{a_1} \neq t_{a_2} \quad \underline{t'_{a_2}} \text{-} \underline{t'_{a_1}} \leq MIN(\overrightarrow{D}[t_{a_1}, t_{a_2}], \overrightarrow{D^{\uparrow}}[\bullet, t_{a_2}] + \overrightarrow{D^{\uparrow}}[t_{a_1}, \bullet]) \end{array}\right.$$

We put

$(F_3) : \forall t_{a_1} \neq t_{a_2} \quad \overrightarrow{D^{\uparrow}}[t_{a_1}, t_{a_2}] := MIN(\overrightarrow{D}[t_{a_1}, t_{a_2}], \overrightarrow{D^{\uparrow}}[\bullet, t_{a_2}] + \overrightarrow{D^{\uparrow}}[t_{a_1}, \bullet])$.

At this stage, when the model does not contain inhibitors arcs, the system D stands as $\overrightarrow{D}$, and the constraints $C_1 \wedge C_3' \wedge C_5$ are eliminated and those of $C_2' \wedge C_6' \wedge C_7'$ can be removed as they are redundant. Therefore, the new system $D^{\uparrow}$ is given by the constraints $C_4^{"} \wedge C_8'$, to which we add the constraints of newly enabled transitions. The system $\widetilde{D^{\uparrow}}$ obtained from the system $\widetilde{D}$ after firing t_f can be computed in the same way as shown previously. Assuming that, if we have $D = \widetilde{D}$, then the algorithm given in Definition 6 computes an exact approximation of the system D, since the coefficients $\widetilde{D^{\uparrow}}[t_{a_1}, t_{a_2}]$ $\widetilde{D^{\uparrow}}[\bullet, t_a]$ and $\widetilde{D^{\uparrow}}[t_a, \bullet]$ are computed also by using respectively the formulae F_3, F_1 and F_2.

On the other hand, in presence of inhibited transitions, we need to operate additional manipulations on constraints C_1, C_2', C_3' and C_5 :

By intersection of the constraints of C_6' and those of C_5, of C_3' and those of C_7', and finally, of C_3' and those of C_6'; we obtain respectively the new constraints C_5', C_9 and C_{10}:

$$\left\{\begin{array}{ll} C_1 \wedge C_2' \wedge C_3' \wedge C_4^{"} \wedge C_6' \wedge C_7' \wedge C_8' & \\ C_5' : \left\{\begin{array}{ll} \forall t_h & \underline{t_h} \leq \overrightarrow{D}[t_f, t_h] + \overrightarrow{\beta}[\bullet] \\ \forall t_h & \text{-}\underline{t_h} \leq \overrightarrow{D}[t_h, t_f] + \overrightarrow{D}[t_f, \bullet] \end{array}\right. & C_9 : \forall t_h \quad \underline{t_f} - \underline{t_h} \leq \overrightarrow{D}[t_h, t_a] \\ & C_{10} : \left\{\begin{array}{ll} \forall t_a, t_h & \underline{t'_a} \text{-} \underline{t_h} \leq \overrightarrow{D}[t_h, t_a] + \overrightarrow{D}[t_f, \bullet] \\ \forall t_a, t_h & \underline{t_h} \text{-} \underline{t'_a} \leq \overrightarrow{D}[t_a, t_h] + \overrightarrow{\beta}[\bullet] \end{array}\right. \end{array}\right.$$

with $\forall x \in Te(M) \cup \{\bullet\}$, $\overrightarrow{\beta}[x] = \underset{\forall t \in Ta(M)}{MIN} \left\{\overrightarrow{D}[x, t]\right\}$

Then by intersection of the constraints of C_9 with those of C_6', we obtain the constraints $C_9^{"}$:

$$\left\{\begin{array}{l} C_1 \wedge C_2' \wedge C_3' \wedge C_4^{"} \wedge C_5' \wedge C_6' \wedge C_7' \wedge C_8' \wedge C_{10} \\ C_9^{"} : \forall t_h, \quad -\underline{t_h} \leq \overrightarrow{D}[t_h, t_a] + \overrightarrow{D}[t_f, \bullet] \end{array}\right.$$

By using the constraints of $C_9^{"}$,C_5' and C_1 we obtain:

$$\begin{cases} C'_2 \wedge C'_3 \wedge C''_4 \wedge C'_6 \wedge C'_7 \wedge C'_8 \wedge C_{10} \\ C'_1 : \begin{cases} \forall t_{h_1} \neq t_{h_2} \quad \underline{t_{h_2}}\text{-}\underline{t_{h_1}} \leq D^{\uparrow}[t_{h_1}, t_{h_2}] \\ \forall t_h \quad -\overrightarrow{D^{\uparrow}}[t_{h_1}, \bullet]) \leq \underline{t_h} \leq \overrightarrow{D^{\uparrow}}[\bullet, t_{h_2}] \end{cases} \end{cases}$$

with

$$F_4 : \forall t_h, \quad \overrightarrow{D^{\uparrow}}[t_h, \bullet] := MIN \begin{pmatrix} \overrightarrow{D}[t_h, \bullet] \\ \overrightarrow{D}[t_f, \bullet] + \overrightarrow{\beta}[t_h] \end{pmatrix}$$

$$F_5 : \forall t_h, \quad \overrightarrow{D^{\uparrow}}[\bullet, t_h] := MIN \begin{pmatrix} \overrightarrow{D}[\bullet, t_h] \\ \overrightarrow{D}[t_f, t_h] + \overrightarrow{\beta}[\bullet] \end{pmatrix}$$

$$F_6 : \forall t_{h_1} \neq t_{h_2} \quad \overrightarrow{D^{\uparrow}}[t_{h_1}, t_{h_2}] := MIN(\overrightarrow{D}[t_{h_1}, t_{h_2}], \overrightarrow{D^{\uparrow}}[\bullet, t_{h_2}] + \overrightarrow{D^{\uparrow}}[t_{h_1}, \bullet]).$$

To achieve the proof, we proceed to the intersection of the constraints of C_{10} with those of C'_1 and C''_4; we obtain the constraints C'_{10}:

$$\begin{cases} C'_1 \wedge C'_2 \wedge C'_3 \wedge C''_4 \wedge C'_6 \wedge C'_7 \wedge C'_8 \\ C'_{10} : \begin{cases} \forall t_a, t_h \quad \underline{t'_a}\text{-}\underline{t_h} \leq \overrightarrow{D^{\uparrow}}[t_h, t_a] \\ \forall t_a, t_h \quad \underline{t_h}\text{-}\underline{t'_a} \leq \overrightarrow{D^{\uparrow}}[t_a, t_h] \end{cases} \end{cases}$$

$$(F_7) : \forall t_h \forall t_a \quad \overrightarrow{D^{\uparrow}}[t_h, t_a] := MIN(\overrightarrow{D}[t_h, t_a] + \overrightarrow{D}[t_f, \bullet], \quad \overrightarrow{D^{\uparrow}}[\bullet, t_a] + \overrightarrow{D^{\uparrow}}[t_h, \bullet]).$$

$$(F_8) : \forall t_h \forall t_a \quad \overrightarrow{D^{\uparrow}}[t_a, t_h] := MIN(\overrightarrow{D}[t_a, t_h] + \overrightarrow{\beta}[\bullet], \quad \overrightarrow{D^{\uparrow}}[\bullet, t_h] + \overrightarrow{D^{\uparrow}}[t_a, \bullet]).$$

The remaining manipulations allow to eliminate the transition t_f, thereby producing only polyhedral constraints that cannot fit into $DBMs$. These manipulations consists in the intersection of the constraints wherein the variable $\underline{t_f}$ occurs: C'_2, C'_3 and C'_6. Therefore, the system $\overrightarrow{D^{\uparrow}}$ is by construction the much precise DBM system that we can derive from $\overrightarrow{D}$ subsequently to the firing of the transition t_f. Further, assuming that the same algorithm is used to compute the coefficients of the system $\overrightarrow{D^{\uparrow}}$ as well as those of the system $\widetilde{D^{\uparrow}}$, then it is obvious that $\left\rceil \widetilde{D^{\uparrow}} \right\lceil = \left\rceil \overrightarrow{D^{\uparrow}} \right\lceil$; the property (P) holds for the systems $\widetilde{D^{\uparrow}}$ and $\overrightarrow{D^{\uparrow}}$. What is more, by assuming the formulae given previously, we prove that if $\widetilde{D}$ is in its normal form then the system $\widetilde{D^{\uparrow}}$ is also in normal form. Put in other way, as the initial class is in normal form, this guarantees, on a hand, that the DBM over-approximation is the tightest that we can compute from $\widetilde{D}$ subsequently to the firing of t_f. On the other hand, this implies also that the number of DBM that the algorithm can compute is finite, since all reachable approximated classes of the graph are in normal form [7]. □

Furthermore, the last algorithm should be provided with class equivalence conditions, in order to put an end to the enumeration process when the net is bounded. These conditions are based generally on the equality of markings and systems, as defined next:

Definition 7. *Two classes* $\widetilde{E} = (M, \widetilde{D})$ *and* $\widetilde{E'} = (M', \widetilde{D'})$*, reachable in* $\widetilde{GR}$ *satisfying the following conditions, are equivalent, and we write* $\widetilde{E} = \widetilde{E'}$*:*
$(i) \quad M = M' \qquad (ii) \quad \forall x, y \in (Te(M) \cup \{\bullet\})^2 \quad \widetilde{D}[x, y] = \widetilde{D'}[x, y].$

It should be noticed that the finiteness of the exact state class graph is undecidable even for bounded nets [5]. However, the graph obtained by DBM over-approximation is ensured to be finite when the net is bounded. This makes it possible to compute a finite DBM over-approximation when the exact one does not terminate. The approaches defined in [7][15] admit also that the DBM over-approximation that they compute, are the tightest possible. Nevertheless, these techniques have an additional cost comparatively to our algorithm. Concretely, these approaches proceed first to compute the polyhedra in its normal form (whose representation in memory and manipulation are costly), before removing the non DBM constraints and normalizing the DBMs. Then the process ends by computing the minimal form of the final DBM system. The normalization and the minimization induce a non neglectable computation effort. This affects the performances of the implementation of the DBM over-approximation.

Furthermore, the implementation in *ROMEO* [17] of the approach defined in [15] does not compute the tightest DBM over-approximation in much of the cases. In actual fact, *ROMEO* computes, first, the system D in its minimal form, to apply then the normalization to the set of DBM constraints. This makes it impossible to improve the performances of the tool but at the expenses of the precision of the DBM over-approximation. To highlight this point, let us consider the example of *Figure 3* already introduced in [7]. This example models three independent tasks that are conflicting for a common resource (CPU): Two periodic tasks 1 and 3 (of period 50 and 150 time units), and one sporadic task with a minimum and maximum inter-arrival times of $[100, 150]$. The task 1 (modeled by the transitions t_1 and t_4), has a higher priority than that of two other tasks, and the sporadic task has a higher priority than that of the third task. The priorities are modeled by using inhibitor arcs.

So, starting from the initial class, the firing of the sequence $(t_4, t_5, t_1, t_4, t_6, t_1, t_2, t_4, t_5, t_1, t_3, t_4, t_6, t_1, t_2, t_4, t_5, t_1, t_4, t_1, t_3, t_4, t_6, t_2, t_5, t_1, t_4, t_1, t_4, t_1, t_3, t_4, t_2)$ yields the classes $\widetilde{E}$, $\widetilde{E}$" and E by using respectively, the algorithm introduced in this paper, the DBM over-approximation implemented in the tool $ROMEO$ and finally the exact approach based on polyhedral representation [13] implemented also in $ROMEO$.

$\widetilde{E}=$

$M{:}p_2,p_3\rightarrow 1$

$\widetilde{D}$	•	t_1	t_2	t_3	t_5	t_6
•	0	40	150	140	20	28
t_1	-18	0	132	100	2	-2
t_2	-100	-60	0	40	-80	-72
t_3	-118	-100	32	0	-98	-102
t_5	-18	22	132	122	0	10
t_6	0	20	150	120	20	0

$E=\widetilde{E"}=$

$M{:}p_2,p_3\rightarrow 1$

$\widetilde{D"}$:

$$\begin{cases} 18 \le \underline{t_1} \le 40 & \underline{t_3}-\underline{t_6} \le 120 \\ 100 \le \underline{t_2} \le 150 & 0 \le \underline{t_6} \le 28 \\ 118 \le \underline{t_3} \le 140 & \underline{t_6}-\underline{t_1} \le -2 \\ 18 \le \underline{t_5} \le 20 & \underline{t_6}-\underline{t_3} \le -102 \\ \underline{t_1}-\underline{t_3} \le -100 & \\ \underline{t_1}-\underline{t_6} \le 20 & \\ \underline{t_3}-\underline{t_1} \le 100 & \end{cases}$$

We notice at this stage that all the resulted classes induce the same firing space whatever the approach we use. In the obtained classes, only the transition t_6 is inhibited and its maximal residual time is evaluated to 28. Now, let us consider the

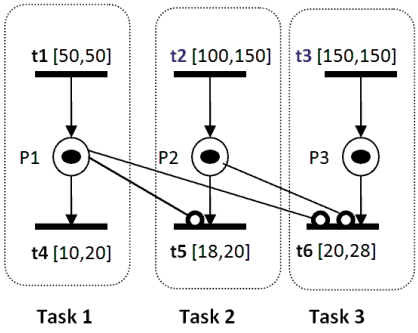

Figure 3: *ITPN modeling two periodic tasks and a sporadic one.*

firing of the transition t_1 from the previous classes to reach respectively the classes $\widetilde{E^1}$, $\widetilde{E^{"1}}$ and E^1.

$\widetilde{E^1} =$

$M^1 : p_1, p_2, p_3 \rightarrow 1$

$\widetilde{D_1}$	•	t_1	t_2	t_3	t_4	t_5	t_6
•	0	50	132	100	20	2	18
t_1	-50	0	82	50	-30	-48	-32
t_2	-80	-30	0	20	-60	-80	-62
t_3	-100	-50	32	0	-80	-98	-82
t_4	-10	40	122	90	0	-8	8
t_5	0	50	132	100	20	0	18
t_6	0	50	132	100	20	2	0

$E^1 =$

$$\begin{cases} M^1 : p_1, p_2, p_3 \rightarrow 1 \\ D^1 : \begin{cases} 50 \leq \underline{t_1} \leq 50 \\ \underline{t_2} \leq 132 \\ 100 \leq \underline{t_3} \leq 100 \\ \underline{t_5} + \underline{t_6} \leq 18 \\ \underline{t_5} - \underline{t_2} \leq \text{-}80 \\ 0 \leq \underline{t_5} \leq 2 \\ 10 \leq \underline{t_4} \leq 20 \\ 0 \leq \underline{t_6} \\ \underline{t_2} + \underline{t_6} \leq 148 \end{cases} \end{cases}$$

$$\widetilde{E^{"1}} = \begin{cases} M^1 : p_1, p_2, p_3 \rightarrow 1 \\ \widetilde{D^{"1}} : \begin{cases} 50 \leq \underline{t_1} \leq 50 & 0 \leq \underline{t_5} \leq 2 \\ 80 \leq \underline{t_2} \leq 132 & 0 \leq \underline{t_6} \leq 28 \\ 100 \leq \underline{t_3} \leq 100 & \underline{t_5} - \underline{t_2} \leq \text{-}80 \\ 10 \leq \underline{t_4} \leq 20 \end{cases} \end{cases}$$

At this stage, the persistence of the inhibited transition t_6 induces polyhedral constraints that are non redundant, as represented in the system D_1. Therefore, the DBM restriction induces an over-approximation of the exact class E_1. However, we notice that the implementation of our algorithm computes a tighter over-approximation. Concretely, the maximal residual time of the inhibited transition t_6 has decreased to 18. This value is exactly approximated by our algorithm, while it stands at 28 by using $ROMEO$. If we look to the system D_1 given in its minimal form, we notice that this time distance is worked out from the constraints $0 \leq \underline{t_5}$ and $\underline{t_5} + \underline{t_6} \leq 18$. This latter which is of a polyhedral form is removed, as well as the constraint $\underline{t_2} + \underline{t_6} \leq 148$, to determine the system $\widetilde{D_1^{"}}$. The normalization is then applied to the set of DBM constraints, whereas it should be done before

removing the polyhedral constraints. By proceeding in this way, the DBM over-approximation implemented in $ROMEO$ improves the complexity of the approach defined in [15], but at the expenses of the precision of the over-approximation.

Let us take a look at *Figure 4* to explain why the residual time of the transition t_6 has decreased during its inhibition. This figure depicts the temporal scenario of the firing sequence (t_3, t_4, t_2, t_1) leading to the class $\widetilde{E_1}$.

From the class enabling t_1 for the first time, the transition t_3 is fired without any delay. Subsequently, the newly enabled transition t_6 is inhibited, and we have $\widetilde{D}[\bullet, t_6] = 28$. The firing of t_4 between $[10, 20]$ yields a new class. There, t_6 becomes activated for the first time and its maximal residual time $\widetilde{D}[\bullet, t_6] = 28$ stands as it was. The transition t_6 is then inhibited again after the firing of t_2 during $[0, 22]$.

Then, to be able to fire the persistent transition t_1 it needs to let time progress with minimum $tmin(t_1) = 50$ from the start of the sequence. Within this intention, only the states (valuations), that fire t_2 during $[10, 22]$ (while t_6 was activated), are satisfying the firing constraints of t_1. This restriction implies that the maximal residual time of t_6 decreases of 10 units (from 28 to 20), after the firing of t_1.

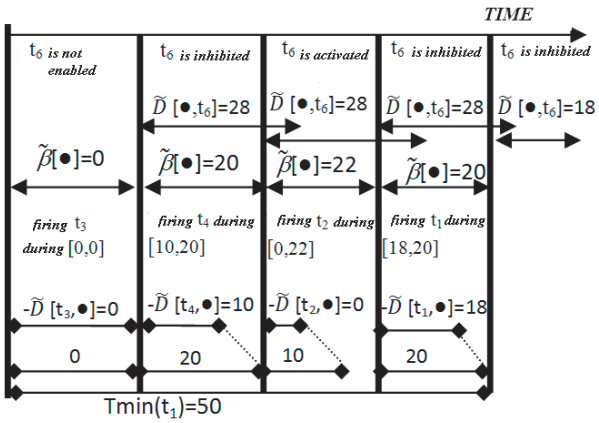

Figure 4: Temporal scenario of the firing sequence (t_3, t_4, t_2, t_1).

The loss of precision reported in the DBM over-approximation implemented in the tool $ROMEO$ can have an impact on the firability of yet persistent transitions. To illustrate this point, let us consider, for instance, the firing of the sequence (t_4, t_5) from the classes $\widetilde{E^1}$, $\widetilde{E"^1}$ and E^1, leading respectively to the classes $\widetilde{E^2}$,$\widetilde{E"^2}$ and E^2.

$\widetilde{E^2}=$ $M^2 : p_3 \to 1$

$\widetilde{D^2}$	•	t_1	t_2	t_3	t_6
•	0	40	122	90	18
t_1	-28	0	82	50	-10
t_2	-60	-30	0	20	-42
t_3	-78	-50	32	0	-60
t_6	0	40	122	90	0

$\widetilde{E^{"2}} :=$

$M^2 : p_3 \to 1$

$\widetilde{D^{"2}}$:

$28 \leq \underline{t_1} \leq 40 \quad \underline{t_1} - \underline{t_3} \leq$ -50
$60 \leq \underline{t_2} \leq 122 \quad \underline{t_2} - \underline{t_1} \leq 82$
$78 \leq \underline{t_3} \leq 90 \quad \underline{t_2} - \underline{t_3} \leq 32$
$0 \leq \underline{t_6} \leq 28 \quad \underline{t_3} - \underline{t_1} \leq 50$
$\underline{t_1} - \underline{t_2} \leq$ -30 $\quad \underline{t_3} - \underline{t_2} \leq 20$

$E^2 :=$

$M^2 : p_3 \to 1$

D^2 :

$\underline{t_1} - \underline{t_3} \leq$ -50 $\quad \underline{t_6} \leq 18$
$28 \leq \underline{t_1} \leq 40 \quad 0 \leq \underline{t_6}$
$\underline{t_1} - \underline{t_2} \leq$ -30 $\quad 60 \leq \underline{t_2}$
$\underline{t_6} + \underline{t_2} - \underline{t_1} \leq 98 \quad \underline{t_6} - \underline{t_1} \leq$ -12
$\underline{t_2} - \underline{t_1} \leq$ -82

We notice that the transition t_1 is firable from the approximated class $\widetilde{E^{"2}}$, while it is not from the exact class E^2 (since $t_6 - \underline{t_1} \leq$ -12), as well as when using our algorithm (since $\widetilde{D^2}[t_1, t_6] = -10 < 0$). This ensures that our construction yields a less coarse over-approximation than that implemented in *ROMEO*.

Comparatively to the implementation of the approach of [7] in *ORIS* tool [16], the resulted *DBM* systems are equivalent to the ones computed by our algorithm. Actually, the implemented algorithm in *ORIS* proceeds by normalizing the pre-computed systems before removing the non *DBM* constraints. However, by proceeding in this way the durations needed to compute the graphs are higher as we will see in section 5 when reporting the simulation results. Moreover, although the graphs computed by *ORIS* are different, they are bisimilar to those computed by our algorithm. Actually, a class is expressed in *ORIS* as a tuple $(M, \widetilde{D}, New)$ where New is a boolean function which indicates whether each transition enabled in the class is persistent or not. As the equivalence test implemented in *ORIS* is based on the equality of all the three parameters[7], this construction therefore yields coarser graphs.

In respect to the approach defined in [5], the latter proceeds by quantization, to over-approximate the exact polyhedra by another one. This approach is configurable, namely the precision of the approximation (grid), can be fixed before the graph computation. When using the thinnest grid, this approach succeeds to compute the exact graph in almost all cases but with the highest cost in terms of computation time and memory usage. However, as this technique still manipulates general polyhedra, it undergoes a higher computation complexity comparatively to *DBM* over approximation.

To sum up, the exact computation as well as the K-grid approximation of the graph of the net of *Figure 3* yield 320 classes. On the other hand, the *DBM*

[7] Only the equality of the parameters M and $\widetilde{D}$ is needed to decide the equivalence, as evidenced in Definition 7.

over-approximation computed by $ROMEO$ contains 403 classes and that by using $ORIS$ produces 429 classes. The implementation of the construction defined in this paper produces a more precise and compact graph of 394 classes.

We explore in the next section how to improve the construction of the DBM over-approximation. This is achieved by reducing still more its size and the effort of its computation, but however while relaxing a little bit in the precision of the over-approximation.

4 Efficient construction of a DBM over-approximation

We have proposed in the last section an efficient algorithm to compute the tightest DBM over-approximation of an $ITPN$. We have shown that the firing interval $[-\widetilde{D}[t,\bullet], \widetilde{D}[\bullet,t]$ of a persistent inhibited transition may narrow along a firing sequence. However, in almost all cases this interval remains unchanged. Therefore, we propose in the sequel to relax a little bit the DBM constraints so that to compute more compact graphs while reducing their computation effort. However, as this new abstraction is not as precise as that defined in Definition 6, it may therefore contain additional sequences. On the other hand, it makes it possible to compute in many cases, with lesser expenses, more compact graphs that are indeed bisimilar to $\widetilde{GR}$ and GR. Formally, this construction is defined as follows:

Definition 8. *The contracted DBM over-approximation graph of an $ITPN$, denoted by $\widetilde{GRC}$, is the tuple*
$(\widetilde{CEC}, \widetilde{E_c^0}, \hookrightarrow)$, *such that :*

- $\widetilde{CEC}$ *is the set of DBM over-approximated classes reachable in* $\widetilde{GRC}$;
- $\widetilde{E_c^0} = (M^0, \widetilde{D_c^0}) \in \widetilde{CEC}$ *is the initial class such that* $\widetilde{D_c^0} = \widetilde{D^0} = D^0$.
- $\hookrightarrow$ *is a transition relation between DBM over-approximated classes defined on* $\widetilde{CEC} \times T \times \widetilde{CEC}$, *such that* $((M, \widetilde{D_c}), t_f, (M^{\uparrow}, \widetilde{D_c^{\uparrow}})) \in \rightsquigarrow$, *iff :*
  - $(t_f \in Ta(M)) \wedge (\widetilde{\beta_c}[t_f] \geq 0)$ *such that:* $\forall x \in Te(M) \cup \{\bullet\}, \widetilde{\beta_c}[x] = \underset{\forall t \in Ta(M)}{MIN} \left\{\widetilde{D_c}[x,t]\right\}$.
  - $\forall p \in P,\ M^{\uparrow}(p) := M(p) - B(p,t_f) + F(p,t_f)$.
  - *The coefficients of the DBM inequalities of the system* $\widetilde{D_c^{\uparrow}}$ *are computed from those of* $\widetilde{D_c}$ *by applying the following algorithm:*
    $\forall t \in Te(M^{\uparrow})$
    $\widetilde{D_c^{\uparrow}}[t,t] := 0; \qquad \widetilde{D_c^{\uparrow}}[\bullet,\bullet] := 0.$
    If t is persistent

$$\textit{If } t \in Ti(M) \qquad \widetilde{D_c^{\uparrow}}[t,\bullet] := \widetilde{D_c}[t,\bullet]; \qquad \widetilde{D_c^{\uparrow}}[\bullet,t] := \widetilde{D_c}[\bullet,t].$$

$$\textit{If } t \notin Ti(M) \qquad \widetilde{D_c^{\uparrow}}[\bullet,t] := \widetilde{D_c}[t_f,t]\ ; \qquad \widetilde{D_c^{\uparrow}}[t,\bullet] := \widetilde{\beta_c}[t].$$

If t is newly enabled.

$$\widetilde{D_c^{\uparrow}}[\bullet,t] := tmax(t)\ ; \qquad \widetilde{D_c^{\uparrow}}[t,\bullet] := -tmin(t).$$

$$\forall (t_1,t_2) \in (Te(M^{\uparrow}))^2 \wedge (t_1 \neq t_2)$$

If t_1 or t_2 are newly enabled. $\qquad \widetilde{D_c^{\uparrow}}[t_1,t_2] := \widetilde{D_c^{\uparrow}}[\bullet,t_2] + \widetilde{D_c^{\uparrow}}[t_1,\bullet].$

If t_1 and t_2 are persistent.

If $(t_1,t_2) \notin (Ti(M))^2$ or $(t_1,t_2) \in (Ti(M))^2$

$$\widetilde{D_c^{\uparrow}}[t_1,t_2] := MIN(\widetilde{D_c}[t_1,t_2], \quad \widetilde{D_c^{\uparrow}}[\bullet,t_2] + \widetilde{D_c^{\uparrow}}[t_1,\bullet]).$$

If $(t_1,t_2) \notin (Ti(M))^2 \wedge (t_1 \in Ti(M)) \vee (t_2 \in Ti(M))$

$$\widetilde{D_c^{\uparrow}}[t_1,t_2] := \widetilde{D_c^{\uparrow}}[\bullet,t_2] + \widetilde{D_c^{\uparrow}}[t_1,\bullet].$$

Compared to the construction introduced in Definition 6, the last algorithm relaxes the constraints of persistent inhibited transitions. The firing interval of the latter is assumed unchanging, while the dwelling time in the class $\widetilde{E_c}$ is neglected when computing the firing distance between a persistent inhibited transition and a persistent activated transition. However, although this construction relaxes the constraints of each class of $\widetilde{GR}$, we need to prove formally that it computes in all cases an over-approximation of $\widetilde{GR}$, and hence of GR.

Theorem 2. *The graph $\widetilde{GRC} = (\widetilde{CE}, (M^0, \widetilde{D_c^0}), \hookrightarrow)$ is a DBM over-approximation of the graph $GR = (CE, (M^0, D^0), \longmapsto)$.*

Proof. The proof is conducted in the same way as for the proof of *Theorem*.1. First of all, we notice that $\left\rceil D^0 \right\lceil = \left\rceil \overrightarrow{D^0} \right\lceil = \left\rceil \widetilde{D^0} \right\lceil = \left\rceil \widetilde{D_c^0} \right\lceil$.

Then by performing the same manipulations on the system $\overrightarrow{D}$ we determine the system:

$$\left\{ \begin{array}{ll} C_1 \wedge C_2' \wedge C_3' \wedge C_4'' \wedge C_6' \wedge C_7' \wedge C_8' & C_9 : \forall t_h \quad \underline{t_f} - \underline{t_h} \leq \overrightarrow{D}[t_h,t_a] \\ C_5' : \left\{ \begin{array}{ll} \forall t_h & \underline{t_h} \leq \overrightarrow{D}[t_f,t_h] + \overrightarrow{\beta}[\bullet] \\ \forall t_h & -\underline{t_h} \leq \overrightarrow{D}[t_h,t_f] + \overrightarrow{D}[t_f,\bullet] \end{array} \right. & C_{10} : \left\{ \begin{array}{ll} \forall t_a,t_h & \underline{t_a'} - \underline{t_h} \leq \overrightarrow{D}[t_h,t_a] + \overrightarrow{D}[t_f,\bullet] \\ \forall t_a,t_h & \underline{t_h} - \underline{t_a'} \leq \overrightarrow{D}[t_a,t_h] + \overrightarrow{\beta}[\bullet] \end{array} \right. \end{array} \right.$$

Then the constraints of the system $\widetilde{D_c^{\uparrow}}$ are obtained by cutting off the constraints of C_5' and C_{10}. Hence we obtain a system that is less precise than $\widetilde{D^{\uparrow}}$ and $\overrightarrow{D^{\uparrow}}$ and which defines an over-approximation of the system $\overrightarrow{D^{\uparrow}}$: $\left\rceil \overrightarrow{D^{\uparrow}} \right\lceil = \left\rceil \widetilde{D^{\uparrow}} \right\lceil \subseteq \left\rceil \widetilde{D_c^{\uparrow}} \right\lceil$. □

We show in the sequel how the construction of the graph $\widetilde{GRC}$, as defined in Definition 8, can be improved still more by reducing as well as its size as the effort of its computation. For this effect, we explore hereafter an equivalence relation

that is less restrictive than the equality which we prove to be a bisimulation. This means that the graphs resulted by using this bisimulation induces the same firing sequences as when using the equality.

We first give clues about the concepts used in this bisimulation, then we define it formally:

1. From Definition 8, we notice that the firing condition of a transition t_f depends only on the sign of the coefficients $\widetilde{D_c}[t_f, t]$. Furthermore, the computation formulae of the system $\widetilde{D_c^{\uparrow}}$ from $\widetilde{D_c}$ does not use the elements $\widetilde{D_c}[\bullet, t]$ as well as $\widetilde{D_c}[t, \bullet]$ for $t \in Ta(M)$. Therefore, as the latter coefficients are not involved in the firing tests, we need not to compare them when performing the class' equivalence test. This property makes it possible to gather classes that are not equal in the graph $\widetilde{GRC}$ but which enjoy indeed the same firing sequences.

2. Let us explore now other firing distances that are useless for the enumeration process. In order to investigate this point, first we need to introduce the following notation:

   - A transition t_i is said to be *inhibiting* t_j, if $\exists p \in P, \quad 0 < IH(p, t_i) \leq B(p, t_j)$. This means that if the transition t_j is enabled for a given marking, then t_j cannot be activated for this marking. We denote hereafter by $Inhib$ the relation defined on T^2, such that $(t_i, t_j) \in Inhib$, if t_j is *inhibiting* t_i. Note that the relation $Inhib$ is not symmetric; however if (t_i, t_j) and $(t_j, t_i) \in Inhib$, then this means that t_i and t_j are always inhibited when they are enabled together.

   Let us consider two transitions that cannot be activated for a same marking (namely, t' is inhibiting t). When building the graph $\widetilde{GRC}$, it seems obvious that the time constraints of the transition t have no impact on the firing of t', and conversely. Therefore, the distances $\widetilde{D_c}[t', t]$ and $\widetilde{D_c}[t, t']$ are useless since they are not required in the firing test since t and t' cannot be activated together. Moreover, we need not even to compute these distances and to compare them when performing the equivalence test.

3. We explore now whether some distances can be left out when dealing with conflicting transitions. Before discussing this point, we need to introduce the following notations:

   - We note AT the set of transitions of T that are not connected to any inhibitor arc: $t \in AT$, if $\nexists p \in P, IH(p, t) \neq 0$.
   - Two transitions t_i and t_j are said to be *twin*, if $\forall p \in P, \quad IH(p, t_i) = IH(p, t_j)$. This means that if two transitions are enabled for a given marking, then they are both either inhibited or activated for this marking. We denote hereafter by $Twin$ the relation defined on T^2, such that

$(t_i,t_j) \in Twin$, if t_i and t_j are *twin*. Note that $AT^2 \subseteq Twin$ and we have if $(t_i,t_j) \in Twin$, then $(t_j,t_i) \in Twin$.

In [6] the authors proposed to contract the state class graph of a TPN. They proved that the firing distances between two conflicting transitions are useless when their values stand positive. Furthermore, they show that it is not required to re-compute these distances in reachable classes as long as the conflicting transitions are not disabled ahead in the firing sequence.
In actual fact, within the context of TPN, if we have $\widetilde{D_c}[t,t'] \geq 0$, then the transition t' has no impact on the firing of t as long as both remain persistent. However, if t is fired, then t' is disabled afterwards.

For an $ITPN$, the enforcement of this property may be inconsistent when dealing with conflicting transitions that are likely to be inhibited ahead in the firing sequence. For such transitions, the problem occurs, for instance, when we have two conflicting transitions t' and t activated for two different classes $\widetilde{E_c}$ and $\widetilde{E'_c}$ such that $\widetilde{D_c}[t,t'] \neq \widetilde{D_c}[t,t'] \geq 0$. If we consider these two classes as equivalent, they might not be bisimilar indeed. To be concrete, let us assume that a sequence is fired from both $\widetilde{E_c}$ and $\widetilde{E'_c}$ during which t is inhibited. Then t becomes activated in the reachable classes $\widetilde{E_c^{\uparrow}}$ and $\widetilde{E_c^{\uparrow'}}$. At this stage, the distance $\widetilde{D_c^{\uparrow}}[t,t']$ may change to negative in $\widetilde{E_c^{\uparrow}}$ but there is no guarantee that the distance $\widetilde{D_c^{\uparrow'}}[t,t']$ may change too in $\widetilde{E_c^{\uparrow'}}$. Hence, t may not be firable from $\widetilde{E_c^{\uparrow}}$, while it might be from $\widetilde{E_c^{\uparrow'}}$. Therefore, at first glance, we should restrict the application of this property only to conflicting transitions of AT; those which are not connected to any inhibitor arc. However, we show that under some assumptions the application of this property can be also extended to inhibited transitions. Actually, to validate this contraction as a bisimulation, we need to ensure that both transitions have been inhibited during the same periods of time. To guarantee that the last condition holds, we need only to assume that the conflicting transitions t' and t are *twin* .

To illustrate this bisimulation, let us consider the $ITPN$ of *Figure* 5.*a* where we have $Inhib = \{(t_4,t_3),(t_5,t_3)\}$. According to the previous discussion, the distances $\widetilde{D_c}[t_4,t_3]$, $\widetilde{D_c}[t_3,t_4]$, $\widetilde{D_c}[t_3,t_5]$ and $\widetilde{D_c}[t_5,t_3]$ should be left out during the computation of any class of the graph as well as when performing the equivalence test. Furthermore, we have $AT = \{t_1,t_2,t_3,t_6\}$ and $Twin = AT^2 \cup \{(t_4,t_5),(t_5,t_4)\}$. However, among elements of $Twin$, only transitions t_1 and t_2, on a hand, and t_4 and t_5, on the other hand, are in conflict for the initial marking. Therefore, since the distances $\widetilde{D_c^0}[t_1,t_2]$, $\widetilde{D_c^0}[t_2,t_1]$, $\widetilde{D_c^0}[t_4,t_5]$ and $\widetilde{D_c^0}[t_5,t_4]$ are positive, we need not to re-compute their values as long as the related transitions remain persistent. Furthermore, as the firing of t_1 (resp, t_4), disables t_2 (resp, t_5), and conversely, we need not too to consider these distances for the equivalence test.

The exact construction GR, the tightest DBM over-approximation $\widetilde{GR}$ and the abstraction $\widetilde{GRC}$ produce all the same graph shown in *Figure* 5.*b*. However, the

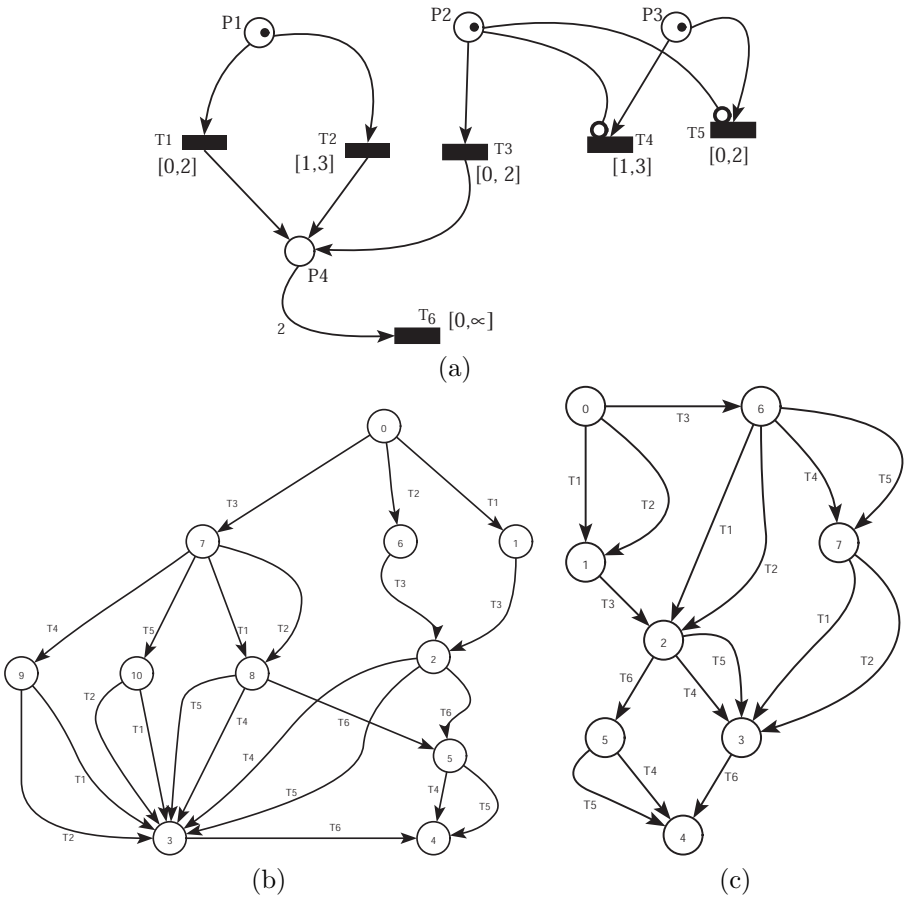

Figure 5: An $ITPN$ model with twin and inhibiting transitions, with its reachability graphs.

application of the last properties makes it possible to contract further the graph $\widetilde{GRC}$, as depicted in *Figure* 5.*c*. Although it is smaller, the resulted graph still remains bisimilar to the former, as it allows gathering classes that derive from the same firing sequences[8]. For instance, firing t_1 (resp, t_2), in $\widetilde{GRC}$ from the initial class $\widetilde{E_c^0}$ leads to the class[9] $\widetilde{E_c^1}$ (resp, $\widetilde{E_c^6}$). In actual fact, the distances $\widetilde{D_c}[\bullet, t_3], \widetilde{D_c}[t_4, t_3]$ and $\widetilde{D_c}[t_5, t_3]$ which impede the equality to hold, are useless since t_3 is inhibiting t_4 and t_5. Furthermore, the classes $\widetilde{E_c^2}$ and $\widetilde{E_c^8}$ are equivalent since the value of the minimal residual time of t_4 can be left out. Finally, the classes $\widetilde{E_c^{10}}$ and $\widetilde{E_c^9}$ can be gathered since the positive distances $\widetilde{D_c}[t_1, t_2]$ can be ignored;

[8]These classes are not equal, but they are bisimilar indeed.

[9]The class $\widetilde{E_c^i}$ corresponds to the node numbered (i) in the graph.

t_1 and t_2 are two twin conflicting transitions. Hence we obtain a more compact graph of 8 nodes and 16 edges, whereas the other constructions produce a graph of 11 classes and 22 edges.

$\widetilde{E_c^0}=$ $M^0 : p_1, p_2, p_3 \to 1$

$\widetilde{D_c^0}$	$\bullet$	t_1	t_2	t_3	t_4	t_5
$\bullet$	0	2	3	2	3	2
t_1	0	0	3	2	3	2
t_2	-1	1	0	1	2	1
t_3	0	2	3	0	3	2
t_4	-1	2	2	1	0	1
t_5	0	2	3	2	3	0

$\widetilde{E_c^1}=$ $M^1 : p_2, p_3, p_4 \to 1$

$\widetilde{D_c^1}$	$\bullet$	t_3	t_4	t_5
$\bullet$	0	2	3	2
t_3	0	0	3	0
t_4	1	1	0	1
t_5	0	2	3	0

$\widetilde{E_c^6}=$ $M^6 : p_2, p_3, p_4 \to 1$

$\widetilde{D_c^6}$	$\bullet$	t_3	t_4	t_5
$\bullet$	0	1	3	2
t_3	0	0	3	0
t_4	1	0	0	1
t_5	0	1	3	0

$\widetilde{E_c^2}=$ $M^2 : p_4 \to 1; p_4 \to 2$

$\widetilde{D_c^2}$	$\bullet$	t_4	t_5	t_6
$\bullet$	0	3	2	∞
t_4	-1	0	1	∞
t_5	0	3	0	∞
t_6	0	3	2	0

$\widetilde{E_c^8}=$ $M^8 : p_4 \to 1; p_4 \to 2$

$\widetilde{D_c^8}$	$\bullet$	t_4	t_5	t_6
$\bullet$	0	3	2	∞
t_4	0	0	1	∞
t_5	0	3	0	∞
t_6	0	3	2	0

$\widetilde{E_c^9}=$ $M^9 : p_1, p_4 \to 1$

$\widetilde{D_c^9}$	$\bullet$	t_1	t_2
$\bullet$	0	1	2
t_1	0	0	2
t_2	0	1	0

$\widetilde{E_c^{10}}=$ $M^{10} : p_1, p_4 \to 1$

$\widetilde{D_c^{10}}$	$\bullet$	t_1	t_2
$\bullet$	0	2	3
t_1	0	0	3
t_2	0	1	0

More formally, we introduce this contraction as an equivalence relation, defined as given next:

Definition 9. *Let $\simeq$ be a relation over state classes of the graph $\widetilde{GRC}$, defined by:* $((M, \widetilde{D_c}), (M', \widetilde{D_c'})) \in \simeq$ *iff:*

(i) $M = M'$

(ii) $\forall t \in Ti(M) \quad \widetilde{D_c}[\bullet, t] = \widetilde{D_c'}[\bullet, t], \quad \widetilde{D_c}[t, \bullet] = \widetilde{D_c'}[t, \bullet]$

(iii) $\forall (t, t') \in Twin \cap Conf(M)$

$$\begin{cases} sg(\widetilde{D_c}[t,t']) = sg(\widetilde{D_c'}[t,t']) \\ \widetilde{D_c}[t,t'] = \widetilde{D_c'}[t,t'] \qquad \text{If } sg(\widetilde{D_c}[t,t']) = <_0 \end{cases}$$

(iv) $\forall (t, t') \in Te(M)^2 - (Twin \cap Conf(M))$ *such that* $(t', t), (t, t') \notin Inhib$, $\widetilde{D_c}[t, t'] = \widetilde{D_c'}[t, t']$.

where $sg(v)$ is a function which gives the sign of the value v, $sg : \mathbb{Q} \cup \{\infty\} \longrightarrow \{\geq_0, <_0\}$ such that $\geq_0$ (resp, $<_0$), denotes "positive or null" (resp, strictly negative).

In concrete terms, two classes $(M, \widetilde{D_c})$ and $(M', \widetilde{D'_c})$ are in the relation $\simeq$, iff: (i) they enjoy the same marking; (ii) the maximum and the minimum residual times of any inhibited transition must be equal in both classes; (iii) for any pair of conflicting twin enabled transitions, the firing distances involving both transitions in both classes hold the same sign, and these distances must be equal in both classes only when they are negative; (iv) For all other pairs of enabled transitions that are not in the relation $Inhib$, the firing distance involving both transitions must be equal. Let us prove now that the relation $\simeq$ is a bisimulation over the classes of the graph $\widetilde{GRC}$.

Theorem 3. *The relation $\simeq$ is a bisimulation over the graph $\widetilde{GRC}$.*

Proof. We should prove that if $(\widetilde{E_c}, \widetilde{E'_c})$ satisfies the hypotheses of Definition 9, then we have:

1 : If an activated transition t_f can fire from $\widetilde{E}_c$, then t_f can fire from $\widetilde{E'_c}$ too.

2 : If $\widetilde{E}_c \overset{t_f}{\hookrightarrow} \widetilde{E_c^{\uparrow}} \wedge \widetilde{E'_c} \overset{t_f}{\hookrightarrow} \widetilde{E_c'^{\uparrow}}$, then $(\widetilde{E_c^{\uparrow}}, \widetilde{E_c'^{\uparrow}}) \in \simeq$; $\widetilde{E_c^{\uparrow}}$ and $\widetilde{E_c'^{\uparrow}}$ satisfy Definition 9.

1. Let us assume that the transition t_f is firable from $\widetilde{E_c} = (M, \widetilde{D_c})$. As $\widetilde{E_c}$ and $\widetilde{E'_c}$ are in the relation $\simeq$, then the hypotheses of Definition 9 are satisfied. Basing on the firing condition, we need to prove that (A1): if $\widetilde{\beta_c}[t_f] \geq 0$, then $\widetilde{\beta'_c}[t_f] \geq 0$, namely that $\underset{\forall t' \in Ta(M)}{MIN} \left\{ \widetilde{D'_c}[t_f, t'] \right\} \geq 0$. As t_f and t' are both activated, then $(t_f, t'), (t', t_f) \notin Inhib$. Hence, from hypotheses (iii) and (iv) of Definition 9 we determine the property (A1); t_f is firable from $\widetilde{E'}$.

2. We have to prove that the hypotheses of Definition 9 are satisfied for $(\widetilde{E_c^{\uparrow}}, \widetilde{E_c'^{\uparrow}})$

   (a) It is obvious that as $M = M'$, we have $M^{\uparrow} = M'^{\uparrow}$.

   (b) Let us prove that $\forall t \in Ti(M^{\uparrow}), \widetilde{D_c}[\bullet, t] = \widetilde{D_c'^{\uparrow}}[\bullet, t]$. Let us replace $\widetilde{D_c^{\uparrow}}[\bullet, t]$ with its computation formula according to the status of t, as given in Definition 8.
   if $t \in New(M^{\uparrow})$, then we have :$\widetilde{D_c^{\uparrow}}[\bullet, t] = \widetilde{D_c'^{\uparrow}}[\bullet, t] = tmax(t)$.
   if $t \notin New(M^{\uparrow})$, then we should consider whether t is inhibited for M or not.

   - If $t \notin Ti(M)$, then we need to prove that $\widetilde{D_c}[t_f, t] = \widetilde{D'_c}[t_f, t]$. This last property holds since $(t_f, t) \notin Twin \cap Conf(M)$, otherwise t should be disabled after firing t_f. Furthermore, $(t_f, t) \notin Inhib$ otherwise t_f should be inhibited for M. Also $(t, t_f) \notin Inhib$, otherwise t should be inhibited for M.

- If $t \in Ti(M)$, then the proof is obvious from the hypothesis (ii).

Notice that likewise we can also prove that $\forall t \in Ta(M^{\uparrow}), \widetilde{D_c}[\bullet, t] = \widetilde{D_c'^{\uparrow}}[\bullet, t]$.

(c) We should prove that $\forall t \in Ti(M^{\uparrow}), \widetilde{D_c^{\uparrow}}[t, \bullet] = \widetilde{D_c'^{\uparrow}}[t, \bullet]$. Let us replace $\widetilde{D_c^{\uparrow}}[t, \bullet]$ with the suitable computation formula according to the status of the transition t.
if $t \in New(M^{\uparrow})$, then we have $\widetilde{D_c^{\uparrow}}[t, \bullet] = \widetilde{D_c'^{\uparrow}}[t, \bullet] = -tmin(t)$.
if $t \notin New(M^{\uparrow})$, then we should consider whether t is inhibited for M or not.

- If $t \notin Ti(M)$, then we need to prove that $\widetilde{\beta_c^{\uparrow}}[t] = \widetilde{\beta_c'^{\uparrow}}[t]$, namely that $\underset{\forall t' \in Ta(M)}{MIN} \left\{ \widetilde{D_c}[t, t'] \right\} = \underset{\forall t' \in Ta(M)}{MIN} \left\{ \widetilde{D_c'}[t, t'] \right\}$.
  If $(t, t') \notin (Twin \cap Conf(M)) \cup Inhib$, then we have $\widetilde{D_c}[t, t'] = \widetilde{D_c'}[t, t']$. However, $(t', t) \notin Inhib$ (resp, $(t, t') \notin Inhib$), otherwise t' must be inhibited (resp, t must be inhibited), for M.
  If $(t, t') \in Twin \cap Conf(M)$, then we have $sg(\widetilde{D}[t, t']) = sg(\widetilde{D'}[t, t'])$ and yet more $\widetilde{D_c}[t, t'] = \widetilde{D_c'}[t, t']$ when $sg(\widetilde{D}[t, t']) =<_0$. As $t \in Ta(M)$ and $\widetilde{D_c}[t, t] = 0$, then $\underset{\forall t' \in Ta(M)}{MIN} \left\{ \widetilde{D_c}[t, t'] \right\} \leq 0$. Therefore the value of $\widetilde{D_c}[t, t']$ has no effect on the calculation of the minimum when $sg(\widetilde{D_c}[t, t']) =\geq_0$; hence the equality holds.
- If $t \in Ti(M)$, then the proof is stemmed from the hypothesis (ii).

Notice that likewise we can also prove that $\forall t \in Ta(M^{\uparrow}), \widetilde{D_c^{\uparrow}}[t, \bullet] = \widetilde{D_c'^{\uparrow}}[t, \bullet]$.

(d) We have to prove that (A2):$\forall (t, t') \in Twin \cap Conf(M^{\uparrow})$

$$\begin{cases} sg(\widetilde{D_c^{\uparrow}}[t, t']) = sg(\widetilde{D_c'^{\uparrow}}[t, t']) \\ \widetilde{D_c^{\uparrow}}[t, t'] = \widetilde{D_c'^{\uparrow}}[t, t'] \qquad \text{If } sg(\widetilde{D_c^{\uparrow}}[t, t']) =<_0 \end{cases}$$

As we deal with safe nets, we can easily show that if two persistent transitions are in conflict for M, then they remain in conflict for $M^{\uparrow}$. Hence, $Conf(M^{\uparrow})$ consists of all the pairs of transitions $(t, t') \in Conf(M)$ that are persistent in $M^{\uparrow}$ to which we add the pairs of conflicting transitions for $M^{\uparrow}$ where at least one transition is newly enabled. First of all, it is obvious that if $t \in New(M^{\uparrow})$ or $t' \in New(M^{\uparrow})$, then (t, t') satisfies the hypothesis (A2).
Let us discuss the case where two twin conflicting transitions are persistent, $(t, t') \notin (New(M^{\uparrow}))^2$. Therefore, according to Definition 8, we have:
$\widetilde{D_c^{\uparrow}}[t, t'] = MIN(\widetilde{D_c}[t, t'], \widetilde{D_c^{\uparrow}}[t, \bullet] + \widetilde{D_c^{\uparrow}}[\bullet, t'])$.
As it is assumed that $sg(\widetilde{D_c}[t, t']) = sg(\widetilde{D_c'}[t, t'])$, and we have already

proved through $(b),(c)$ that (A3): $\forall\ t \in Te(M)$, $\widetilde{D_c^{\uparrow}}[\bullet,t] = \widetilde{D_c'^{\uparrow}}[\bullet,t]$ and $\widetilde{D_c^{\uparrow}}[t,\bullet] = \widetilde{D_c'^{\uparrow}}[t,\bullet]$, we can easily determine that $sg(\widetilde{D_c^{\uparrow}}[t,t']) = sg(\widetilde{D_c'^{\uparrow}}[t,t'])$.
Furthermore, if $sg(\widetilde{D_c^{\uparrow}}[t,t']) = sg(\widetilde{D_c'^{\uparrow}}[t,t']) =<_0$, then we should prove that $\widetilde{D_c^{\uparrow}}[t,t'] = \widetilde{D_c'^{\uparrow}}[t,t']$; two cases can be seen:

- $\widetilde{D_c^{\uparrow}}[t,t'] = \widetilde{D_c}[t,t']$: as $\widetilde{D_c}[t,t'] = \widetilde{D'}[t,t']$ when $sg(\widetilde{D_c}[t,t']) =<_0$, we guarantee that $\widetilde{D_c^{\uparrow}}[t,t'] = \widetilde{D_c'^{\uparrow}}[t,t']$.
- $\widetilde{D_c^{\uparrow}}[t,t'] = \widetilde{D_c^{\uparrow}}[t,\bullet] + \widetilde{D_c^{\uparrow}}[\bullet,t']$: the property (A3) guarantees that $\widetilde{D_c^{\uparrow}}[t,t'] = \widetilde{D_c'^{\uparrow}}[t,t']$.

(e) We have to prove that (A4): $\forall (t,t') \in Te(M^{\uparrow})^2 - ((Twin \cap Conf(M^{\uparrow}))$, such that $(t,t'),(t',t) \notin Inhib$, $\widetilde{D_c^{\uparrow}}[t,t'] = \widetilde{D_c'^{\uparrow}}[t,t']$.
If $t \in New(M^{\uparrow})$ or $t' \in New(M^{\uparrow})$, then by using property (A3) we prove that,
$\widetilde{D_c^{\uparrow}}[t,t'] = \widetilde{D_c'^{\uparrow}}[t,t'] = \widetilde{D_c^{\uparrow}}[t,\bullet] + \widetilde{D_c^{\uparrow}}[\bullet,t'] = \widetilde{D_c'^{\uparrow}}[t,\bullet] + \widetilde{D_c'^{\uparrow}}[\bullet,t']$.
Let us discuss the case where both transitions are persistent; we have either
$\widetilde{D_c^{\uparrow}}[t,t'] = MIN(\widetilde{D_c}[t,t'], \widetilde{D_c^{\uparrow}}[t,\bullet] + \widetilde{D_c^{\uparrow}}[\bullet,t'])$ or $\widetilde{D_c^{\uparrow}}[t,t'] = \widetilde{D_c^{\uparrow}}[t,\bullet] + \widetilde{D_c^{\uparrow}}[\bullet,t']$.
As $(t,t') \notin (Twin \cap Conf(M^{\uparrow}) \cup Inhib$, then $\widetilde{D_c}[t,t'] = \widetilde{D_c'}[t,t']$; hence the property (A4) holds.

□

4.1 Discussion

By avoiding, on one hand, to compute some distances when working out each reachable class, and on the other hand, to compare them during the equivalence test, we succeed in reducing the computation effort of the approximated graph $\widetilde{GRC}$. This construction achieves, in general, to significantly reduce the size of the graphs, but however loses a bit in the precision of the approximation. In fact, the relaxation of some DBM constraints in the abstraction $\widetilde{GRC}$ allows to gather many classes as equivalent even when they are not equal. On the other hand, these classes are not considered as equivalent in the other constructions GR and $\widetilde{GR}$ although they often stand bisimilar.

For example, let us go back to the net of $Figure.3$ and let us consider again the sequence leading to the classes $\widetilde{E^2}$,$\widetilde{E^{"2}}$ and E^2 (see page 20). Through the same sequence we reach the class $\widetilde{E_c^2}$ in the graph $\widetilde{GRC}$. As it is the case for the DBM over-approximation defined in [15], the class $\widetilde{E_c^2}$ allows the firing of transition t_1. On the other hand, the exact construction [13], the K-grid approximation [5], the

algorithm introduced in Definition.6 as well as the approach defined in [7] do not. Actually, the firing of the transition t_1 is due to the relaxation of the constraints of the transition t_6 in $\widetilde{E_c^2}$. However, despite that the abstraction $\widetilde{GRC}$ does not build the tightest DBM over-approximation, it succeeds in some cases to compute much compact graphs than other approaches. For instance, the resulted graph obtained for the net of $Figure.3$ contains only 309 classes.

$\widetilde{E_c^2}=$ $M^2 : p_3 \rightarrow 1$

$\widetilde{D_c^2}$	•	t_1	t_2	t_3	t_6
•	0	40	122	90	28
t_1	-28	0	82	50	0
t_2	-58	-30	0	20	-30
t_3	-78	-50	32	0	-50
t_6	0	40	122	90	0

To illustrate this fact, let us consider the firing of the transition t_6 from the previous classes $\widetilde{E^2}$,$\widetilde{E_c^2}$, $\widetilde{E"^2}$ and $E^{\mathcal{2}}$ to reach respectively the classes $\widetilde{E^3}$,$\widetilde{E_c^3}$, $\widetilde{E"^3}$ and $E^{\mathcal{3}}$.

$\widetilde{E^3}=$ $M^3 :\rightarrow 0$

$\widetilde{D^3}$	•	t_1	t_2	t_3
•	0	40	122	90
t_1	-10	0	82	50
t_2	-42	-30	0	20
t_3	-60	-50	32	0

$\widetilde{E_c^3}=$ $M^3 :\rightarrow 0$

$\widetilde{D_c^3}$	•	t_1	t_2	t_3
•	0	40	122	90
t_1	-28	0	82	50
t_2	-30	-30	0	20
t_3	-50	-50	32	0

$E^{\mathcal{3}} :=$

$M^3 :\rightarrow 0$

$D^3 :$

$\begin{cases} 12 \leq \underline{t_1} \leq 40 & \underline{t_1} - \underline{t_3} \leq \text{-}50 \\ \underline{t_1} - \underline{t_2} \leq \text{-}30 & \underline{t_2} - \underline{t_1} \leq 82 \end{cases}$

$\widetilde{E"^3}:=$

$M^3 :\rightarrow 0$

$\widetilde{D"^3} :$

$\begin{cases} 0 \leq \underline{t_1} \leq 40 & \underline{t_1} - \underline{t_2} \leq \text{-}30 \\ 32 \leq \underline{t_2} \leq 122 & \underline{t_1} - \underline{t_3} = \text{-}50 \\ 50 \leq \underline{t_3} \leq 90 & \underline{t_3} - \underline{t_2} \leq 20 \end{cases}$

As we can see, all the resulted classes contain only DBM constraints. However the ones computed by the exact approach and the K-grid based approximation are the tighter ones (see the class $E^{\mathcal{3}}$). On the other side, the DBM over-approximations introduced in Definition.6 and in [7] produce the same class $\widetilde{E^3}$, whereas the one defined in [15] as well as the construction $\widetilde{GRC}$ compute less precise DBM systems (see $\widetilde{E"^3}$ and $\widetilde{E_c^3}$). Although all these classes are unequal, they still derive the same firing sequences. Furthermore, the classes $\widetilde{E^3}$,$\widetilde{E"^3}$ and $E^{\mathcal{3}}$ stand unique and are not equal to any other reachable class in their related graphs. However, the use of the equivalence $\simeq$ rather than the equality in the construction of $\widetilde{GRC}$ makes it possible to gather in a same node four other classes that stand bisimilar to $\widetilde{E_c^3}$ in the graph $\widetilde{GRC}$. Actually, according to Definition 9, the constraints $0 \leq \underline{t_1} \leq 40$, $32 \leq \underline{t_2} \leq 122$ and $50 \leq \underline{t_3} \leq 90$ in the class $\widetilde{E_c^3}$ are not needed

to carry out the equivalence test. Therefore, despite the loss of precision in the over-approximation (which may induce additional sequences in the graph $\widetilde{GRC}$), this construction still computes a much compact graph than all other approaches.

Consequently, the abstraction over the classes of the graph $\widetilde{GRC}$ is the quotient graph of $\widetilde{GRC}$ w.r.t the relation $\simeq$. It preserves, markings and both firing sequences while it is, in general, smaller. The $\widetilde{GRC}$ may be more appropriate than $\widetilde{GR}$ to check over linear properties of the model, especially when the number of additional sequences that have been added due to constraint relaxation is limited. However, when the graph $\widetilde{GRC}$ provides a too coarse over-approximation, it may yield a larger graph than $\widetilde{GR}$; the additional sequences are too numerous to be wrapped by the contraction. Indeed, the construction of $\widetilde{GRC}$ is more convenient to build when many inhibiting and conflicting transitions are reported in the net, otherwise the construction of $\widetilde{GR}$ should be considered. In other respects, it should be noticed that all the sequences firable in GR are preserved in $\widetilde{GR}$ and hence in the $\widetilde{GRC}$.

5 Experimental results

We have implemented the algorithm using $C++$ builder language on a Windows workstation. The graph construction is based on breadth-first graph generation search strategy. The experiments have been performed on a Pentium V with a processor speed of $2,7\ GHZ$ and $1,9\ GB$ of memory capacity. The different tests have been carried out by using different tools: *TINA* tool[18], *ROMEO* tool[17], $ORIS$ tool[16] and our tool named $ITPNT$.

The performances of the experiments are assessed by considering three parameters, the number of classes, the number of edges, and finally in terms of computation times. It is noteworthy that *ROMEO* and $ORIS$ tool do not bring out some parameters; we denote that by the notation *NA* (*Not Available*). Also, we denote by *NF* (*Not Finished*) the tests that had led to memory overflows or to a big time computation; more than 5 minutes.

Through the first experiments we have checked whether the *TPN* graph con-

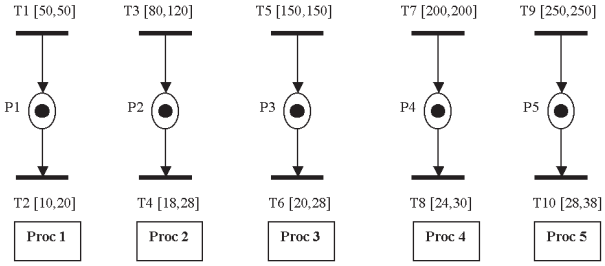

Figure 6: TPN used in the experiments.

Table 2: Results of experiments performed on TPN.

Examples	*Tools*	*TINA*	*ROMEO*	*ORIS*	*ITPNT*
Proc 1	*Classes*	2	2	2	2
	Edges	2	2	*NA*	2
	Times (ms)	0	*NA*	0	0
Proc 1 2	*Classes*	186	186	188	186
	Edges	262	262	*NA*	262
	Times (ms)	0	*NA*	16	0
Proc 1 2 3	*Classes*	958	958	1038	958
	Edges	1506	1506	*NA*	1506
	Times (ms)	1	*NA*	391	3
Proc 1 2 3 4	*Classes*	5.219	5.219	6.029	5.219
	Edges	8.580	8.580	*NA*	8.580
	Times (ms)	31	*NA*	2.719	38
Proc 1 2 3 4 5	*Classes*	42.909	42.909	52.452	42.909
	Edges	73.842	73.842	*NA*	73.842
	Times (ms)	734	*NA*	30000	786

struction by using our algorithm is conforming with that of other tools. For this effect, we have considered the combination of the TPN shown in *Figure 6*. First, we started by testing the net *Proc1,* then by combining it with *Proc2,* and so on. The results of these experiments are reported in *Table 2*. The latter shows that when assuming the equality as equivalence relation, the computed graphs are identical whatever the tool we use. However, as the expression of a class is extended to the parameter NEW in $ORIS$, the graphs computed by using this tool are coarser.

In the second series of tests, we aim at comparing the graph constructions defined in this paper with other fellow approaches. First of all, we have considered the $ITPN$ given in *Figure 3* while varying the intervals of transitions t_2,t_3 and t_6. The results of the tests are given in *Table 3*. We notice that for all the tests performed, our algorithms outperform the other tools in terms of computation time.

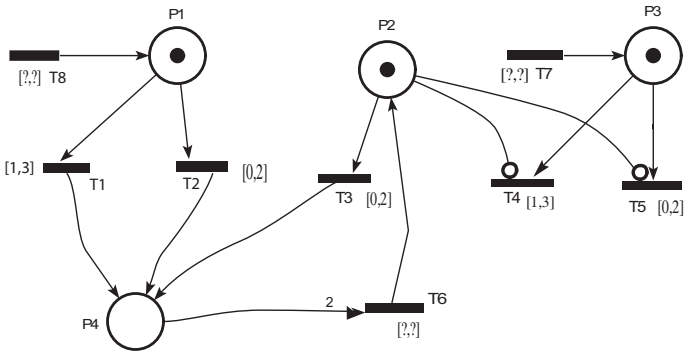

Figure 7: $ITPN$ used in the experiments.

Indeed, the optimization in the computation of DBM systems makes it possible to speed up the construction of ours graph, faster than in the other tools. Comparing to $ORIS$ (which is assumed to compute the tightest DBM over-approximation as in $ITPNT$), the computation times are reduced approximatively by 30. Concerning the tool $ROMEO$, besides that the computed graphs are less precise than ours, it seems also that the times needed for that are slower, even though they are not revealed. Moreover, the exact construction of the graph GR needed high computation times to achieve (more than 1 minute in average), even failed in many cases as evidenced in the fourth test of $Table.3$. However, in some other cases (see the two final tests in $Table.3$), the exact construction succeeds to compute finite graphs whereas all DBM-over approximations techniques fail. This happens when the additional sequences (due to over-approximation), compute persistently new markings that stand unbounded in the graph.

Table 3: Results of experiments performed with $ITPN$ of Figure 3

	TOOLS	TINA	ROMEO		ITPNT		ORIS
Examples	Methods	K-grid	Exact	DBM	$\widetilde{GR}(=)$	$\widetilde{GRC}(\simeq)$	DBM
t_2 [100,150]	Classes	4.489	4.489	5.431	5.378	5.098	5.538
t_3 [160,160]	Edges	6.360	6.360	7.608	7.530	7.251	NA
t_6 [20,28]	Times(ms)	1632	NA	NA	51	45	1578
t_2 [100,150]	Classes	320	320	403	394	309	429
t_3 [150,150]	Edges	460	460	575	562	446	NA
t_6 [20,28]	Times(ms)	110	NA	NA	2	1	156
t_2 [100,150]	Classes	4.142	4.142	5.034	4.982	4.759	5140
t_3 [140,140]	Edges	5.889	5.889	7.095	7.014	6.781	NA
t_6 [20,28]	Times(ms)	1502	NA	NA	41	37	1765
t_3 [80,120]	Classes	28 392	NF	47.622	40.842	43.462	NF
t_5 [155,155]	edges	41.452	NF	67.309	57.766	60.951	NF
t_6 [20,28]	Times (ms)	15703	NA	NA	878	886	NF
t_2 [80,120]	Classes	7.018	7.018	12.379	10.004	10.888	10.400
t_3 [140,140]	Edges	10.242	10.242	17.829	14.406	15.490	NA
t_6 [20,28]	Times(ms)	2834	NA	NA	178	162	3.516
t_2 [100,150]	Classes	11.351	11.351	16.354	15.178	16.646	15.318
t_3 [135,135]	Edges	15.649	15.649	22.230	20.486	23.225	NA
t_6 [20,28]	Times(ms)	4907	NA	NA	220	236	4765
t_2 [100,150]	Classes	17.612	17.612	21.857	21.626	22.290	21.942
t_3 [155,155]	Edges	24.522	24.522	30.065	29.711	31.151	NA
t_6 [20,28]	Times(ms)	7951	NA	NA	285	289	5594
t_2 [100,150]	Classes	12.874	12.874	NF	NF	NF	NF
t_3 [135,135]	Edges	18.424	18.424	NF	NF	NF	NF
t_6 [20,38]	Times(ms)	2340	NA	NF	NF	NF	NF
t_2 [100,150]	Classes	19.827	19.827	NF	NF	NF	NF
t_3 [155,155]	Edges	28.532	28.532	NF	NF	NF	NF
t_6 [20,39]	Times(ms)	3635	NA	NF	NF	NF	NF

Comparing to $TINA$ which implements the K-grid based approximation, there are points in our favor and other against. As the results reported with $TINA$ are obtained with the highest level grid, this construction achieves to build the exact graphs in almost all cases sensibly faster than $ROMEO$. However, although ours graphs are more coarse, they needed less time to be built (approximatively 25 times less). In other respects, the construction of the abstraction $\widetilde{GRC}$ seems to be more appropriate than that of $\widetilde{GR}$ when dealing with smaller graphs (less than 10.000 nodes). Otherwise the additional sequences due to the precision loss in $\widetilde{GRC}$ overwhelms the benefits of the contraction. However, the contraction is supposed to be more important in presence of conflicting transitions as we will see in the next experiments.

In the last experiments, we intend to advocate the benefits of building $\widetilde{GRC}$ rather than $\widetilde{GR}$ when dealing with both conflicting and inhibiting transitions. For this effect, we have considered the $ITPN$ given in *Figure 7* while varying the intervals of transitions t_6,t_7 and t_8; the results of these experiments are reported in *Table 4*.

All the experiments show that the graph computation times are in favor of our constructions, yet more when computing the abstraction $\widetilde{GRC}$. Furthermore, in the first two experiments, the construction of the graph $\widetilde{GR}$ succeeds to build the exact graph, unlike the DBM over-approximations implemented in $ROMEO$.

On the other side, the construction of the abstraction $\widetilde{GRC}$ achieves to reduce significantly the size of the graphs as well as their computation effort. Moreover, this contraction has provided very compact graphs which are even smaller than those computed by the exact approach. However, this does not mean that the graphs $\widetilde{GRC}$ are more precise in the approximation than $\widetilde{GR}$, but denote that many classes that stand unequal in $\widetilde{GR}$ and GR are bisimilar indeed. The application of the equivalency $\simeq$ makes it possible to gather these bisimilar classes, and therefore to compact sensibly the graphs.

Furthermore, we notice that for the last four tests in $Table.4$, the exact computation of the graph as well as the K-grid approximation fail to build the graphs, unlike DBM over-approximation approaches. This happens when the number of different polyhedra computed in the exact graph is unbounded, while the number of DBM systems obtained by approximation is always bounded.

6 Conclusion

We have proposed in this paper an efficient algorithm to construct the tighter DBM over-approximation of the state class graph of preemptive systems modeled by using the $ITPN$ model. Similarly as in [7][15], our approach is based on over-approximating the polyhedron of each reachable class by relaxing its non DBM constraints. For this effect, we have proposed to shun the computation of the intermediary polyhedra, and have provided an algorithm that computes efficiently and straightforwardly the full DBM system in its normal form. We have thereby

Table 4: Results of experiments performed with *ITPN* of Figure 7.

	TOOLS	TINA	ROMEO		ITPNT	
Examples	Methods	K-grid	Exact	DBM	$\widetilde{GR}(=)$	$\widetilde{GRC}(\simeq)$
t_6 [2,5]	Classes	1.035	1.035	1.318	1.035	842
t_7 [22,35]	Edges	1.830	1.830	2.323	1.830	1.471
t_8 [20,30]	Times(ms)	312	NA	NA	8	4
t_6 [2,5]	Classes	750	750	1.685	750	742
t_7 [12,15]	Edges	1.363	1.363	3.106	1.363	1.356
t_8 [10,20]	Times(ms)	219	NA	NA	4	3
t_6 [4,8]	Classes	2.346	2.346	3.398	2.402	1.880
t_7 [14,20]	Edges	4.969	4.969	7.198	5.090	4.118
t_8 [10,20]	Times(ms)	1312	NA	NA	19	12
t_6 [4,10]	Classes	3.203	3.203	4.451	3.238	2.648
t_7 [16,22]	Edges	6.603	6.603	9.184	6.756	5.758
t_8 [10,18]	Times(ms)	1.594	NA	NA	28	19
t_6 [4,8]	Classes	NF	NF	20.638	19.739	16.981
t_7 [16,22]	Edges	NF	NF	46.216	43.378	38.338
t_8 [10,15]	Times(ms)	NA	NF	NA	426	363
t_6 [4,10]	Classes	NF	NF	20.875	20.048	17.272
t_7 [16,25]	Edges	NF	NF	46.945	44.261	39.137
t_8 [10,18]	Times(ms)	NA	NF	NA	440	361
t_6 [4,10]	Classes	NF	NF	20.451	19.636	16.891
t_7 [16,25]	Edges	NF	NF	45.993	43.338	38.237
t_8 [10,20]	Times(ms)	NA	NF	NA	425	342
t_6 [4,10]	Classes	NF	NF	19.092	18.481	15.915
t_7 [16,27]	Edges	NF	NF	43.116	40.941	36.085
t_8 [10,28]	Times(ms)	NA	NF	NA	403	324

succeeded to remove the drawbacks due to the manipulation of the intermediary polyhedra, and improved significantly the graph construction by removing the cost of the normalization and the minimization of the DBM system.

Then, in the second part of this work, we have proposed a new approach to compute an abstraction of the state space of an $ITPN$. For this effect, we showed that by relaxing a little bit in the precision of the DBM over-approximation, we can compute graphs that can be more appropriate, in certain cases, to model-check the linear properties of the $ITPN$. We have discussed how this construction can be improved yet more by leaving out all the distances that are useless for the class computation process. Hence, we have put forward an equivalence relation that makes it possible to contract sensibly the size of the graphs as well as to reduce the effort of their computation. Experimental results have been reported to advocate the benefits of both constructions.

References

[1] Avis, D., K. Fukuda and S. Picozzi, On canonical representations of convex polyhedra. First International Congress of Mathematical Software (2002), pp. 350–360.

[2] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, S. Yovine: The Algorithmic Analysis of Hybrid Systems. Theor. Comput. Sci. 138(1): 3-34 (1995)

[3] Bernard Berthomieu, Miguel Menasche: An Enumerative Approach for Analyzing Time Petri Nets. IFIP Congress 1983: 41-46

[4] B. Berthomieu, and M. Diaz. "Modeling and verification of time dependant systems using Time Petri Nets". IEEE TSE, 17(3):(259-273), March 1991.

[5] Bernard Berthomieu, Didier Lime, Olivier H. Roux, François Vernadat: Reachability Problems and Abstract State Spaces for Time Petri Nets with Stopwatches. Discrete Event Dynamic Systems 17(2): 133-158 (2007).

[6] Hanifa Boucheneb, Hind Rakkay: A More Efficient Time Petri Net State Space Abstraction Useful to Model Checking Timed Linear Properties. Fundam. Inform. 88(4): 469-495 (2008).

[7] G. Bucci, A. Fedeli, L. Sassoli, and E.Vicario. Timed State Space Analysis of Real-Time Preemptive Systems. IEEE TSE, Vol 30, No. 2, Feb 2004.

[8] Dill, D.L.: Timing assumptions and verification of finite-state concurrent systems; Workshop Automatic Verification Methods for Finite-State Systems. Vol 407. (1989) 197-212.

[9] P. Merlin. "A study of the recoverability of computer system". PhD thesis Dep. Comp. Science, Uni. California, Irvine, 1974.

[10] F. Cassez and K.G. Larsen. The Impressive Power of Stopwatches. LNCS, vol. 1877, pp. 138-152, Aug. 2000.

[11] Thomas A. Henzinger: The Theory of Hybrid Automata. LICS 1996: 278-292

[12] Thomas A. Henzinger, Rupak Majumdar, Jean-François Raskin: A classification of symbolic transition systems. ACM Trans. Comput. Log. 6(1): 1-32 (2005)

[13] D.Lime, and O.H.Roux. Expressiveness and analysis of scheduling extended time Petri nets. In 5th IFAC International Conference on Fieldbus Systems and their Applications, (FET'03), Elsevier Science, July, 2003.

[14] Morgan Magnin, Didier Lime, Olivier H. Roux: An Efficient Method for Computing Exact State Space of Petri Nets With Stopwatches. Electr. Notes Theor. Comput. Sci. 144(3): 59-77 (2006).

[15] Olivier H. Roux, Didier Lime: Time Petri Nets with Inhibitor Hyperarcs. Formal Semantics and State Space Computation. ICATPN 2004: 371-390.

[16] ORIS TOOL:http://www.stlab.dsi.unifi.it/oris/index.html.

[17] ROMEO TOOL http://romeo.rts-software.org.

[18] TINA Tool http://www.laas.fr/tina/.

Received 13th January 2010