Time-Based Deadlock Prevention for Petri Nets

Kamel Barkaoui
Cedric - Cnam, Paris

Joint work with
Hanifa Boucheneb, Gaiyun Liu and Zhiwu Li

Discrete Event Systems

- Dynamic systems whose state changes only at distinct moments in time, triggered by the occurrence of discrete events.

- **Key application areas include** : RAS , Workflows, Manufacturing, logistics, transportation, telecommunications, computer networks, robotics, healthcare, and more.

  - These domains often involve concurrency, conflict, and causal relationships among events, which can frequently lead to situations where further progress is blocked.

  - For this reason, families of Petri nets are widely recognized as highly effective and efficient tools for modeling and simulating the dynamic behaviors present in these kinds of systems.

DES Behaviour

- The behavior of a system is determined by both external and internal events..
  - **Controllable events** are those that a controller can enable or disable, and they are typically observable.
  - **Uncontrollable events** occur spontaneously, cannot be prevented by the controller, and may or may not be observable.

Key behavioral properties of Petri nets

- **Boundedness** : ensures that number of system states or resources remains within a finite limit, preventing uncontrolled growth or overflow.
  - It is decidable, with EXPSPACE-complete complexity

- **Liveness :** guarantees that, from any reachable state, each transition can eventually occur.
    - Also decidable, with EXPSPACE-complete complexity.
      - certain subclasses of nets allow polynomial-time verification.
- **Deadlock-freeness** means that, in every reachable state, at least one transition remains enabled.

Liveness implies deadlock-freeness, but not vice versa.

Deadlock : occurrence of states in which the system is totally blocked.

Livelock: system keeps changing state but makes no progress.

Around deadlock state

- A **deadlock marking/state** is a marking/state with no enabled transitions.

- A bad marking is a marking that leads eventually to deadlock markings.
  - A deadlock is then a bad marking.

- A dangerous marking is a marking such that among its successors, there is, at least, a bad marking and, at least, a not bad marking.

- A **legal marking** is neither a bad nor a dangerous marking.

Deadlock control problem

The deadlock control problem seeks to prevent deadlocks and ensure liveness, especially in Petri net models

- Literature classifies deadlock control methods into three strategies.

  - **Dedlock detection and recovery**
    - Detect and resolve deadlock when it occurs

  - **Deadlock avoidance**
    - An online control policy selects feasible system actions to avoid deadlocks.

  - **Deadlock prevention**
    - **An offline policy sets constraints to prevent system deadlocks**.

Using both structural and state-space analysis

(reachability graph , siphons, resource circuits, configurations, linear programming,..)

Deadlock prevention methods

- **State space analysis-based methods**
  - maximally permissive controllers
    - prevent deadlocks without over-restricting system behavior thus maximizing resource utilization
    - cannot avoid the problem of state explosion.

- **Structural methods** (siphons in PN, circuits in graphic representation)
  - efficient deadlock controllers but limited to specific Petri net subclasses
    - does not need to analyze the state-space of the system
    - Exponential growth in structural objects; highly time-consuming

- **New method :** Time-Based Deadlock Prevention (TBDP)
  - Given a Petri Net N_u , the TBDP problem asks whether deadlocks can be prevented by assigning firing intervals that guarantee deadlock-freeness in a Time Petri Net N.
    - This may restrict transition sequences and make some markings unreachable.

Time Petri Net and its semantics (Boucheneb & al 2013)

- Formally, a TPN is a tuple (P,T,Pre,Post,M0, Is) where :
  - **(P,T,Pre,Post,M0) is a Petri net**
  - **Is is a static firing interval function (Is : T → INT)**
    - The static firing interval **[ai , bi]** of **ti** specifies its minimal and maximal firing delays (relatively to its enabling date).
    - When ti is newly enabled, **I (ti) = [ai , bi]**
      - Bounds of I(ti) decrease synchronously with time until **ti** is fired or disabled by a conflicting firing.
    - ti is firable, if **↓ I (ti) = 0**.
      - It must fire immediately, when **↑ I (ti) = 0,** unless it is disabled.
    - Its firing takes no time but leads to a new marking.
- Petri nets can be viewed as Time Petri nets in which each transition carries static interval [0,∞[.
  - So deadlock- control under timing constraints is realistic

Continued…

- For a TPN N, a state s is the pair *(M, I)* ,where M is a marking and I assigns firing intervals to all enabled transitions at M. $(I: En(M) \rightarrow I\breve{N}T_{\mathbb{R}+})$
  - Let ***(M, I)*** ,***(M', I')*** *two states,* ***dh*** $\in$ R+ a nonnegative real number, ***t*** $\in T$ a transition and → the transition relation over states that consists of continuous (time progression) and discrete transitions (firing progression)
  - **Continuous transition :** $s \xrightarrow{dh} s'$ iff *s'* is reachable from *s* in *dh* time units i.e.,
    $\forall t \in En(M), dh \leq \uparrow I(t)$, $M'=M$ and $\forall t' \in En(M'), I'(t')=[Max(0, \downarrow I(t')-dh), \uparrow I(t')-dh]$.
  - **Discrete transition :** $s \xrightarrow{t} s'$ if t is immediately firable from s and its firing leads to s', i.e.,
    $t \in En(M), \downarrow I(t)=0, \forall p \in P, M'(p)=M(p)-Pre(p,t)+Post(p,t)$ and
    $\forall t' \in En(M'), I'(t') = Is(t')$ *if* $t' \in Nw(M,t)$,$I(t')$ *otherwise.*

Property : If the **lower bounds** of the firing intervals are all **0** or the **upper bounds** of the firing intervals are all ∞ then TPN preserve the firing sequences and the markings of underlying PN.

Analysis of TPN Behavior

- Time Petri nets generate infinitely many possible successor states, since transitions can fire at any moment within their specified firing intervals.
  - As a result, it is not possible to fully capture their behavior using a standard state reachability graph.
  - In general, the problems of reachability, boundedness, and liveness are **undecidable** for Time Petri nets
- To address this challenge, **verification relies on state space abstraction techniques**.
  - States that are reachable by the same firing sequence are grouped together into sets, according to an equivalence relation—these are known as **state classes** or state zones.
  - These abstract sets make it possible to analyze Time Petri nets by **providing finite representations of their infinite state spaces, as long as the nets are bounded.**

- **The State Class Graph (SCG)** : This method groups states that are reachable by the same firing sequence into state classes—**equivalence classes defined by a common marking and a set of timing constraints known as the firing domain (**Berthomieu & al. 91).
  - SCG is finite if and only if the TPN is bounded, a sufficient condition is that the underlying Petri net is bounded
  - Its complexity is PSACE-complete (Boucheneb & al 2009).

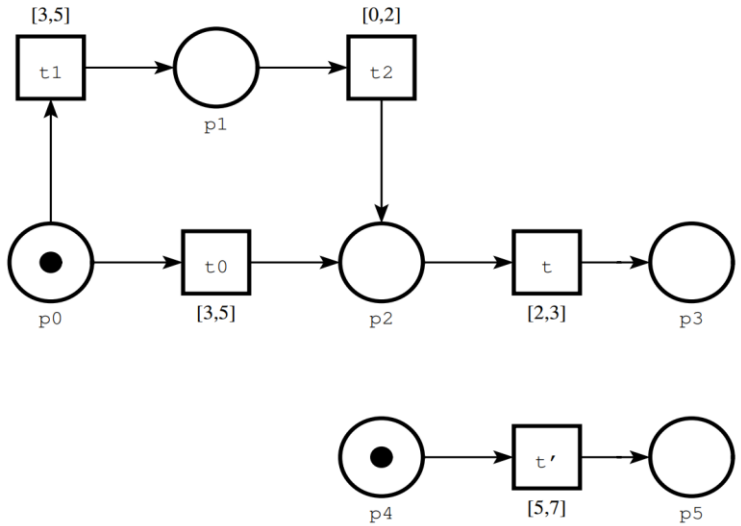
Main Steps of the SCG Method (Berthomieu & al. 91 , CSCG Boucheneb & al 08)

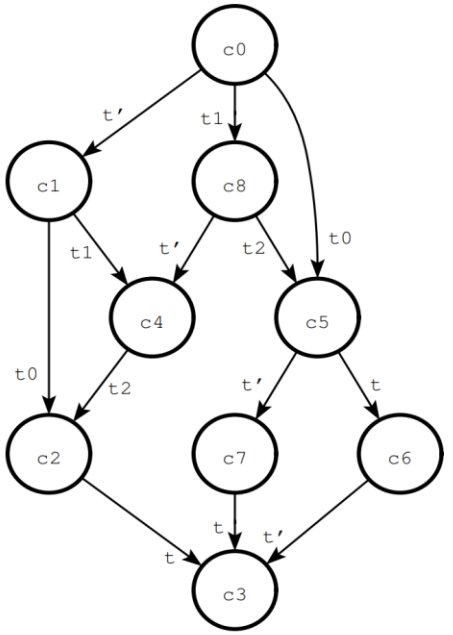
- **Initialization:**
  - Start from the initial marking M_0 and firing domain D_0 ,which expresses the bounds for all initially enabled transitions: $a(t) \leq x_t \leq b(t)$.

- **State Class Construction:**
  - For each current class (M, D) ,consider each enabled transition.
  - Apply the firing rules: when a transition t fires, update the marking and the constraints for enabled transitions. For transitions remaining enabled, preserve their clock values; clocks for newly enabled transitions are reset to zero.
- **Successor Classes:**
  - Compute successors for all firable transitions, generating new classes (M', D') corresponding to post-firing markings and updated firing domains.

- **Equivalence and Abstraction:**
  - Group together states that share the same marking and satisfy equivalent firing domains. Only distinct classes are kept in the graph, making it finite if the net is bounded.
- **Iteration:**
  - Repeat the process for all reachable classes, expanding the graph until no new classes are generated.

State Class Graph : an example from G. Berthomieu , MSR 2001





class marking firing domain	$c0$ $p0,\ p4$ $5 \leq t' \leq 7$ $3 \leq t0 \leq 5$ $3 \leq t1 \leq 5$	$c1$ $p0,\ p5$ $0 \leq t0 \leq 0$ $0 \leq t1 \leq 0$	$c2$ $p2,\ p5$ $2 \leq t \leq 3$	$c3$ $p3,\ p5$	$c4$ $p1,\ p5$ $0 \leq t2 \leq 2$
class marking firing domain	$c5$ $p2,\ p4$ $2 \leq t \leq 3$ $0 \leq t' \leq 4$	$c6$ $p3\ , p4$ $0 \leq t' \leq 2$	$c7$ $p2\ , p5$ $0 \leq t \leq 3$	$c8$ $p1\ , p4$ $0 \leq t' \leq 4$ $0 \leq t2 \leq 2$	

TBDP : A Parametric model- checking problem

Parametric model-checking involves determining whether there are parameter values for which a parametric model satisfies a given property, and, if so, identifying all such parameter values.

- **The TBDP is formulated as a parametric safety model-checking** problem in which :
  - the input model parametric TPN is a Petri net extended with $2 \times |T|$ **distinct, non-negative integer parameters**, where each parameter represents either the lower or the upper bound of a transition's firing interval.
  - the safety property to be verified is deadlock-freeness.

- **Due to the inclusion relationship between TPN extensions**,
  - **the TBDP problem can be reduced to deadlock prevention by associating to each transition a single (parametric) firing interval (** i.e. a = b).
    - As result, the number of parameters reduced by half.

TBDP problem : decidability and complexity

- The reachability problem for parametric TPN is shown to be **not decidable**, even for bounded parametric TPN. (Traonouez, L & al, 2018)

  - **However,** for the subclass known as **L/U bounded parametric TPNs**, where the parameters defining the lower bounds of firing intervals are entirely distinct from those of the upper bounds, **the reachability is decidable**. (Traonouez, L & al, 2018)

    - **This also holds for the bounded parametric TPNs used in the TBDP problem**.

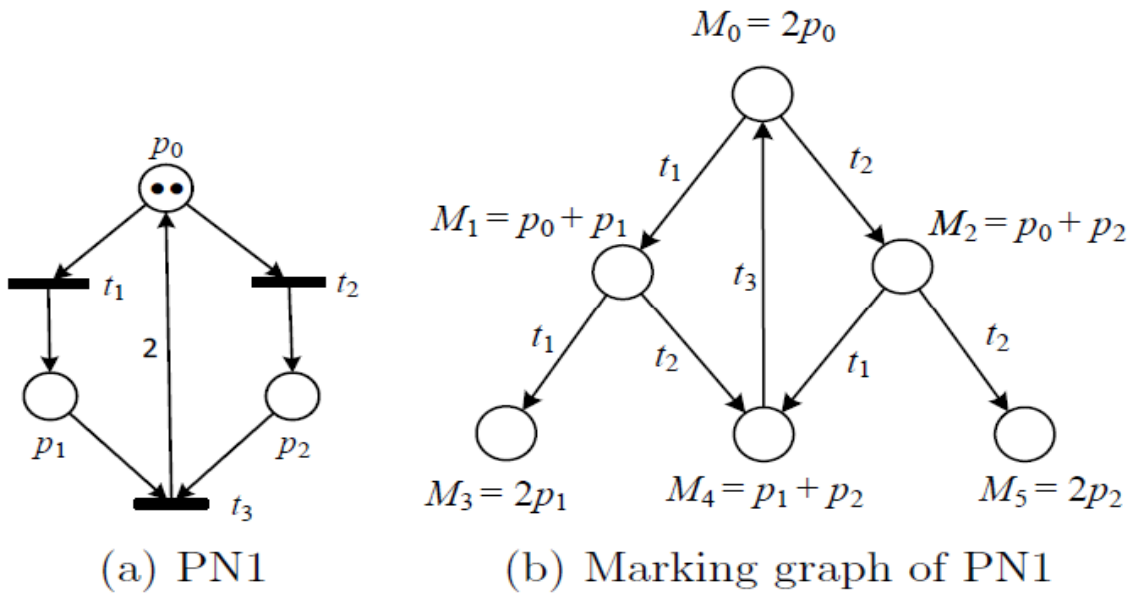
  - Moreover , any L/U bounded parametric TPN can be converted into an equivalent timed L/U parametric timed automaton, for which reachability analysis is PSPACE-complete (Hune, T. & al, 2008).

Theorem : **The TBDP problem is decidable and PSPACE-complete for bounded PN**

Example 2

Consider model **PN1** and its marking graph at Fig.1. To deal with TBDP problem for PN1:

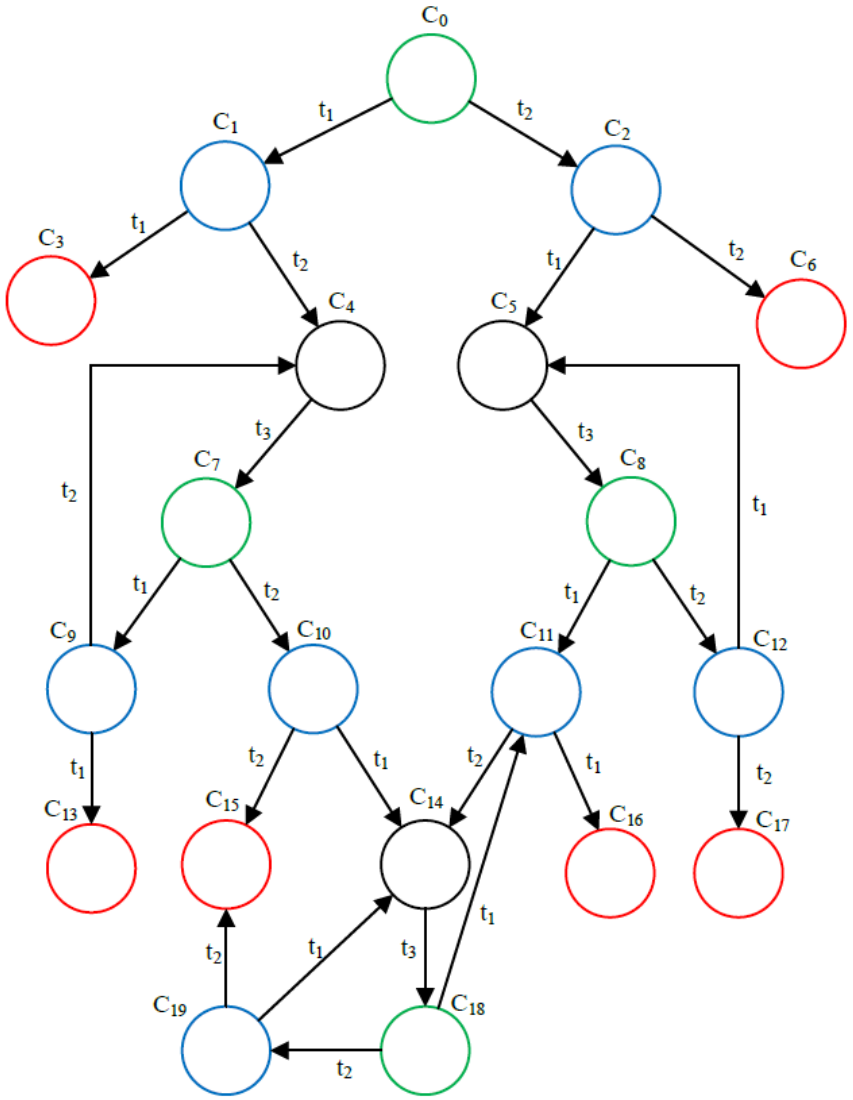
- we associate with each transition t_i of PN1, a parametric interval $\mathbf{[a_i; b_i]}$
- the **safety property AG not deadlock** (i.e all reachable markings are not deadlock).



(a) PN1 (b) Marking graph of PN1

Fig. 1. Model PN1 and its marking graph

We then use the **parametric model checker Roméo** to compute the parametric state class graph (**PSCG**) in a similar way as the non-parametric case (Lime D & all 2009)



PSCG of the parametric TPN of PN1

C	Marking	Parametric constraints
C_0	$2p_0$	$a_1 \leq t_1 \leq b_1 \wedge a_2 \leq t_2 \leq b_2 \wedge a_3 \leq b_3$
C_1	$p_0 + p_1$	$a_1 \leq t_1 \leq b_1 \wedge a_1 \leq b_2 \wedge a_2 \leq b_2 \wedge$ $a_2 - b_1 \leq t_2 \leq b_2 - a_1 \wedge a_3 \leq b_3$
C_2	$p_0 + p_2$	$a_1 - b_2 \leq t_1 \leq b_1 - a_2 \wedge$ $a_2 \leq t_2 \leq b_2 \wedge a_2 \leq b_1$
C_3	$2p_1$	$a_1 \leq b_1 \wedge 2a_1 \leq b_2 \wedge a_2 \leq b_2 \wedge a_3 \leq b_3$
C_4	$p_1 + p_2$	$a_1 \leq b_1 \wedge a_1 \leq b_2 \wedge a_2 \leq 2b_1 \wedge$ $a_2 \leq b_2 \wedge a_3 \leq t_3 \leq b_3$
C_5	$p_1 + p_2$	$a_1 \leq b_1 \wedge a_1 \leq 2b_2 \wedge$ $a_2 \leq b_1 \wedge a_2 \leq b_2 \wedge a_3 \leq t_3 \leq b_3$
C_6	$2p_2$	$a_1 \leq b_1 \wedge 2a_1 \leq b_1 \wedge a_2 \leq b_2 \wedge a_3 \leq b_3$
C_7	$2p_0$	$a_1 \leq t_1 \leq b_1 \wedge a_2 \leq t_2 \leq b_2 \wedge$ $a_1 \leq b_2 \wedge a_2 \leq 2b_1 \wedge a_3 \leq b_3$
C_8	$2p_0$	$a_1 \leq t_1 \leq b_1, a_2 \leq t_2 \leq b_2 \wedge$ $a_1 \leq 2b_2 \wedge a_2 \leq b_1 \wedge a_3 \leq b_3$
C_9	$p_0 + p_1$	$a_1 \leq t_1 \leq b_1 \wedge a_2 - b_1 \leq t_2 \leq b_2 - a_1 \wedge$ $a_1 \leq b_2 \wedge a_2 \leq 2b_1 \wedge a_2 \leq b_2 \wedge a_3 \leq b_3$
C_{10}	$p_0 + p_2$	$a_1 - b_2 \leq t_1 \leq b_1 - a_2 \wedge a_2 \leq t_2 \leq b_2 \wedge$ $a_1 \leq b_1 \wedge a_1 \leq b_2 \wedge a_2 \leq b_1 \wedge a_3 \leq b_3$
C_{11}	$p_0 + p_1$	$a_1 \leq t_1 \leq b_1 \wedge t_2 \leq b_2 - a_1 \wedge$ $a_1 \leq b_2 \wedge 0 \leq a_2 \leq b_1 \wedge a_2 \leq b_2 \wedge a_3 \leq b_3$
C_{12}	$p_0 + p_2$	$a_1 - b_2 \leq t_1 \leq b_1 - a_2 \wedge a_1 \leq b_1 \wedge a_2 \leq b_1 \wedge$ $a_2 \leq t_2 \leq b_2 \wedge a_1 \leq 2b_2 \wedge a_3 \leq b_3$
C_{13}	$2p_1$	$a_1 \leq b_1 \wedge 2a_1 \leq b_2 \wedge a_2 \leq 2b_1 \wedge$ $a_2 \leq b_2 \wedge a_3 \leq b_3$
C_{14}	$p_1 + p_2$	$a_1 \leq b_1 \wedge a_1 \leq b_2 \wedge a_2 \leq b_1 \wedge$ $a_2 \leq b_2 \wedge a_3 \leq b_3$
C_{15}	$2p_2$	$a_1 \leq b_1 \wedge a_1 \leq b_2 \wedge 2a_2 \leq b_1 \wedge$ $a_2 \leq b_2 \wedge a_3 \leq b_3$
C_{16}	$2p_1$	$a_1 \leq b_1 \wedge 2a_1 \leq b_2 \wedge a_2 \leq b_1 \wedge$ $a_2 \leq b_2 \wedge a_3 \leq b_3$
C_{17}	$2p_2$	$a_1 \leq b_1 \wedge a_1 \leq 2b_2 \wedge 2a_2 \leq b_1 \wedge$ $a_2 \leq b_2 \wedge a_3 \leq b_3$
C_{18}	$2p_0$	$a_1 \leq t_1 \leq b_1 \wedge a_2 \leq t_2 \leq b_2 \wedge$ $a_1 \leq b_2 \wedge a_2 \leq b_1 \wedge a_3 \leq b_3$
C_{19}	$p_0 + p_2$	$t_1 \leq b_1 - a_2 \wedge a_2 \leq t_2 \leq b_2 \wedge a_1 \leq b_1 \wedge$ $a_1 \leq b_2 \wedge a_2 \leq b_1 \wedge a_2 \leq b_2 \wedge a_3 \leq b_3$

20 Parametric state classes of the PSCG

The solution

- The **parametric model checker Romeo** verifies that the deadlock-free property holds for the following parameter domain $\boldsymbol{D}$:

$$\boldsymbol{D} = \{(a1, a2, a3, b1, b2, b3) \in \mathbb{N}^6 \mid \boldsymbol{a1 \leq b1 < 2a2 \wedge\ a2 \leq b2 < 2a1 \wedge\ a3 \leq b3}\}.$$

Each value in D defines a TPN extension of PN1 that remains free of deadlocks.

The constrain $\boldsymbol{b1<2a2}$ **and** $\boldsymbol{b2<2a1}$ ensure that two consecutive firings of transitions t_1 and t_2 cannot occur

- **Different solutions correspond to controllers with varying levels of permissiveness.**
  - *The solution* $a1 = b1 = 3$, $a2 = b2 = 4$ *and* $a3 = b3 = 1$ *yields a TPN, where the markings M2, M3 and M5 are not reachable.*
  - *The solution* $a1 = b1 = a2 = b2 = 4$ *and* $a3 = b3 = 1$ *preserves all, and only all, paths of PN1 that are free from bad markings.* ***It is then maximally permissive***

Parametric model checking approach issues

How can PSCG's state explosion problem be significantly mitigated?

- This proliferation of states mainly occurs because different interleavings of the same transitions typically produce distinct (parametric) state classes or firing domains
  - Propose mitigation strategies to eliminate unnecessary distinctions between firing orders that result in identical behavioral outcomes.
    - the number of explored state classes can be significantly decreased.

- **How to identify permissive conditions?**

- We seek solutions that are maximally permissive, imposing the fewest possible restrictions on the model's behavior.
  - What conditions ensure that only undesirable transitions are prevented from firing, so that all acceptable behaviors remain allowed?
    - use inequalities and parameters to describe all possible states reachable within specific timing constraints.

To address these challenges of the TBDP problem, we propose a symbolic approach.

Symbolic approach to deal with the TBDP problem

This approach consists of a depth-first search (DFS) of a **symbolic reachability graph (SRG) of the input PN.**

- Two main advantages of exploring SRG of a given PN instead of SCG (parametric) of its extended TPN (parametric)
  - Effective mitigation of PSCG state explosion:
    - Unlike the State Class Graph (SCG), the Symbolic Reachability Graph (SRG) groups all interleavings of the same transitions into a single marking.
      - This greatly reduces redundant state exploration.
  - Elimination of time constraints tied to firing order
    - Unlike the SCG, the SRG eliminates unnecessary distinctions related to firing order.
      - This avoids costly manipulation of parametric time domains and makes the analysis of larger or more complex systems significantly more scalable.

Construction of the symbolic reachability graph (SRG)

The firing delay interval of each transition t_i is represented symbolically by $[a_i , b_i]$, where a_i and b_i are two non-negative integer parameters such that $a_i \leq b_i$.

The SRG is computed based on a **DFS exploration** (i.e., exploration path-by-path).

During the exploration of the SRG, the kind of each marking M is determined and recorded : **bad, dangerous or legal**

- We denote by **parent** of an enabled transition *t* , the fired transition that has enabled *t*.

Some upper bounds of the distances between the firing dates of transitions enabled at **M** are computed and saved as linear combinations over the set of parameters Pr and denoted by **Δ :**

A symbolic marking is defined as a couple $\alpha = (M, \Delta)$

The initial symbolic marking is $\alpha_0 = (M_0, \Delta_0)$, where M_0 is the initial marking and

for all t, t' $\in En(M_0)$, $\Delta_0(t, t') = b - a'$

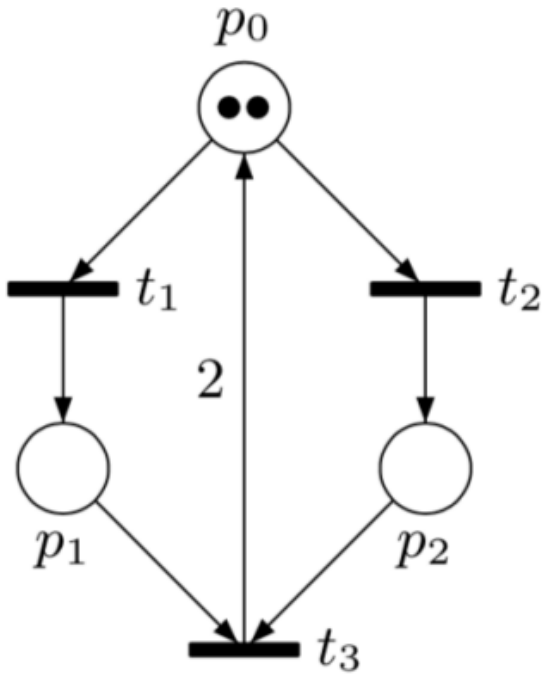
Construction of the symbolic reachability graph (SRG)

- Starting from the initial symbolic marking $\alpha_0 = (M_0, \Delta_0)$, , the successor symbolic markings are computed using the following firing rule:
- Let $\alpha = (M, \Delta)$ be a symbolic marking and t_f a transition firable from α
  - **its successor symbolic marking is $\alpha' = (M', \Delta')$ such that** :

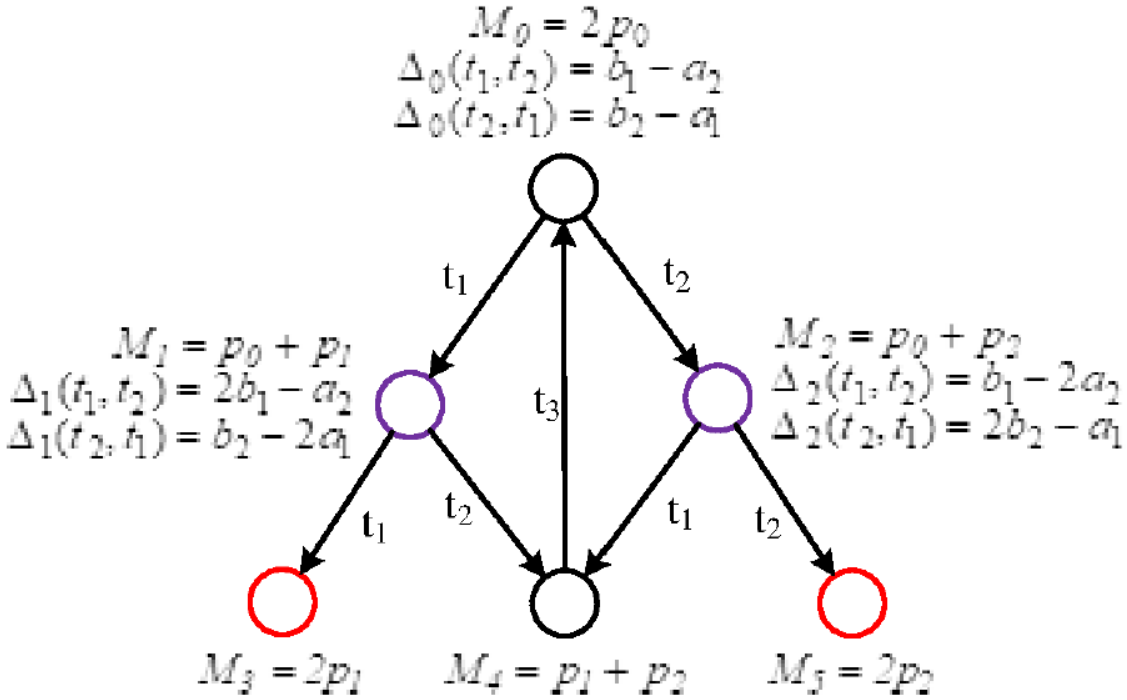
$$M' = M - pre(t_f) + post(t_f) \text{ and}$$
$$\forall t, t' \in En(M'),\ \Delta'(t, t') =$$
$$\begin{cases} b - a' & \text{if } t, t' \in Nw(M, t_f) \\ b + \Delta(t_f, t') & \text{if } t \in Nw(M, t_f) \wedge t' \notin Nw(M, t_f) \\ \Delta(t, t_f) - a' & \text{if } t \notin Nw(M, t_f) \wedge t' \in Nw(M, t_f) \\ \Delta(t, t') & \text{otherwise} \end{cases}$$

- Note that t_f is the parent of all transitions within $Nw(M, t_f)$ and $\Delta'(t, t')$ is an upper bound of the distance between the firing dates of t and t ' :
  - Therefore **$\Delta(t, t') < 0 \Rightarrow$ (t ' cannot occur before t from α)**.
  - **$\Delta(t, t') < 0$** a sufficient condition that ensures that t ' cannot occur before t from α in the parametric TPN
  - By contraposition **(t ' can occur before t from α) $\Rightarrow \Delta(t, t') \geq 0$**
  - $\Delta(t, t') \geq 0$ is a necessary condition of t ' can occur before t from α

Construction of the symbolic reachability graph (SRG)



PN1



SRG of PN1

It consists of six symbolic markings

The markings M_3 and M_5 are bad.

The markings M_1 and M_2 are dangerous.

M_0 and M_4 are **legal**.

(its PSCG contains 20 state classes)

SRG analysis : Deadlock preventing conditions

- **A transition t** enabled at a **dangerous marking M** is **said to be bad** iff the **successor marking of M by t is a bad marking**.
  - The marking **M** has a non-empty set of bad output transitions, denoted by ***O*(M)**, otherwise, it is legal

- Let a dangerous marking **M** of a symbolic marking **α = (M, Δ)**

  - The bad output transitions of M cannot occur in case for each bad transition $\mathbf{t_o}$ of *O*(M), there is, at least, a non bad transition $\mathbf{t_s}$ in *E*n(M) – *O*(M) that must be fired strictly before $\mathbf{t_o}$ from **α.**

  - **We denote by DPC(α) the deadlock preventing condition of α = (M, Δ) defined by:**

$$\bigwedge_{t_o \in \mathcal{O}(M)} \bigvee_{t_s \in \overline{\mathcal{O}(M)}} \Delta(t_s, t_o) < 0.$$

SRG analysis : Deadlock preventing conditions

- Given a PN, the preventing condition of a reachable dangerous marking M is the **conjunction of the DPCs of all reachable symbolic markings that share the same marking M**.

- If the **conjunction of all the DPCs of the dangerous markings is consistent**, then the TBDP has, at least, a solution and **this conjunction characterizes a set of solutions for the TBDP problem.**

- Note that the deadlock preventing condition of a dangerous marking is sufficient but not necessary to prevent reaching the deadlock markings.

SRG analysis of PN1 example

- There are, in the **dangerous marking M_1** of α_1, two enabled transitions t_1 and t_2 but **only t_1 is bad**.
  - The **deadlock preventing condition** of α_1 is then $\mathbf{\Delta_1(t_2, t_1) < 0}$, which is equivalent to $\mathbf{b_2 < 2a_1}$. It guaranties that, in α_1, the firing date of t_1 is strictly larger than the firing date of t_2.

- Similarly, in the **dangerous marking M_2** of α_2, among the two enabled transitions t_1 and t_2, only t_2 is bad.
  - The deadlock preventing condition of α_2 is then $\mathbf{\Delta_2(t_1, t_2) < 0}$, which is equivalent to $\mathbf{b_1 < 2a_2}$.

- **The conjunction of the deadlock preventing conditions of α_1 and α_2** (i.e., $b_1 < 2a_2 \wedge b_2 < 2a_1$) with the basic constraints (i.e., $0 \leq a_1 \leq b_1 \wedge 0 \leq a_2 \leq b_2 \wedge 0 \leq a_3 \leq b_3$) **characterizes a set of deadlock-free TPN extensions of PN1**.

SRG analysis : Permissive conditions

- The SRG can provide different solutions for the TBDP problem that impose different restrictions on the behaviour of its input PN model.

- Among these solutions, we are interested in those yielding less restriction in the behaviour of the input model (i.e., more permissive solutions).

- **In this sense, we introduce the permissive conditions which** are necessary conditions for the firability of all the enabled transitions of the legal symbolic markings

SRG analysis : Permissive conditions

- Let $\alpha = (M, \Delta)$ be a symbolic marking and $t_i , t_j \in En(M)$ two transitions enabled at M.
- We have shown that **$\Delta(t_i , t_j) < 0$ implies that t_j cannot fire before t_i from α.**
- The contraposition of this implication allows us to reach a conclusion:
  - that **t_j is firable before t_i from α implies $\Delta(t_i , t_j) \geq 0$.**
- A necessary condition for the firability of t_j before t_i from α is then $\Delta(t_i , t_j) \geq 0$.
- The **permissive condition of α, denoted by PC(α),** is defined by:

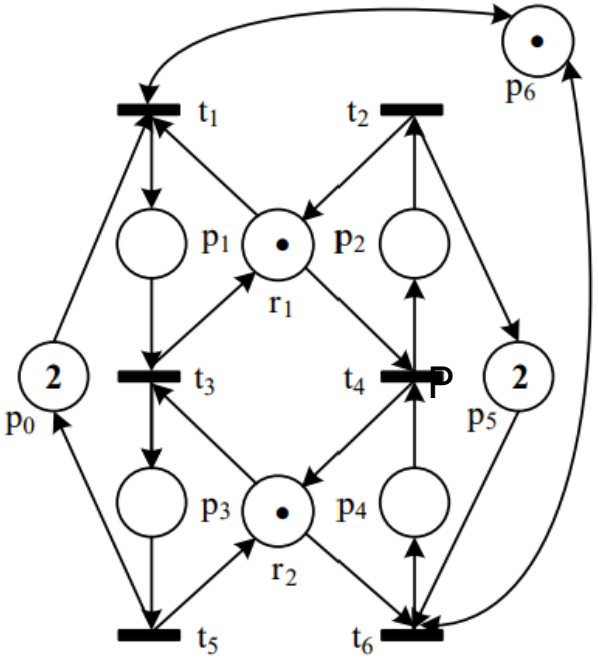
$$\bigwedge_{t_i, t_j \in En(M)} \Delta(t_i, t_j) \geq 0.$$

It is a necessary condition for the firability of all transitions enabled at M.

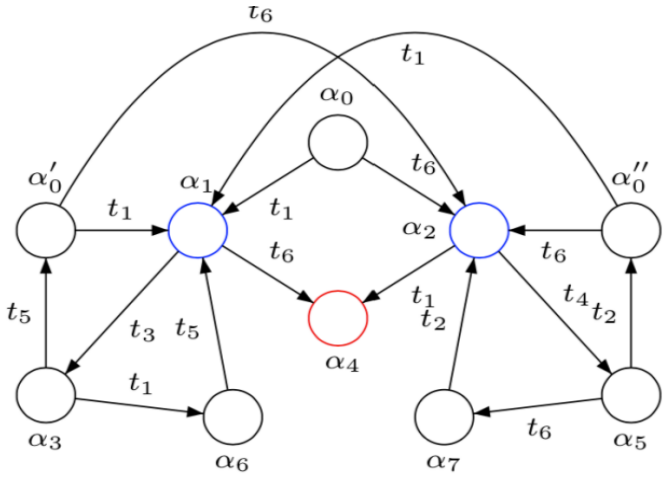
Adding the PCs of the legal symbolic markings to the DPCs of the dangerous symbolic markings will discard some less permissive solutions for the TBDP problem.

However, all the kept solutions are not necessarily maximally permissive.

SRG analysis of PN2 example



PN2



$\alpha_0 : M_0 = 2p_0 + 2p_5 + r_1 + r_2 + p_6,$
$\Delta_0(t_1, t_6) = b_1 - a_6, \Delta_0(t_6, t_1) = b_6 - a_1.$
$\alpha_1 : M_1 = p_0 + p_1 + 2p_5 + r_2 + p_6,$
$\Delta_1(t_3, t_6) = b_3 - a_6, \Delta_1(t_6, t_3) = b_6 - a_3.$
$\alpha_2 : M_2 = 2p_0 + p_4 + p_5 + r_1 + p_6,$
$\Delta_2(t_1, t_4) = b_1 - a_4, \Delta_2(t_4, t_1) = b_4 - a_1.$
$\alpha_3 : M_3 = p_0 + p_3 + 2p_5 + r_1 + p_6,$
$\Delta_3(t_1, t_5) = b_1 - a_5, \Delta_3(t_5, t_1) = b_5 - a_1.$
$\alpha_4 : M_4 = p_0 + p_1 + p_4 + p_5 + p_6.$
$\alpha_5 : M_5 = 2p_0 + p_2 + p_5 + r_2 + p_6,$
$\Delta_5(t_2, t_6) = b_2 - a_6, \Delta_5(t_6, t_2) = b_6 - a_2.$
$\alpha_6 : M_6 = p_1 + p_3 + 2p_5 + p_6.$
$\alpha_7 : M_7 = 2p_0 + p_2 + p_4 + p_6.$
$\alpha'_0 : M'_0 = M_0 = 2p_0 + 2p_5 + r_1 + r_2 + p_6.$
$\Delta'_0(t_1, t_6) = b_1 - a_5 - a_6, \Delta'_0(t_6, t_1) = b_6 + b_5 - a_1.$
$\alpha''_0 : M''_0 = M_0 = 2p_0 + 2p_5 + r_1 + r_2 + p_6.$
$\Delta''_0(t_1, t_6) = b_1 + b_2 - a_6, \Delta''_0(t_6, t_1) = b_6 - a_1 - a_2.$

SRG of PN2

SRG analysis of PN2 example

Table 2
DPC of the dangerous markings of PN2

Dangerous M	$En(M)$	$\mathcal{O}$	$\Delta(t,t')$	DPC
M_1	$\{t_3,t_6\}$	$\{t_6\}$	$\Delta_1(t_3,t_6)=b_3-a_6$	$b_3<a_6$
M_2	$\{t_1,t_4\}$	$\{t_1\}$	$\Delta_2(t_4,t_1)=b_4-a_1$	$b_4<a_1$

Table 3
PC of the legal markings of PN2

Legal M	$\Delta(t,t')$	PC
M_0	$\Delta_0(t_6,t_1)=b_6-a_1$	$a_1\leq b_6$
	$\Delta_0(t_1,t_6)=b_1-a_6$	$\wedge\ a_6\leq b_1$
$M_0'=M_0$	$\Delta_0'(t_1,t_6)=b_1-a_5-a_6$	$a_5+a_6\leq b_1$
	$\Delta_0'(t_6,t_1)=b_5+b_6-a_1$	$\wedge\ a_1\leq b_6+b_5$
$M_0''=M_0$	$\Delta_0''(t_1,t_6)=b_1+b_2-a_6$	$a_1+a_2\leq b_6$
	$\Delta_0''(t_6,t_1)=b_6-a_1-a_2$	$\wedge\ a_6\leq b_1+b_2$
M_3	$\Delta_3(t_1,t_5)=b_1-a_5$	$a_1\leq b_5$
	$\Delta_3(t_5,t_1)=b_5-a_1$	$\wedge\ a_5\leq b_1$
M_5	$\Delta_5(t_2,t_6)=b_2-a_6$	$a_2\leq b_6$
	$\Delta_5(t_6,t_2)=b_6-a_2$	$\wedge\ a_6\leq b_2$

The conjunction of the deadlock preventing conditions with the permissive conditions is consistent and provides for this example **a set of maximally permissive solutions for the TBDP problem.**

This set is defined by :

$0\leq a_i\leq b_i$

$b_3<a_6\leq b_2\wedge b_4<a_1\leq b_5\wedge a_5+a_6\leq b_1\wedge a_1+a_2\leq b_6$.

Handling repetitive sequences

- The finiteness of the SRG is not guaranteed even for bounded PN
- Indeed, the SRG is infinite for any bounded Petri net that has at least one marking M with a **repetitive firing sequence** ω keeping one transition continuously enabled while another is alternately disabled and enabled.
  - **This situation generates an infinite number of distinct timing constraints along a loop-free path, leading to infinitely many symbolic states.**

- To guarantee the finiteness of the SRG, exploration of any infinite loop-free path (e.g., $\alpha_0\omega_0\alpha_1\omega\alpha_2\omega\alpha_3\omega\alpha_4 \ldots$) **is restricted to a finite prefix**, such as $\alpha_0\omega_0\alpha_1\omega\alpha_2\omega\alpha_3$.
  - By using this over-approximation of symbolic markings, the length of every loop-free path in the resulting SRG is bounded.
    - **This ensures a manageable, finite state space for analysis, even when the original SRG might be infinite.**

TPN with controllable/uncontrollable transitions

- The two proposed approaches (**parametric model checking and the SRG approach) can be easily extended to address the TBDP problem where a TPN with** controlled/uncontrolled transitions is used as the input model.

  - In such a context, the TBDP problem consists in deciding whether the static firing intervals of the input TPN's controllable transitions can be restricted so as to enforce deadlock-freeness.

  - With parametric model checking approach, the input model is a parametric TPN, where the static **firing intervals of uncontrollable transitions are fixed, while the others are parametric intervals.**

  - With the SRG approach, the deadlock prevention conditions are calculated by taking into account the **exact values of the bounds of the firing intervals of uncontrollable transitions**, and adding, for each controllable transition t_i, the constraint $\downarrow Is(t_i) \leq a_i \leq b_i \leq \uparrow Is(t_i)$. Finally, the same process is applied for permissive conditions

Conclusion

- we study the problem of deciding whether or not there exists a deadlock-free TPN extension for a given bounded PN (TBDP problem).

  - We first formulate the TBDP problem as a **parametric model checking of a parametric TPN.**
    - We show that this problem is decidable but suffers from a severe state explosion problem caused mainly by firing order constraints.

  - In a second step, to cope with this state explosion problem, we proposed a **symbolic approach**, where the firing order constraints are abstracted to keep only the constraints between transitions and their parents.

Conclusion

- Compared with the existing untimed PN deadlock prevention methods, the main advantages of TBDP are:
  - **an important gain with respect to the cost of control places** which are implemented as cost devices while for TBDP, we use timers (less expensive).
  - we can exploit better parametrization of the TBDP method if time constraints are modified.
  - **The TBDP method is the same with or without uncontrollable transitions**
    - it is not the case with untimed PN deadlock prevention methods, which require modifications in the case of uncontrollable transitions
- **Limits**
  - TBDP methods don't automatically provide the most permissive solutions.
  - **SRG approach**
    - Deadlock preventing condition of a dangerous marking is sufficient but not necessary to prevent reaching the deadlock
    - The finiteness of the SRG is not guaranteed even for a bounded PN

Current and Future work

- We focus on the improvement of the SRG approach especially
  - by adapting and integrating the partial order reduction techniques developed for PN and TPN.
  - by investigation of a weaker deadlock preventing conditions
  - by computation of a finite over-approximation of the symbolic markings in the case of infinte SRG.

Thanks for your attention

kamel.barkaoui@cnam.fr

This work is published in Automatica Journal (vol. 137)

https://doi.org/10.1016/j.automatica.2021.110119