Réseaux de Petri T-temporisés pour la résolution des problèmes d'ordonnancement

Control design with T-TPN models for scheduling problems

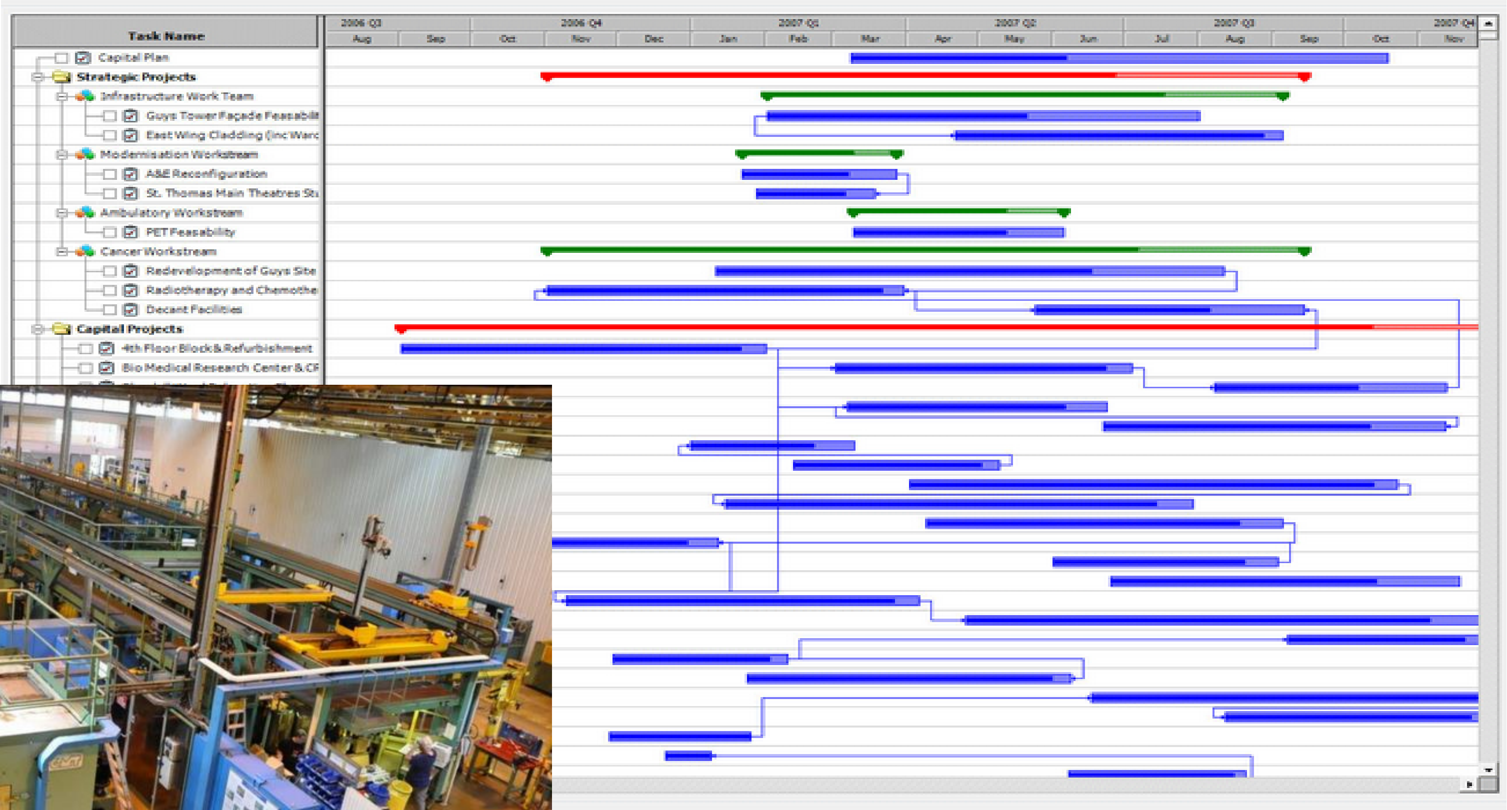
Dimitri Lefebvre
Normandie Université UNIHAVRE, GREAH, 76600 Le Havre, France.

RÉGION NORMANDIE
UNION EUROPEENNE
Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Outline

1. Problems and objectives
2. Modeling scheduling problems with T-TPN
3. Timed Extended Reachability Graph
4. Approximated Timed Extended Reachability Graph
5. Beam Search
6. Conclusion and future works

Scheduling problems



Routing and resource allocation

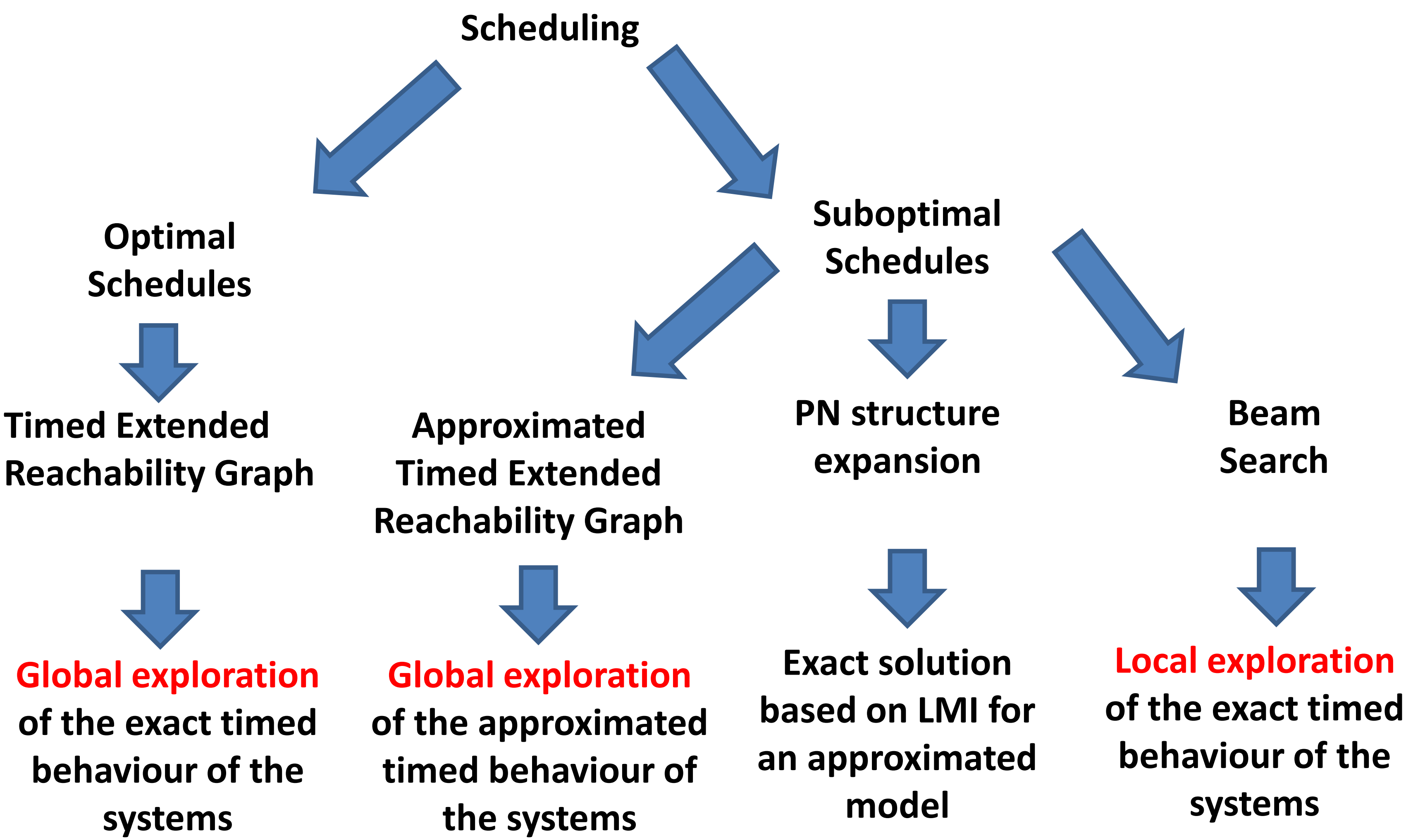
Problems and objectives

To compute control sequences that reach a reference state from an initial one for FMS in minimal time with :

- **Limited resources**
- **State constraints**
- **Temporal constraints**

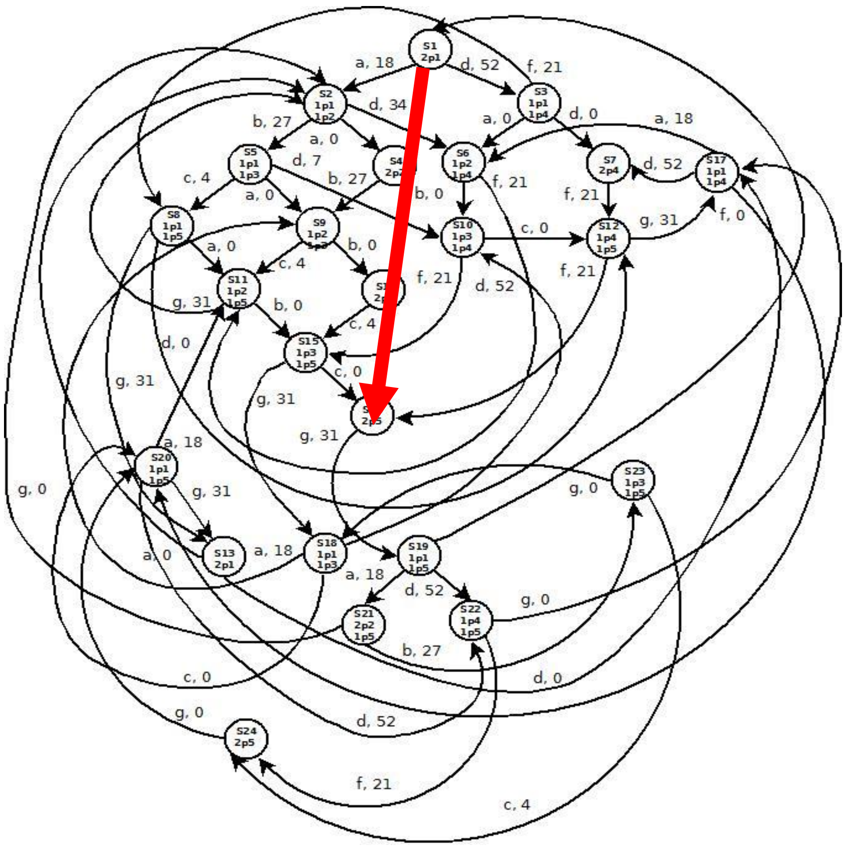
To compute firing sequences that reach a reference marking from an initial one for Transition – Timed Petri net models in minimal time with:

- **Structural constraints (precedence)**
- **Marking constraints (deadlocks and dead branches avoidance,…)**
- **Minimal firing duration specifications**

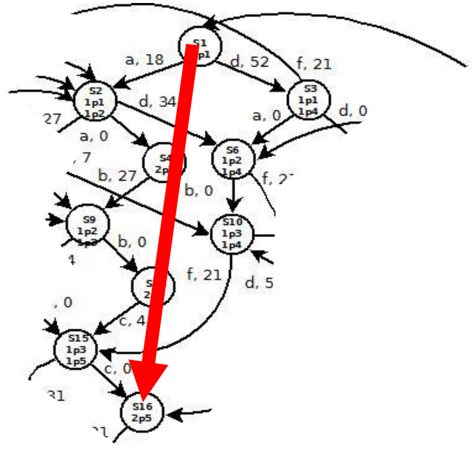

Scheduling
Optimal Schedules
Suboptimal Schedules
Timed Extended Reachability Graph
Approximated Timed Extended Reachability Graph
PN structure expansion
Beam Search
Global exploration of the exact timed behaviour of the systems
Global exploration of the approximated timed behaviour of the systems
Exact solution based on LMI for an approximated model
Local exploration of the exact timed behaviour of the systems

Scheduling problems

Global exploration



Local exploration



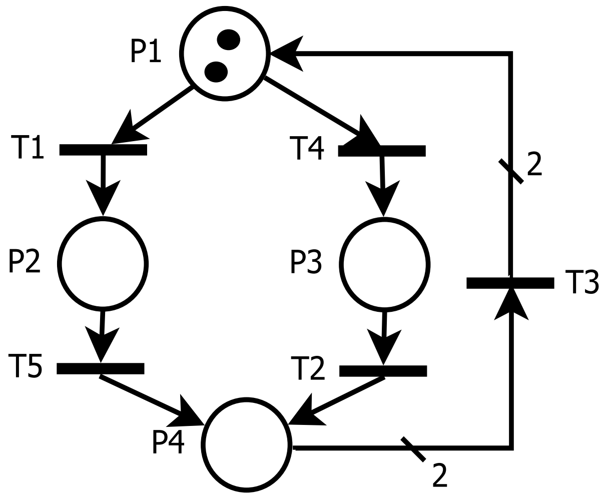
Outline

1. Problems and objectives
2. **Modeling scheduling problems with T-TPN**
3. Timed Extended Reachability Graph
4. Approximated Timed Extended Reachability Graph
5. Beam Search
6. Conclusion and future works

Petri nets

PN = <P, T, W, M$_I$>

- **P : set of places**
- **T : set of transitions**
- **W: model structure**
- **M$_I$: initial marking**



Incidence matrix

$$\mathbf{W_{PR}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \end{pmatrix}$$

$$\mathbf{W_{PO}} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{W} = \mathbf{W_{PO}} - \mathbf{W_{PR}} = \begin{pmatrix} -1 & 0 & 2 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & -2 & 0 & 1 \end{pmatrix}$$

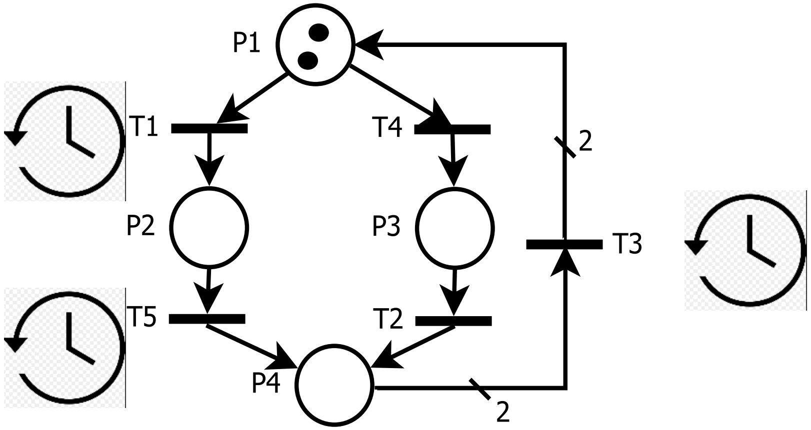
State and dynamics

$$\begin{matrix}\mathbf{P}1\\ \mathbf{P}2\\ \mathbf{P}3\\ \mathbf{P}4\end{matrix}\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \mathbf{W}. \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}\begin{matrix}\mathbf{T}1\\ \mathbf{T}2\\ \mathbf{T}3\\ \mathbf{T}4\\ \mathbf{T}5\end{matrix}$$

Transition -Timed Petri nets (T-TPN)

TPN = <P, T, W, d, M_I>

- **P : set of places**
- **T : set of transitions**
- **W: model structure**
- **M_I : initial marking**
- **$\delta : T \rightarrow R^+$: minimal firing duration for**
- **transition T**
- **$\Delta : T \rightarrow R^+$: maximal firing duration for transition T**



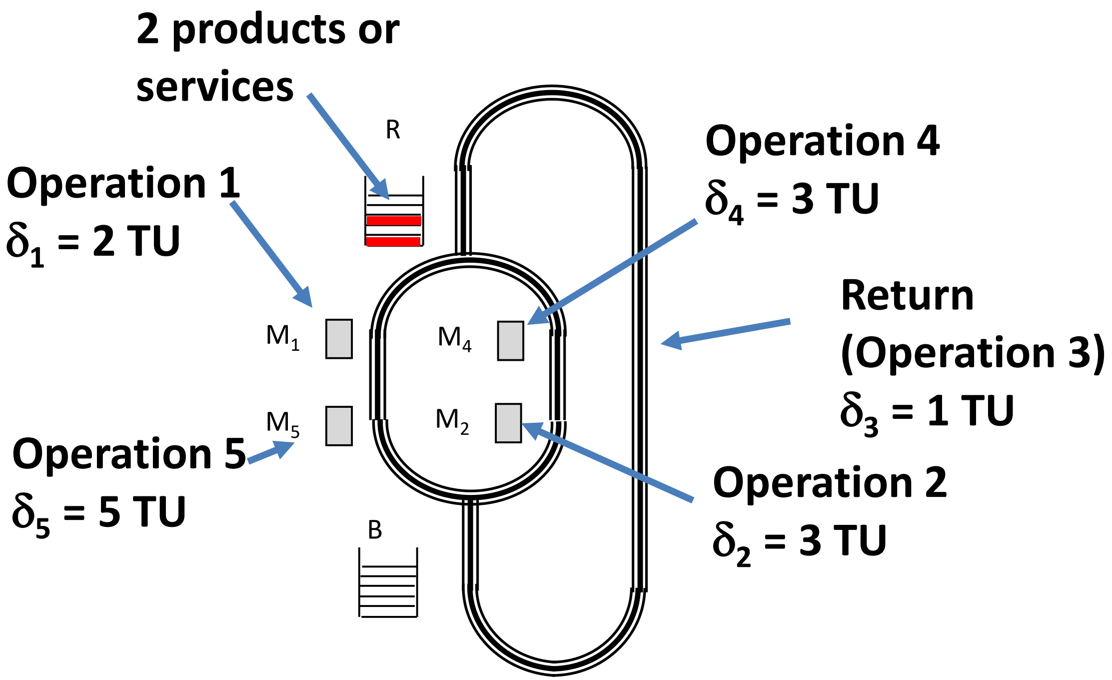
Choice policy : preselection policy

Server policy : infinite or *k*-server policy

Memory policy : enabling memory policy

The firings occurs according to the earliest firing policy

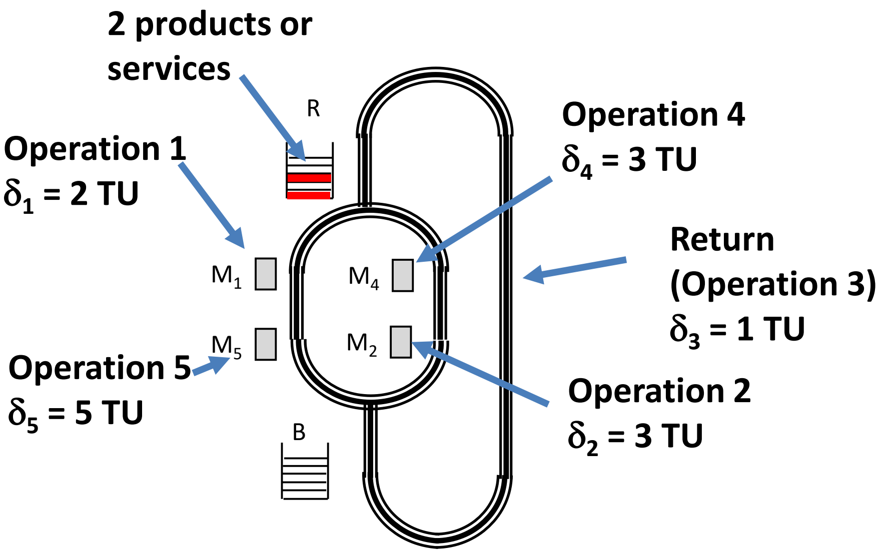
Modeling manufacturing systems with MATLAB



Définir le modèle T-TPN avec MATLAB

>> [Wpr,Wpo,delta,MI] = Exemple1_MACS

Modeling manufacturing systems with MATLAB



Définir le modèle T-TPN avec MATLAB
>> [Wpr,Wpo,delta,MI] = Exemple1_MACS

Wpr =

$$\begin{matrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \end{matrix}$$

Wpo =

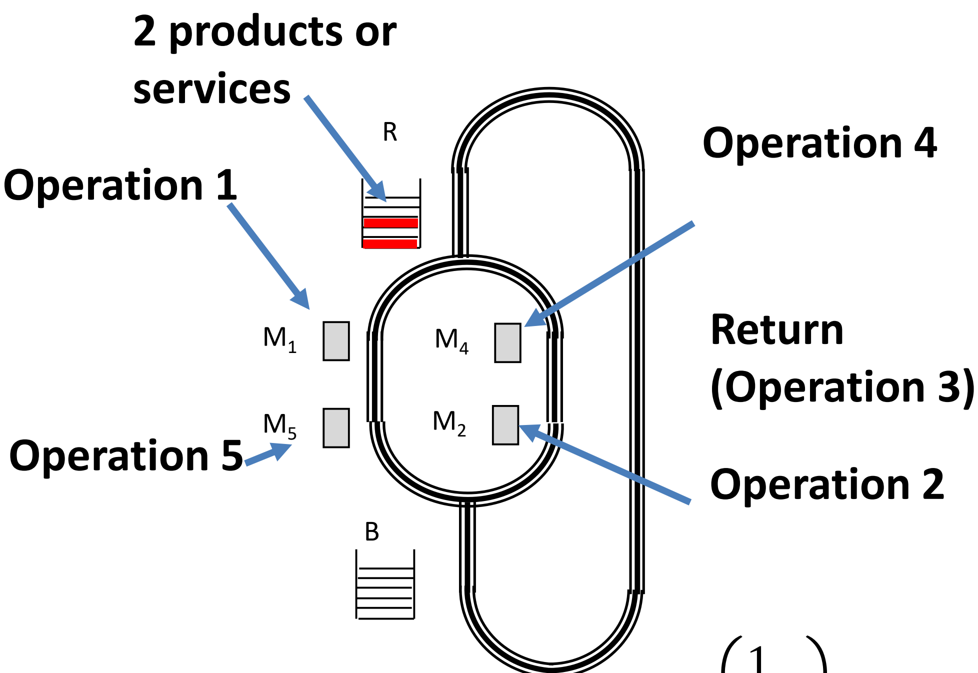
$$\begin{matrix} 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{matrix}$$

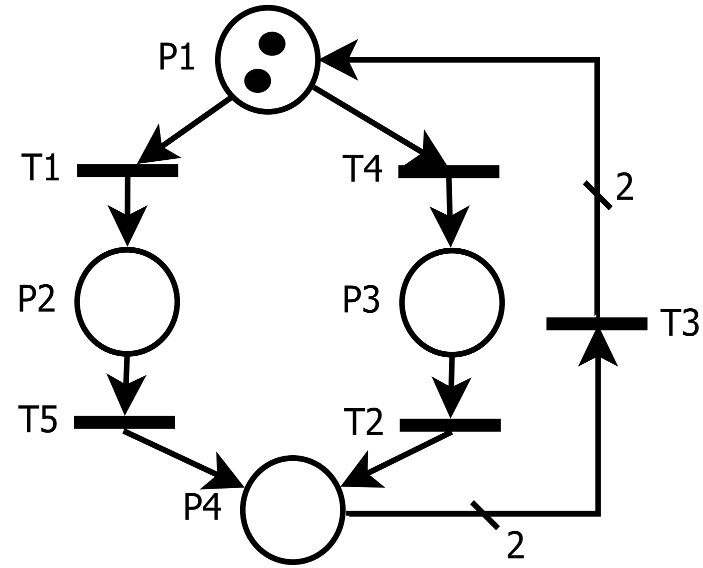
delta =

$$\begin{matrix} 2 \\ 3 \\ 1 \\ 3 \\ 5 \end{matrix}$$

MI =

$$\begin{matrix} 2 \\ 0 \\ 0 \\ 0 \end{matrix}$$





PN model

$$\mathbf{M_I} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Initial marking

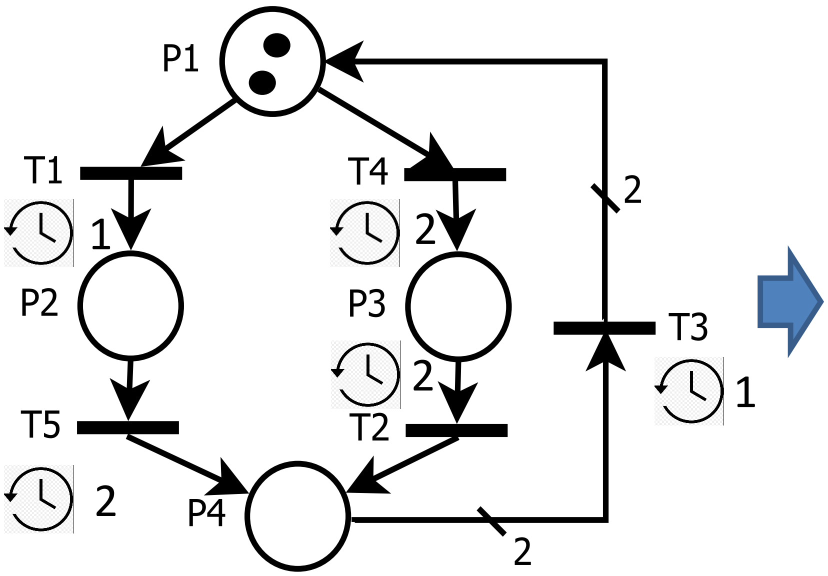
$$\delta = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$

Time parameters

$$\mathbf{W} = \begin{pmatrix} -1 & 0 & 2 & -1 & 0 \\ 1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & -2 & 0 & 1 \end{pmatrix}$$

Incidence matrix

T-TPN



Information about time

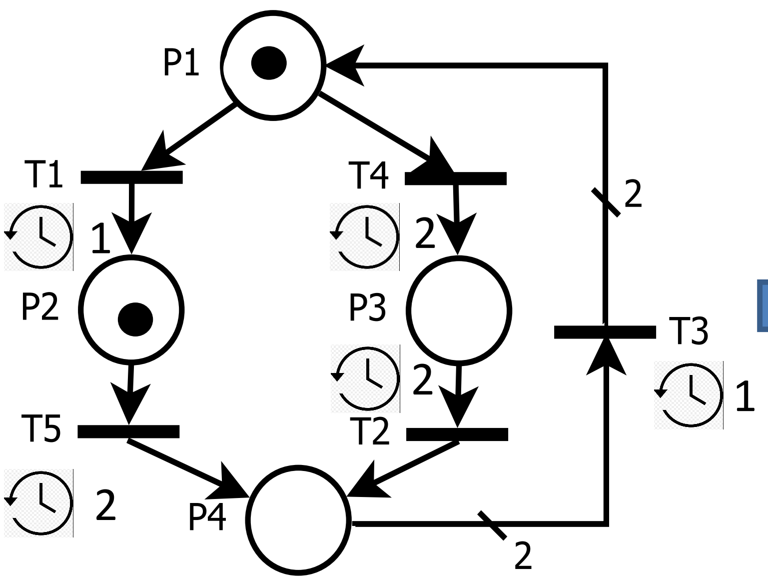
Agenda

Time: t = 0 TU

T1 : t+1 TU

T4 : t+2 TU

T-TPN



Information about time

$$\sigma = T_1(1)$$

Agenda

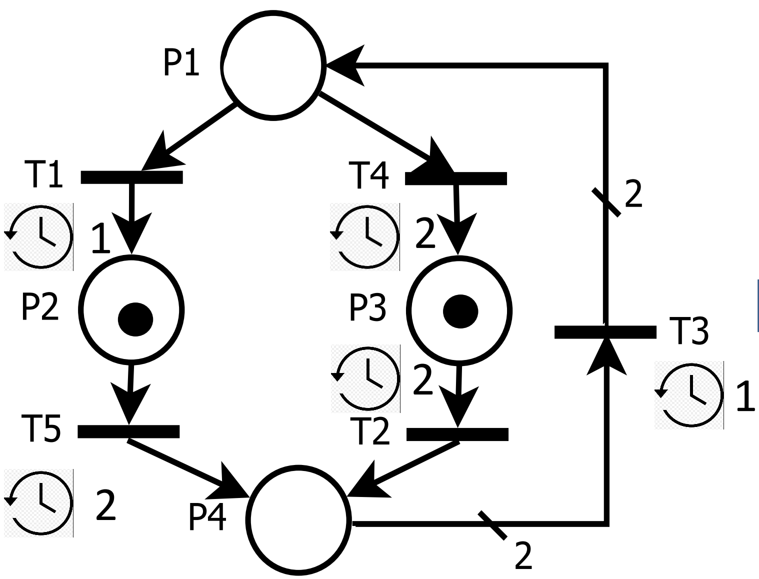
Time: t = 1 TU

T1 : t+0 TU

T4 : t+1 TU

T5 : t+2 TU

T-TPN



Information about time

$$\sigma = T_1(1)\ T_4(2)$$

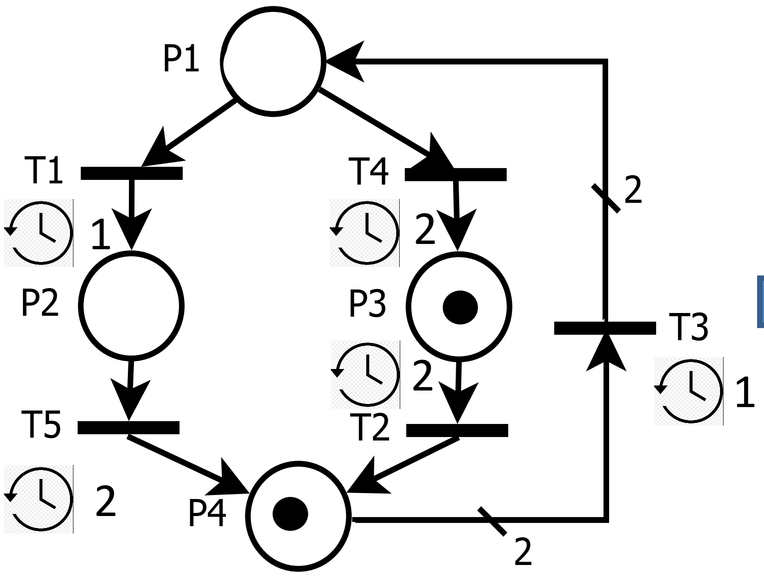
Agenda

Time: t = 2 TU

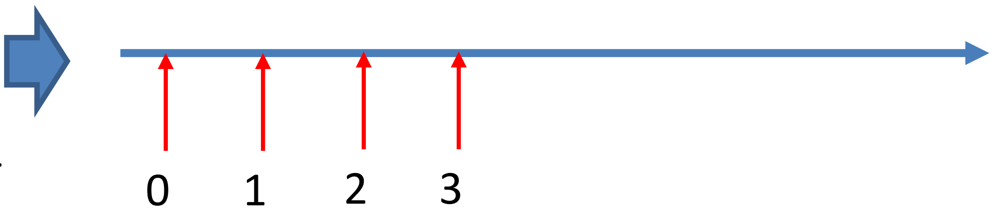
T5 : t+1 TU

T2 : t+2 TU

T-TPN



Information about time



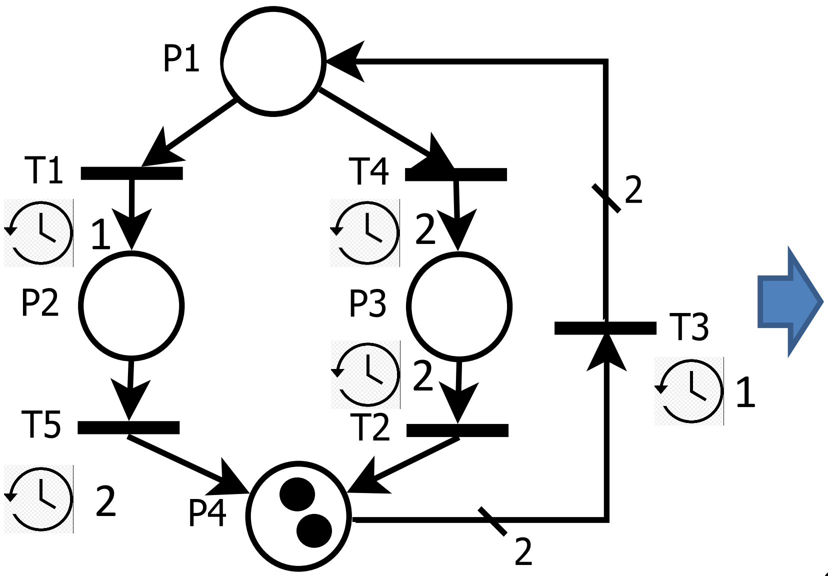
$$\sigma = T_1(1)\ T_4(2)\ T_5(3)$$

Agenda

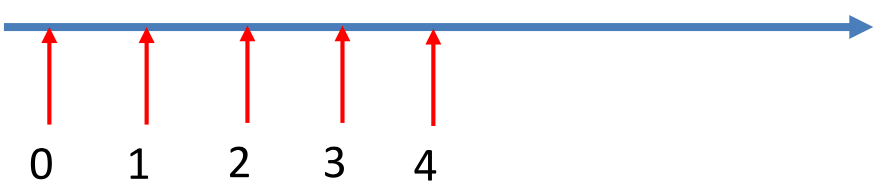
Time: t = 3 TU

T2 : t+1 TU

T-TPN



Information about time



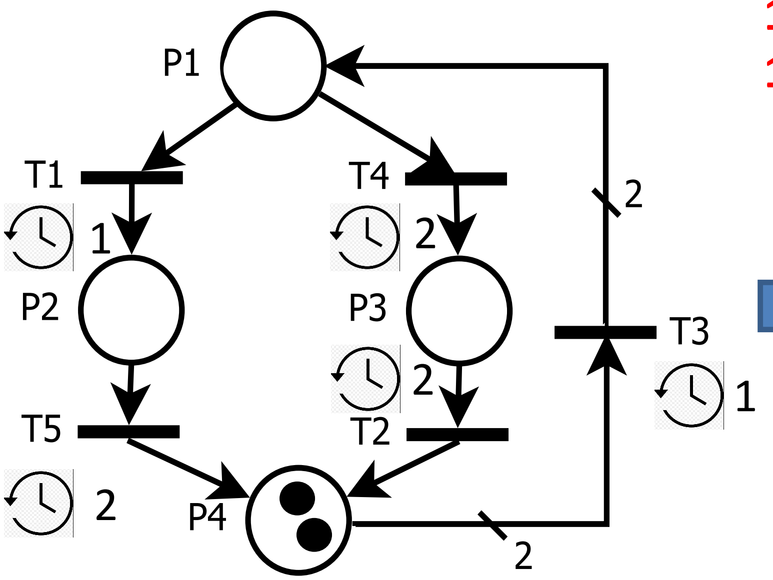
$$\sigma = T_1(1)\ T_4(2)\ T_5(3) T_2(4)$$

Agenda

Time: t = 4 TU

T3 : t+1 TU

T-TPN



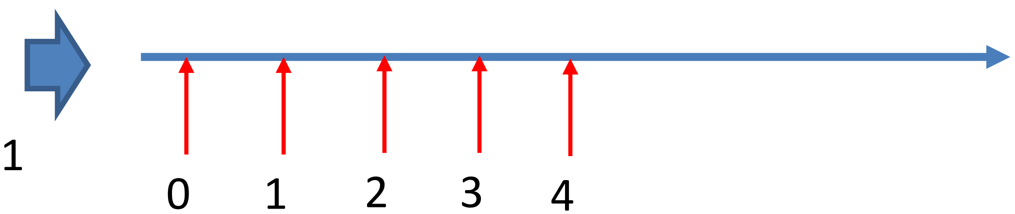
Dater une séquence avec MATLAB

>> seq =[1 4 5 2]

>>[timed_seq]=dateur_MACS(Wpr,Wpo,delta,seq,MI)

>>timed_seq =

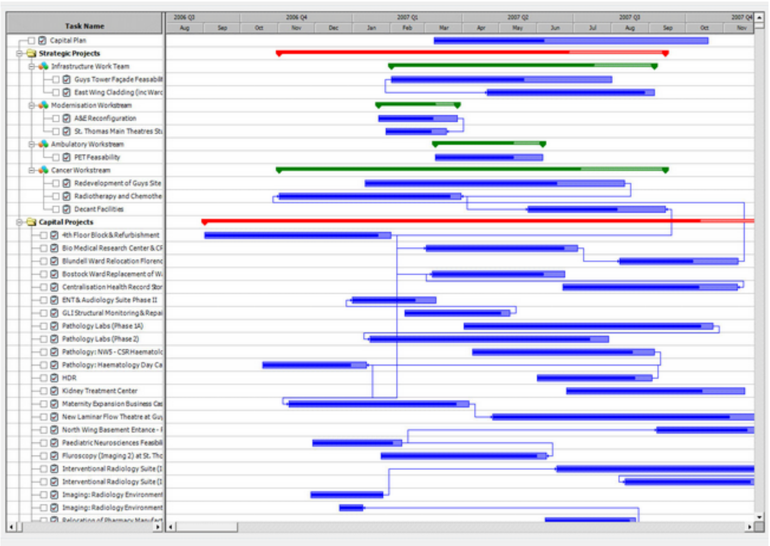
1	**4**	**5**	**2**
1	**2**	**3**	**4**



$$\sigma = T_1(1)\ T_4(2)\ T_5(3) T_2(4)$$

Modeling scheduling problem with T-TPN

Job Shop Scheduling problems



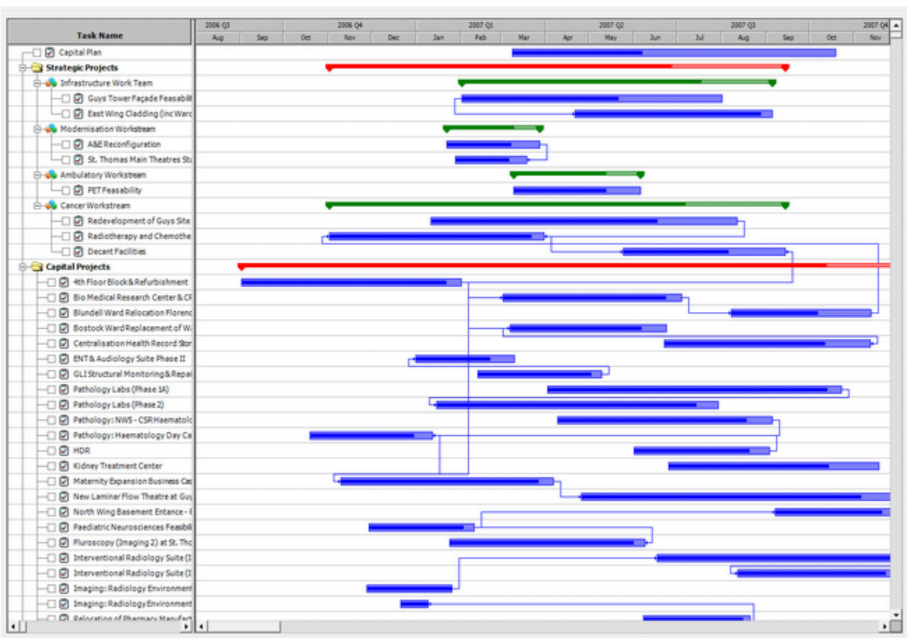
Problem specification

- Set of jobs J
- Each job J is composed by a set of operations o
- Each operation has a duration d and needs a set of resource R
- Resources are shared

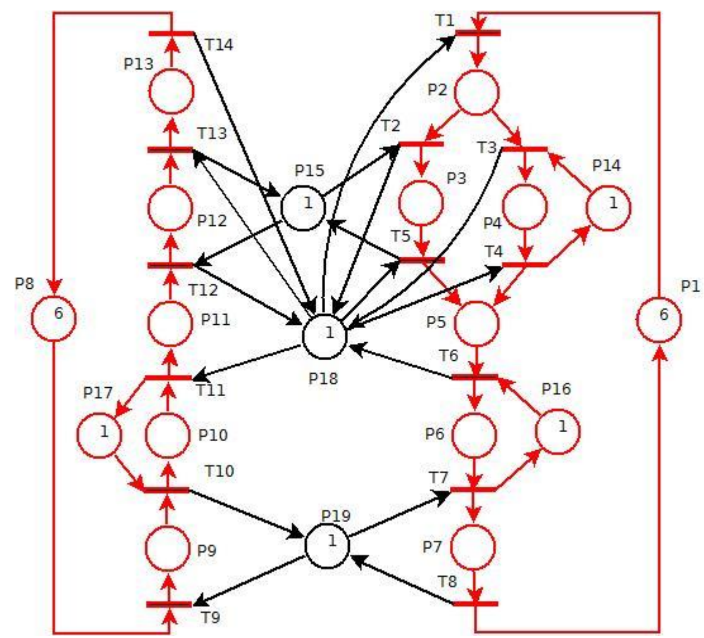
- Set of services to be completed according to the jobs
- Earch job can be single or multi server
- Each operation can be single or multi server

Modeling scheduling problem with T-TPN

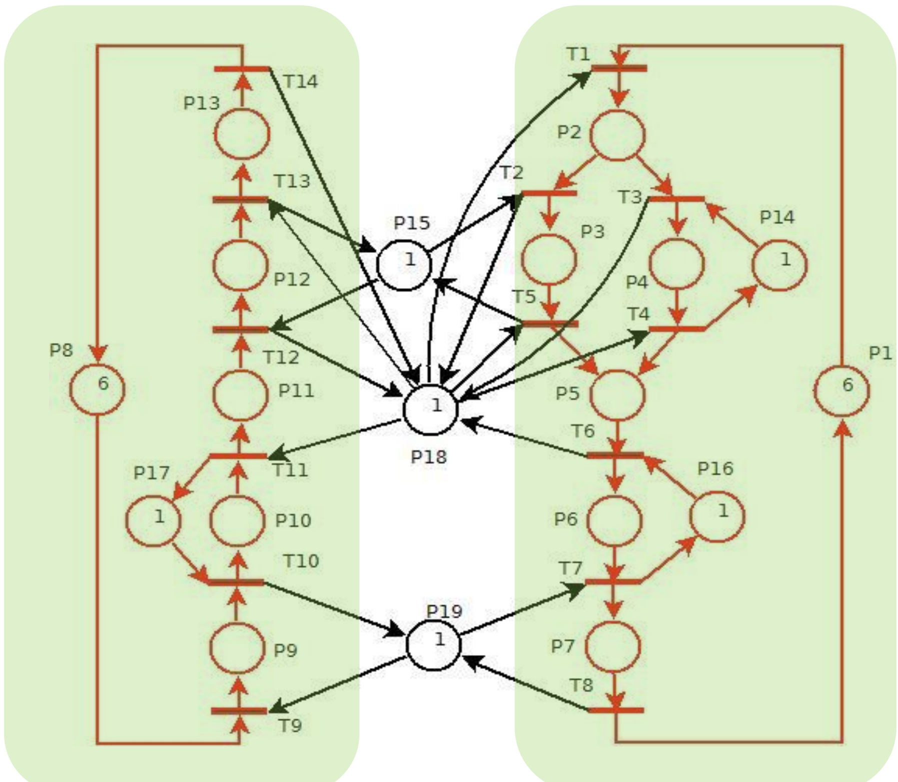
Job Shop Scheduling problems



Petri net modeling

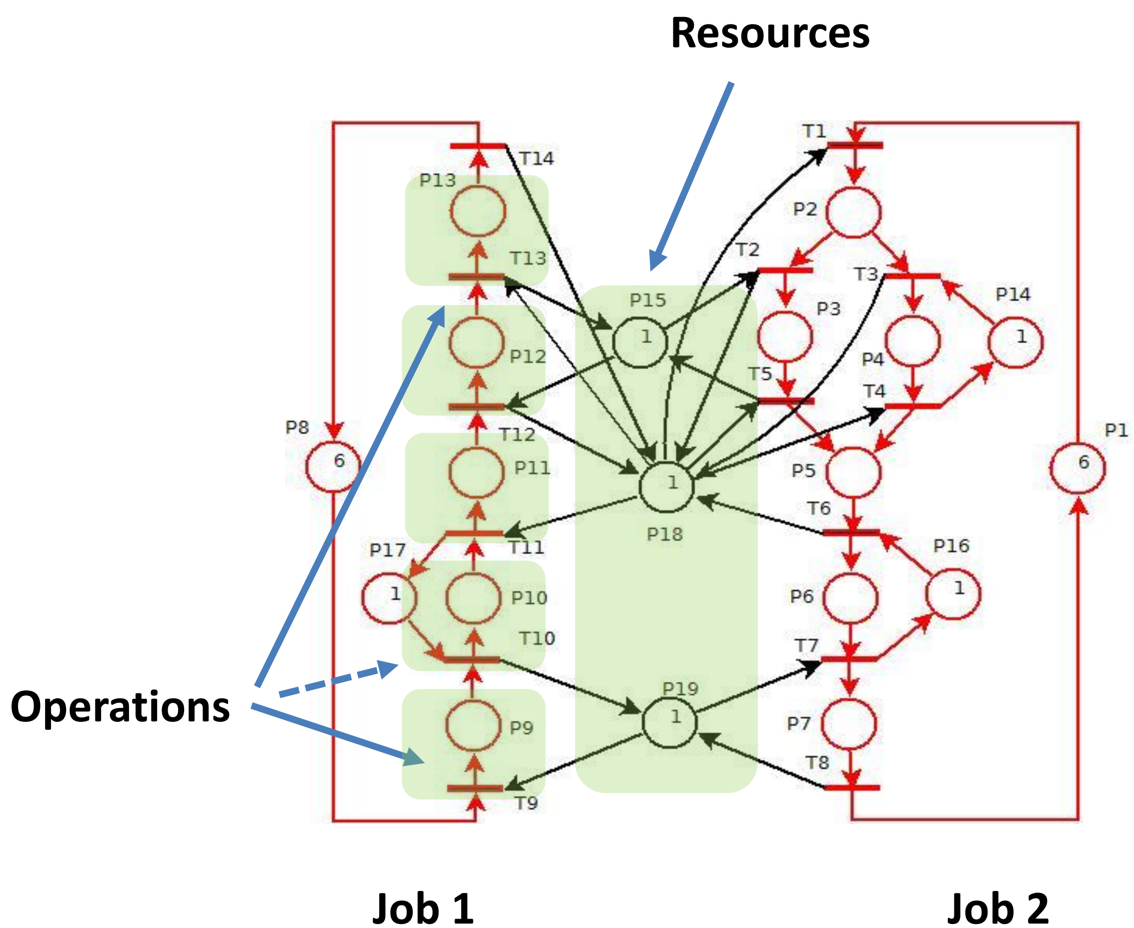


Scheduling problems



Job 1 Job 2

Scheduling problems



Exercice1

Construire le modèle T-TPN d'un problème d'ordonnancement avec:

2 jobs à réaliser chacun 1 fois
Capacité des jobs : 1
3 opérations o1, o2, o3 dans le job 1 de durée 5, 5 et 2
3 opérations o4, o5, o6 dans le job 2 de durée 5, 2 et 3
2 ressources partagée : pr1 partagée par o1, o2 et o5 et pr2 partagée par o2, o4 et o5

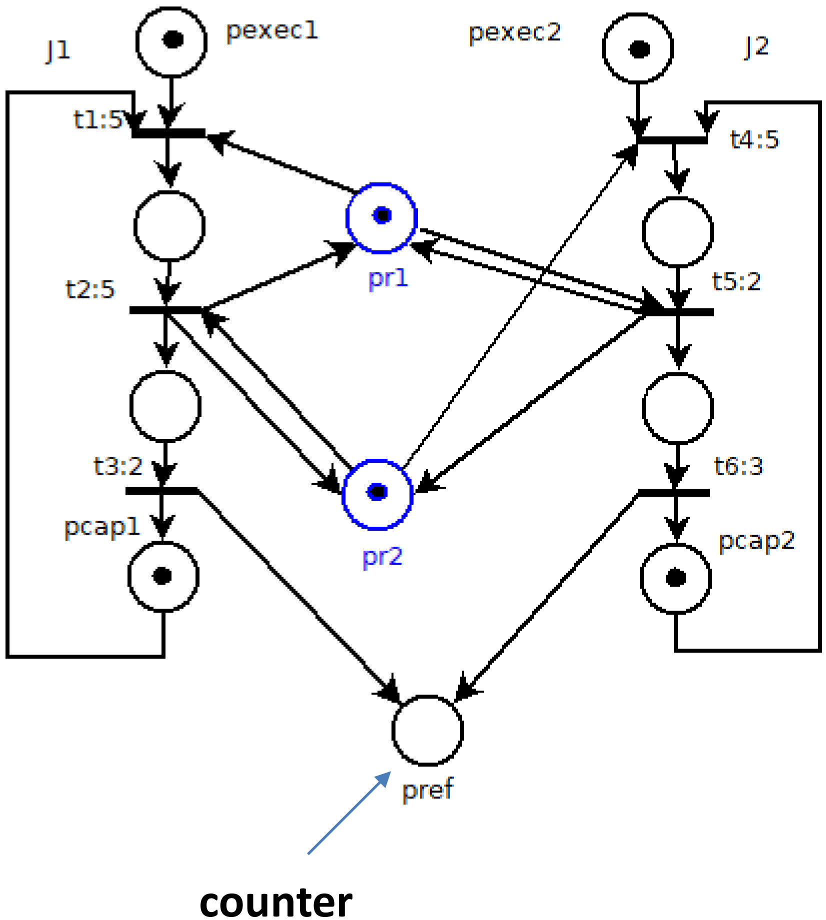
=> W_{PR}, W_{PO}, δ, M_I ???

Exercice1

Construire le modèle T-TPN d'un problème d'ordonnancement avec:

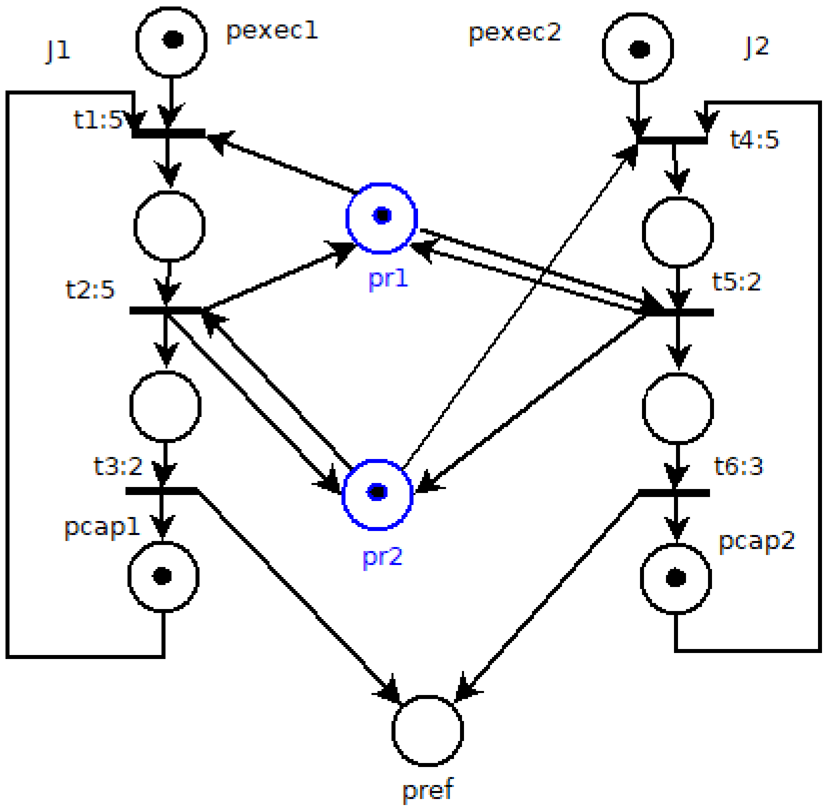
2 jobs à réaliser chacun 1 fois
Capacité des jobs : 1
3 opérations o1, o2, o3 dans le job 1 de durée 5, 5 et 2
3 opérations o4, o5, o6 dans le job 2 de durée 5, 2 et 3
2 ressources partagée : pr1 partagée par o1, o2 et o5 et pr2 partagée par o2, o4 et o5

=> W_{PR}, W_{PO}, δ, M_I ???



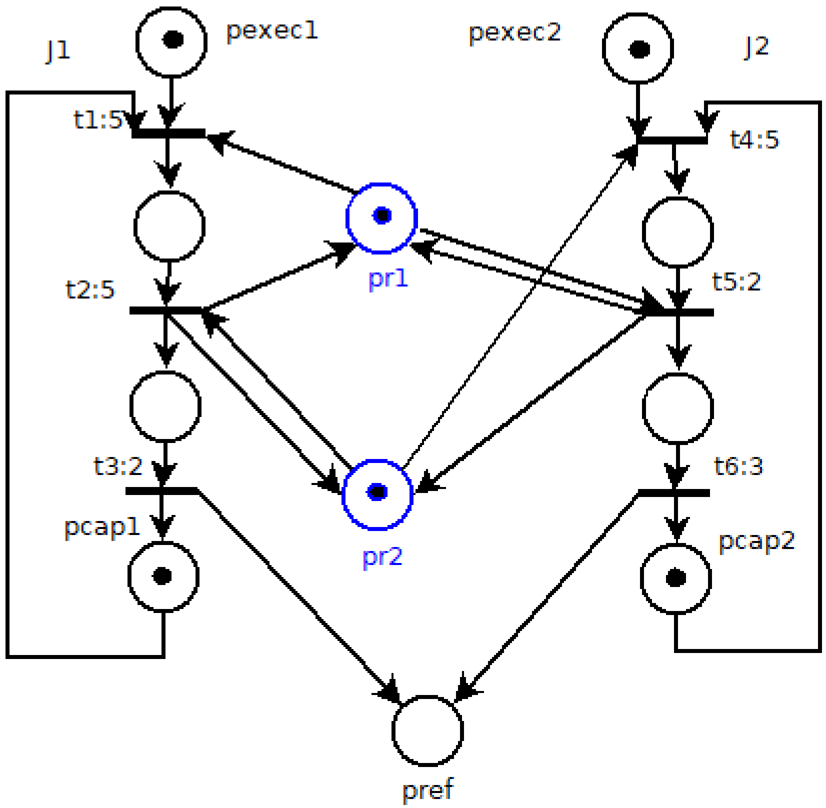
MATLAB

```
>> [Wpr,Wpo,delta,MI,Pjob] = Exemple2_MACS(1,1)

       %1 2 3 4 5 6
 Wpr=[ 1 0 0 0 0 0; %1
       1 0 0 0 0 0; %2
       0 1 0 0 0 0; %3
       0 0 1 0 0 0; %4
       0 0 0 1 0 0; %5
       0 0 0 1 0 0; %6
       0 0 0 0 1 0; %7
       0 0 0 0 0 1; %8
       1 0 0 0 1 0; %9
       0 1 0 1 0 0; %10
       0 0 0 0 0 0];%11
```



MATLAB

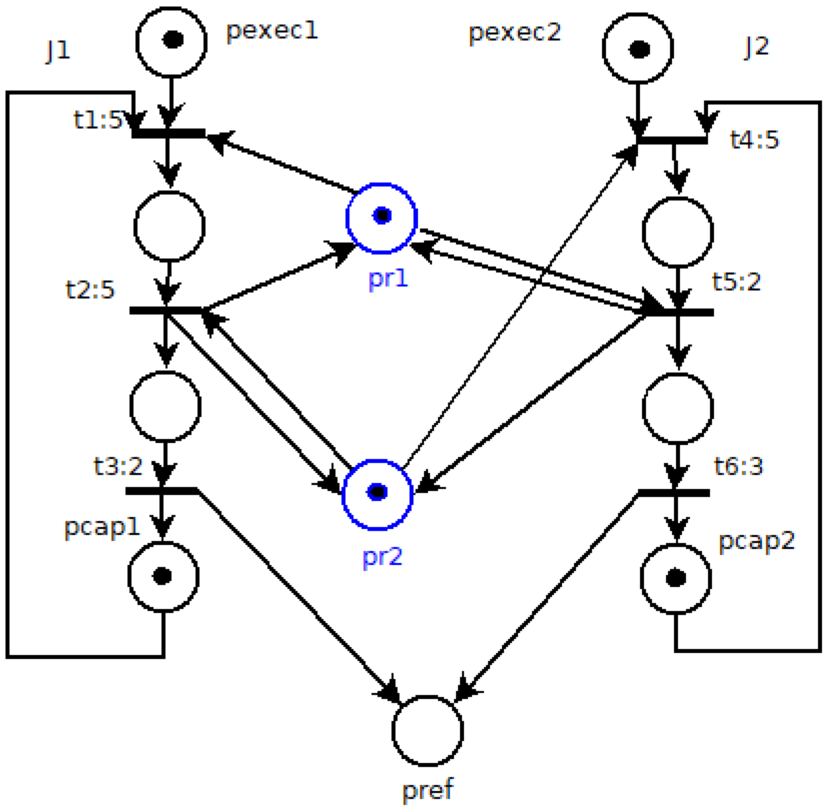
```
>> [Wpr,Wpo,delta,MI,Pjob] = Exemple2_MACS(1,1)

          %1 2 3 4 5 6
   Wpo=[ 0 0 0 0 0 0; %1
         0 0 1 0 0 0; %2
         1 0 0 0 0 0; %3
         0 1 0 0 0 0; %4
         0 0 0 0 0 0; %5
         0 0 0 0 0 1; %6
         0 0 0 1 0 0; %7
         0 0 0 0 1 0; %8
         0 1 0 0 1 0; %9
         0 1 0 0 1 0; %10
         0 0 1 0 0 1];%11
```



MATLAB

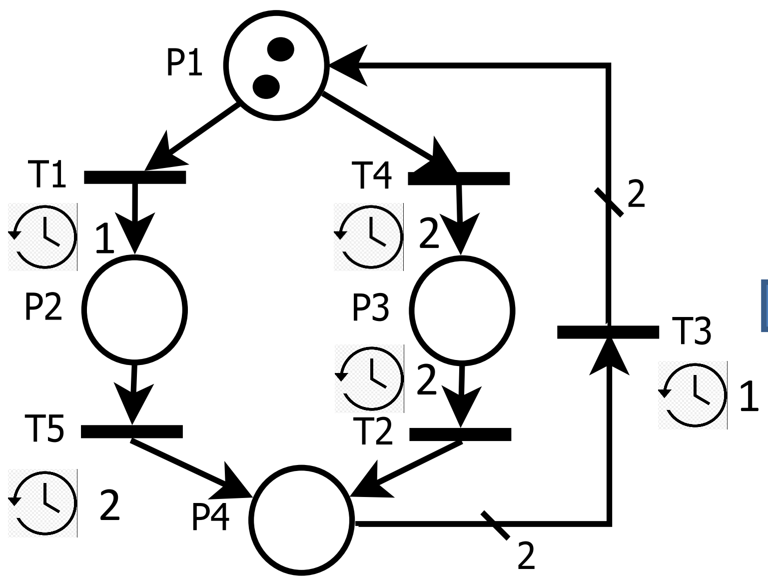
```
>> [Wpr,Wpo,delta,MI,Pjob] = Exemple2_MACS(1,1)
```

```
Pjob{1}.exec=1;
Pjob{1}.cap=2;
Pjob{1}.valcap=1;
Pjob{1}.job=[3 4];
Pjob{1}.ref=11;
Pjob{1}.Tc=[1:3];
Pjob{2}.exec=5;
Pjob{2}.cap=6;
Pjob{2}.valcap=1;
Pjob{2}.job=[7 8];
Pjob{2}.ref=11;
Pjob{2}.Tc=[4:6];
```

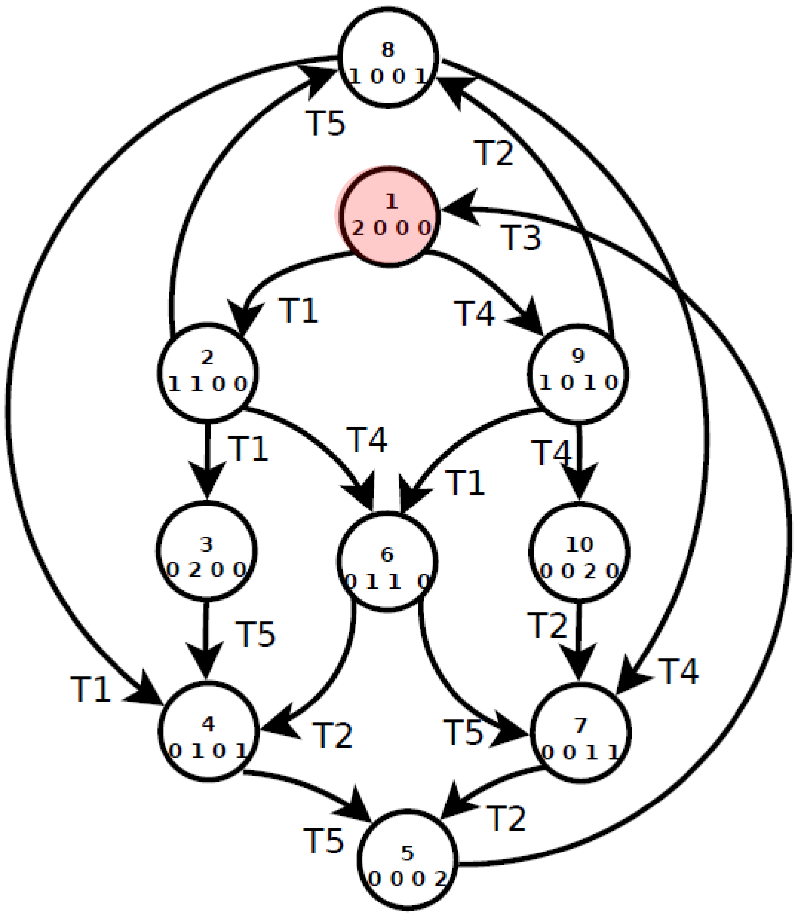



```
delta= [5  5   2   5   2   3]';
MI =   [1 1 0 0 1 1 0 0 1 1 0]';
```

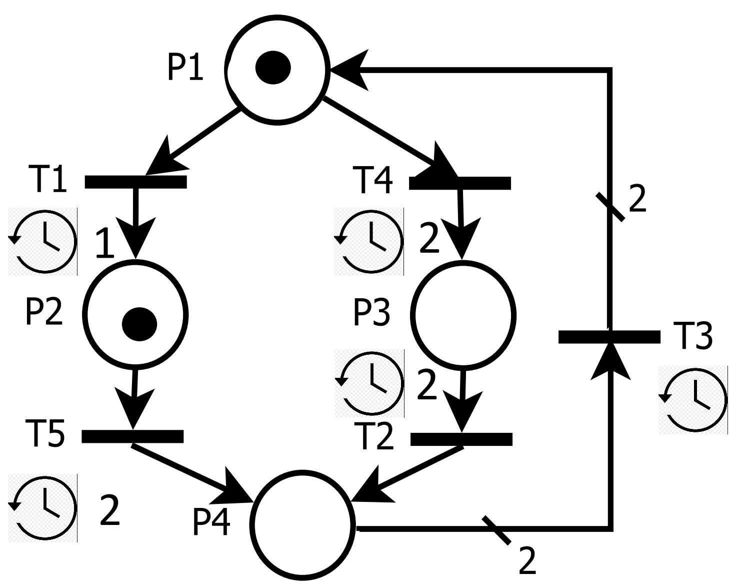
T-TPN



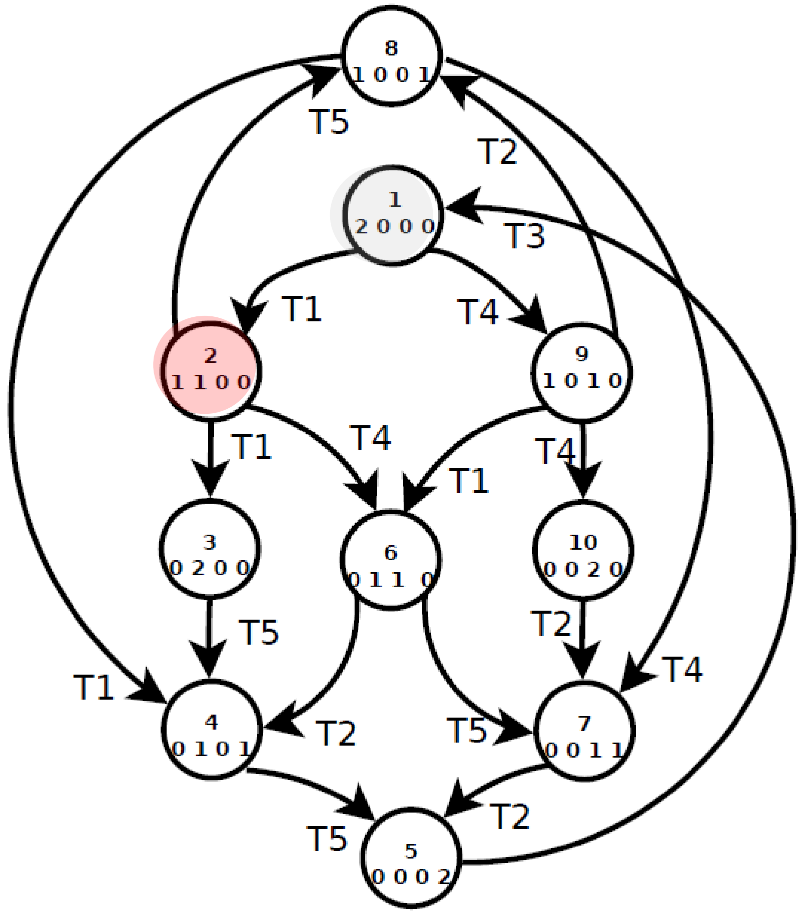
(untimed) reachability graph (RG)



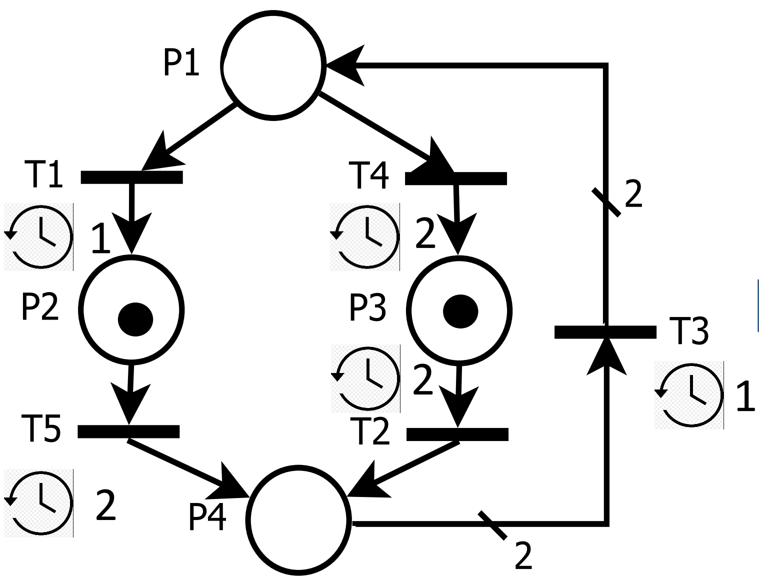
T-TPN



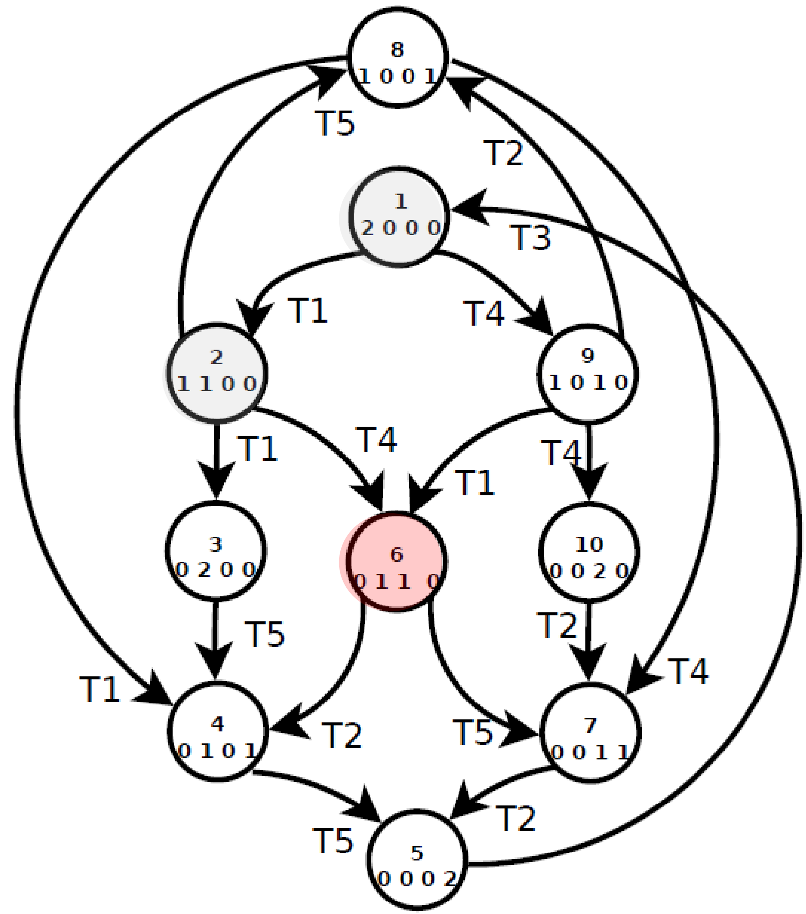
(untimed) reachability graph (RG)



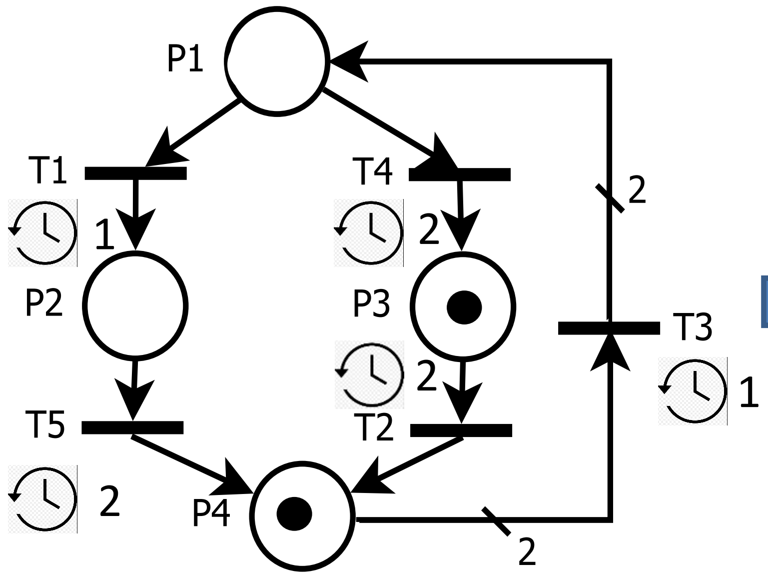
T-TPN



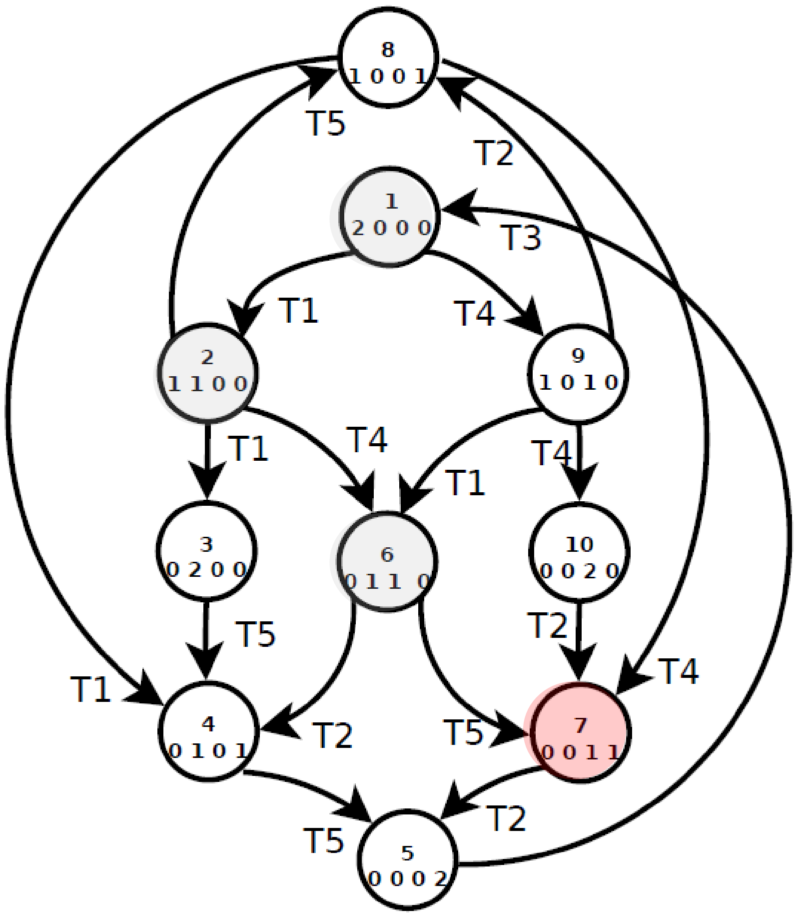
(untimed) reachability graph (RG)



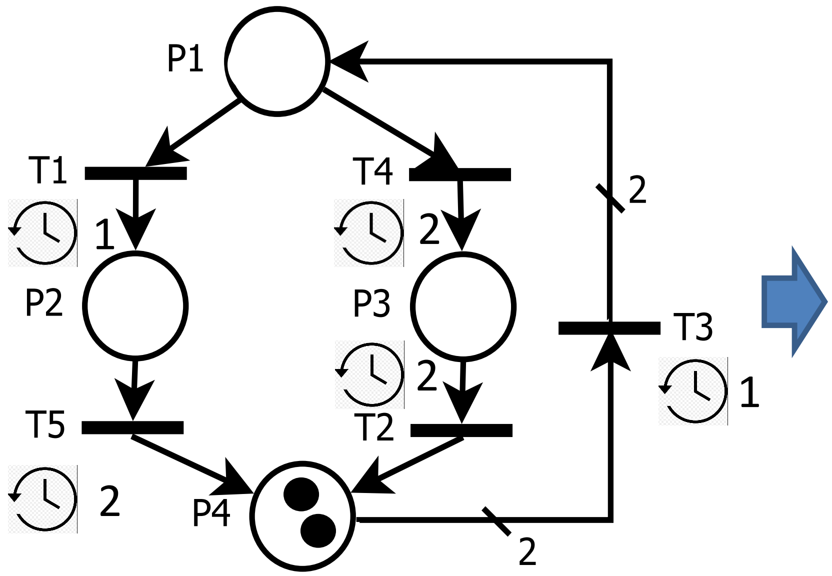
T-TPN



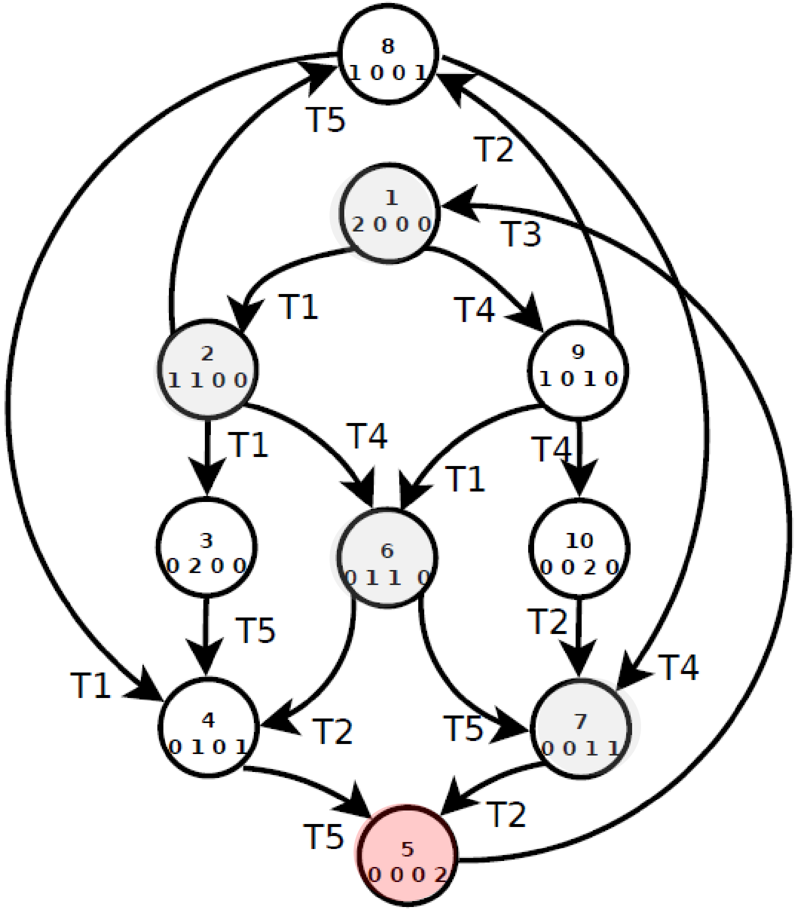
(untimed) reachability graph (RG)

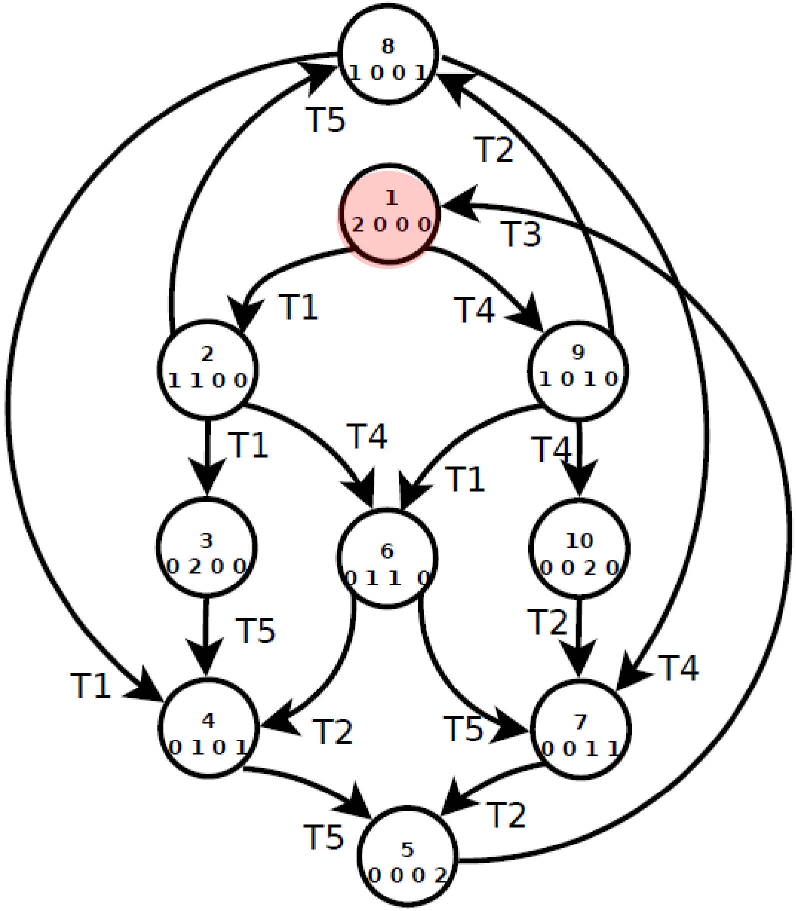


T-TPN

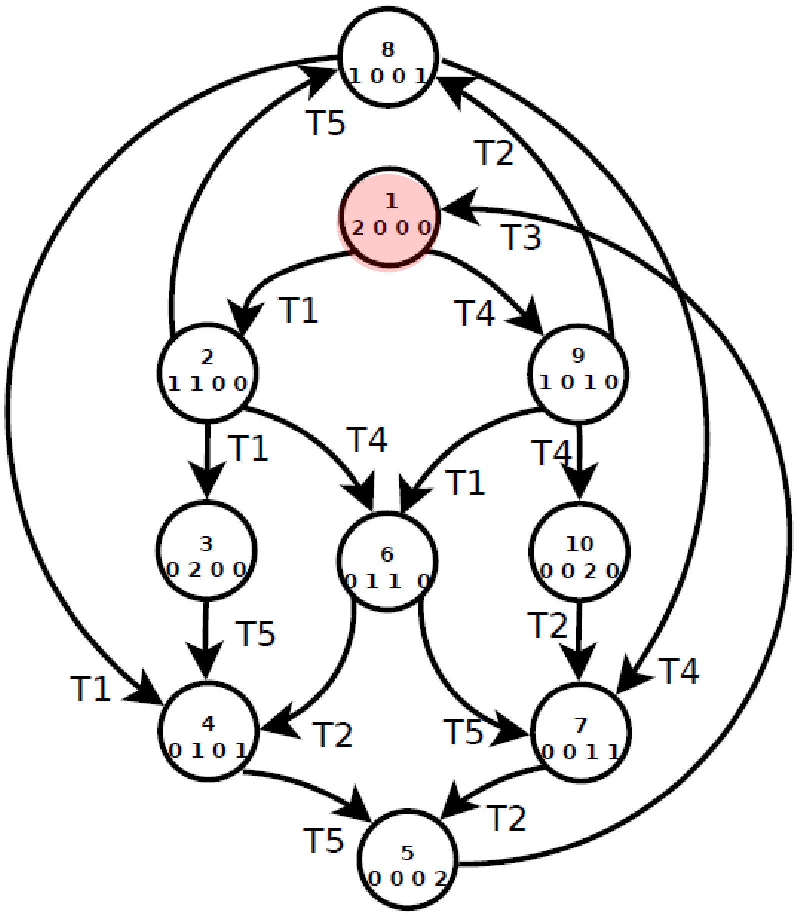


(untimed) reachability graph (RG)





Compute the reachability graph with MATLAB
>>[G,S]=RG_MACS(Wpr,Wpo,delta,MI)



RG =

0	1	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	1	4	5	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	1	2	4	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	Inf	5	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	2	5	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	1	4	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	5
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	2
3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

S=1	2	3	4	5	6	7	8	9	10
2	1	1	0	0	1	0	0	0	0
0	1	0	2	1	0	0	1	0	0
0	0	1	0	1	0	2	0	1	0
0	0	0	0	0	1	0	1	1	2

Compute the reachability graph with MATLAB
>>[RG,S]=RG_MACS(Wpr,Wpo,delta,MI)

MATLAB with example 2

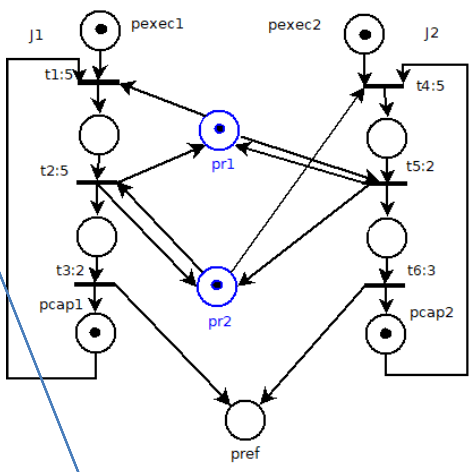
>> [Wpr,Wpo,delta,MI,Pjob] = Exemple2_MACS(1,1)

>>[RG,S]=RG_MACS(Wpr,Wpo,delta,MI)

RG: 16 states

RG =

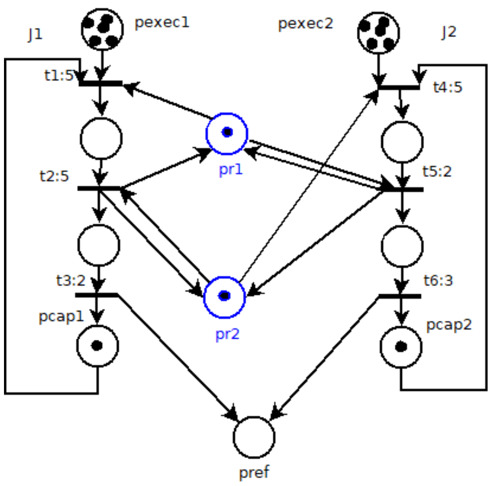
0	1	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	2	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	1	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	3	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	1	6	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	4	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	3	5	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	2	6	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	1	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	3	6	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	6
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	3
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0



Deadlocks

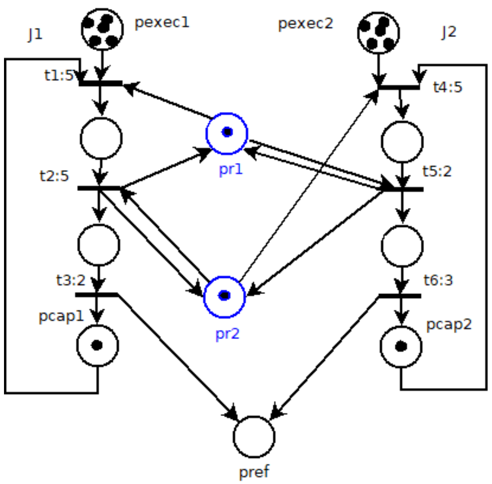
MATLAB with example 2
>> [Wpr,Wpo,delta,MI,Pjob] = Exemple2_MACS(5,5)
>>[RG,S]=RG_MACS(Wpr,Wpo,delta,MI)
RG: 256 states

Find the deadlocks in the problem ?



MATLAB with example 2
>> [Wpr,Wpo,delta,MI,Pjob] = Exemple2_MACS(5,5)
>>[RG,S]=RG_MACS(Wpr,Wpo,delta,MI)
RG: 256 states

Find the deadlocks in the problem ?

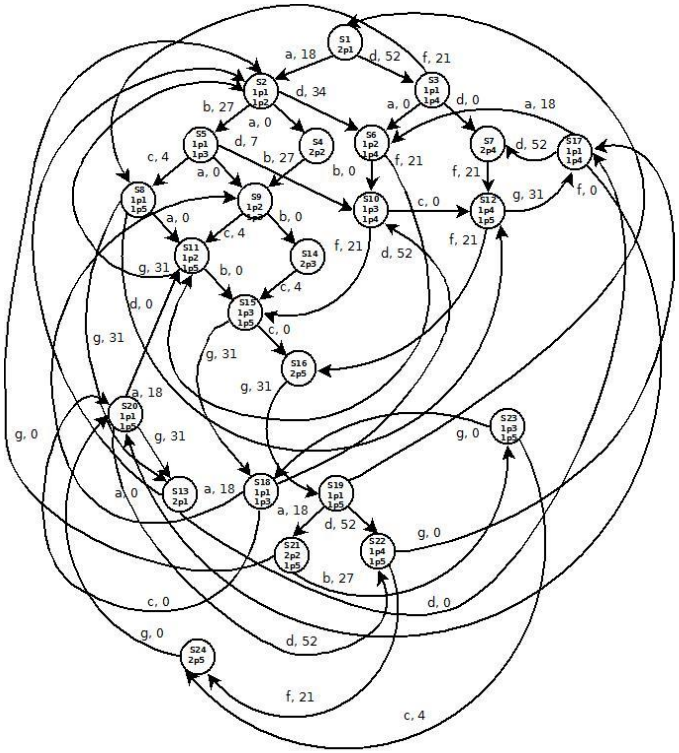


26 deadlocks:

[5 17 20 38 41 44 68 71 74 77 107 110 113 116 119 154 157 160 163 193 196 199 223 226 244 256]

Outline

1. Problems and objectives
2. Modeling scheduling problems with T-TPN
3. **Timed Extended Reachability Graph**
4. Approximated Timed Extended Reachability Graph
5. Beam Search
6. Conclusion and future works



How to compute an optimal schedule from the RG

Search for paths of minimal durations using

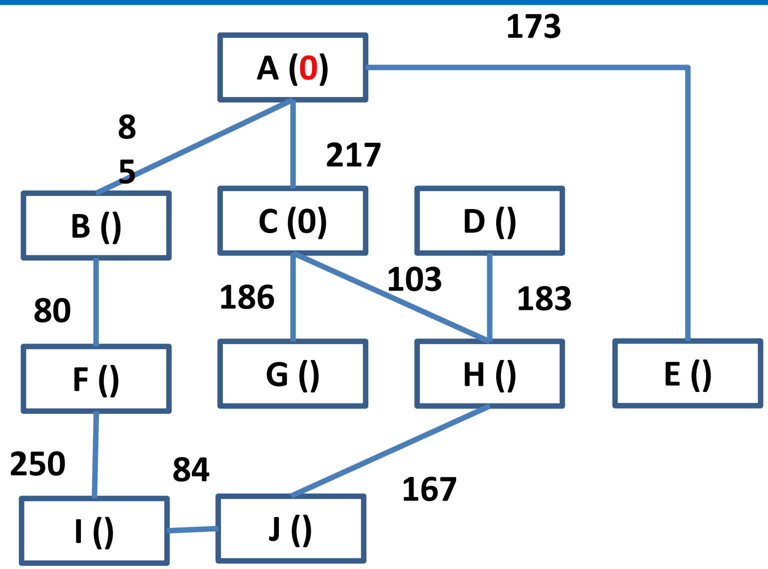
1) **The generator matrix of the RG**
2) **The time specifications δ**
3) **Replace each transition T_j in RG by $\delta(T_j)$**
4) **Dijkstra algorithm that searches for shortest path**

Cormen TH, Leiserson CE, Rivest RL, Stein C (2001) *Introduction to algorithms*. MIT Press and McGraw-Hill

Dijkstra EW (1971) A short introduction to the art of programming

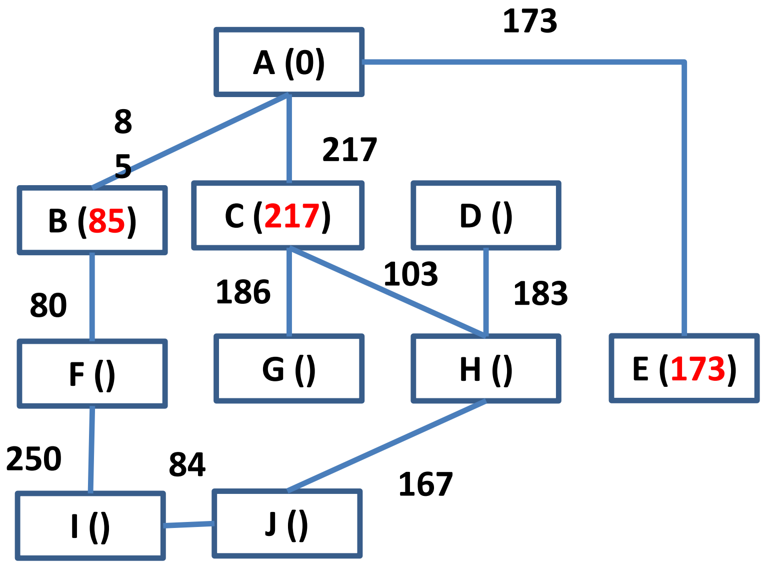
Dijkstra algorithm

	A	B	C	D	E	F	G	H	I	
1	0									
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
pre	A									



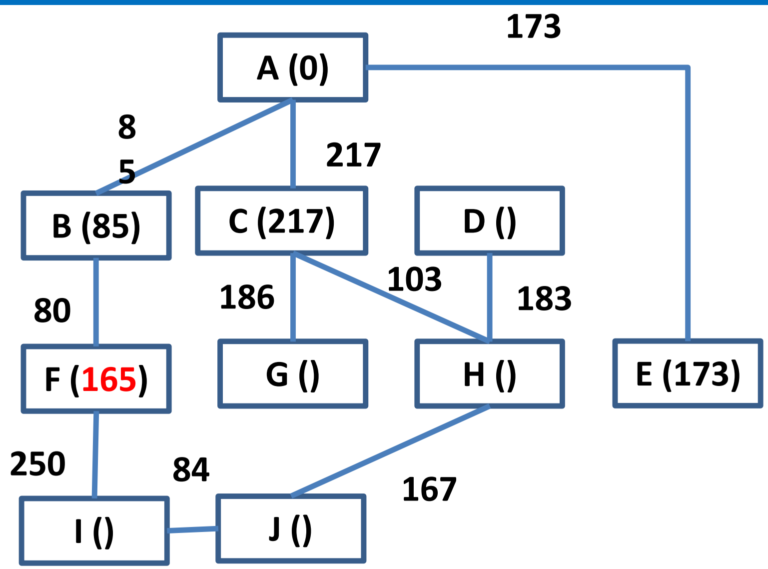
Dijkstra algorithm

	A	B	C	D	E	F	G	H	I
1	0								
2	X	85	217		173				
3									
4									
5									
6									
7									
8									
9									
10									
11									
pre	A	A							



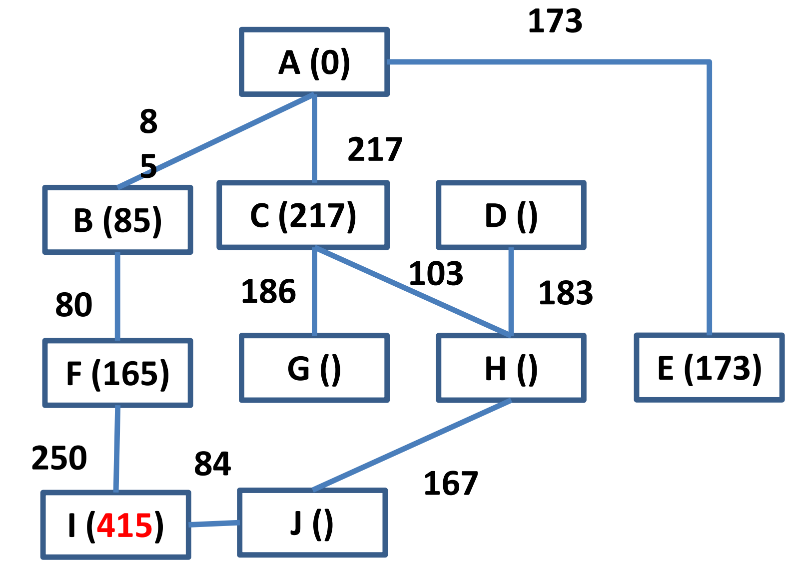
Dijkstra algorithm

	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4										
5										
6										
7										
8										
9										
10										
11										
pre	A	A				B				



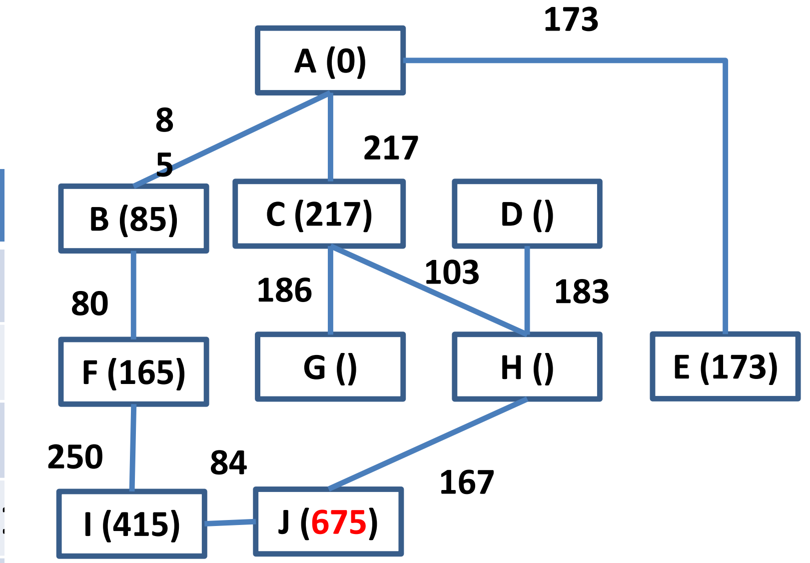
Dijkstra algorithm

	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4	X	X	217		173	X			41	
5										
6										
7										
8										
9										
10										
11										
pre	A	A			A	B				



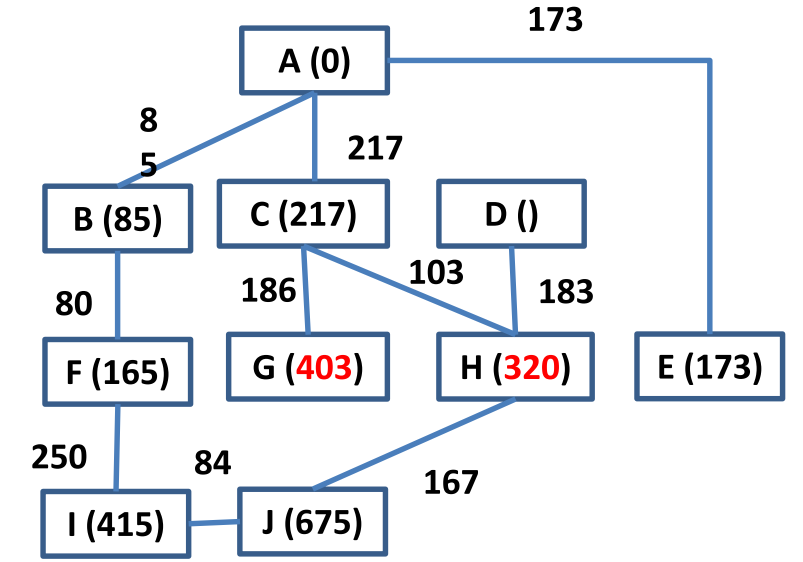
Dijkstra algorithm

	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4	X	X	217		173	X			41	
5	X	X	217		X	X			415	675
6										
7										
8										
9										
10										
11										
pre	A	A	A		A	B				

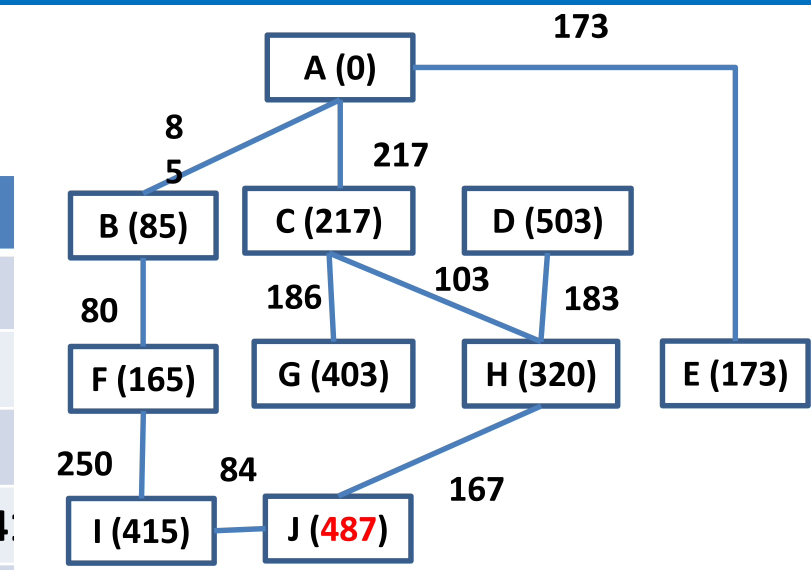


Dijkstra algorithm

	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4	X	X	217		173	X			41	
5	X	X	217		X	X			415	675
6	X	X	X		X	X	403	320	415	675
7										
8										
9										
10										
11										
pre	A	A	A		A	B		C		

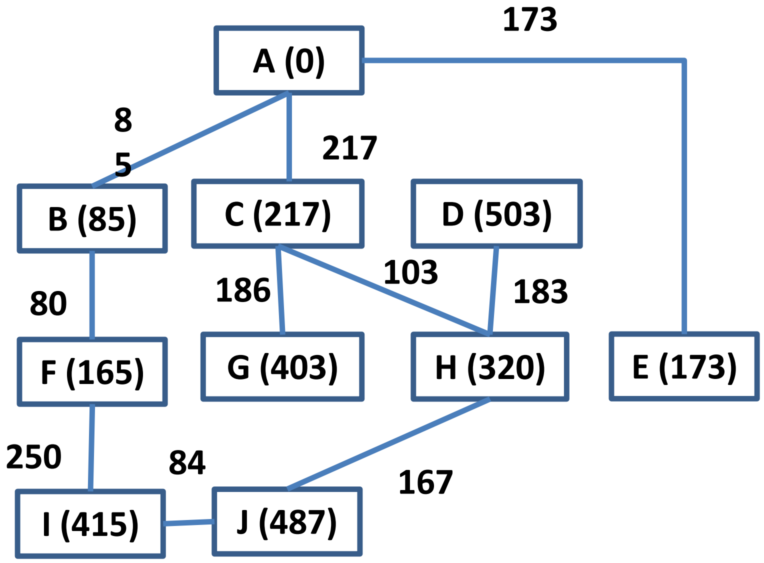


Dijkstra algorithm



	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4	X	X	217		173	X			4[illegible]	
5	X	X	217		X	X			415	675
6	X	X	X		X	X	403	320	415	675
7	X	X	X	503	X	X	403	X	415	487
8										
9										
10										
11										
pre	A	A	A		A	B	C	C		

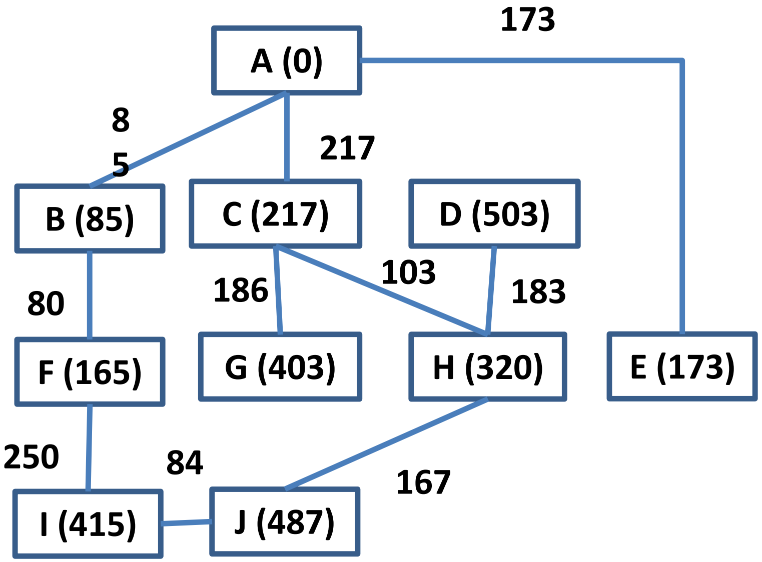
Dijkstra algorithm



	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4	X	X	217		173	X			41	
5	X	X	217		X	X			415	675
6	X	X	X		X	X	403	320	415	675
7	X	X	X	503	X	X	403	X	415	487
8	X	X	X	503	X	X	X	X	415	487
9										
10										
11										
pre	A	A	A		A	B	C	C	F	

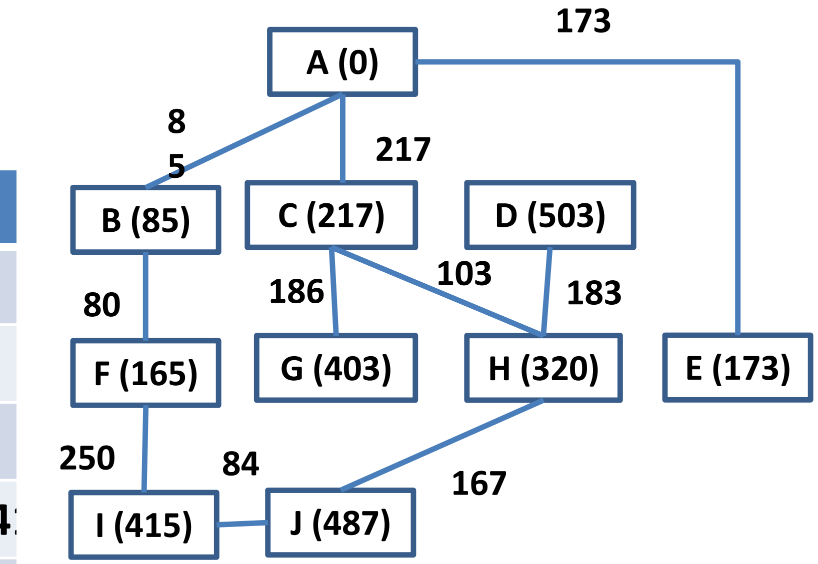
Dijkstra algorithm

	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4	X	X	217		173	X			4	
5	X	X	217		X	X			415	675
6	X	X	X		X	X	403	320	415	675
7	X	X	X	503	X	X	403	X	415	487
8	X	X	X	503	X	X	X	X	415	487
9	X	X	X	503	X	X	X	X	X	487
10										
11										
pre	A	A	A		A	B	C	C	F	H



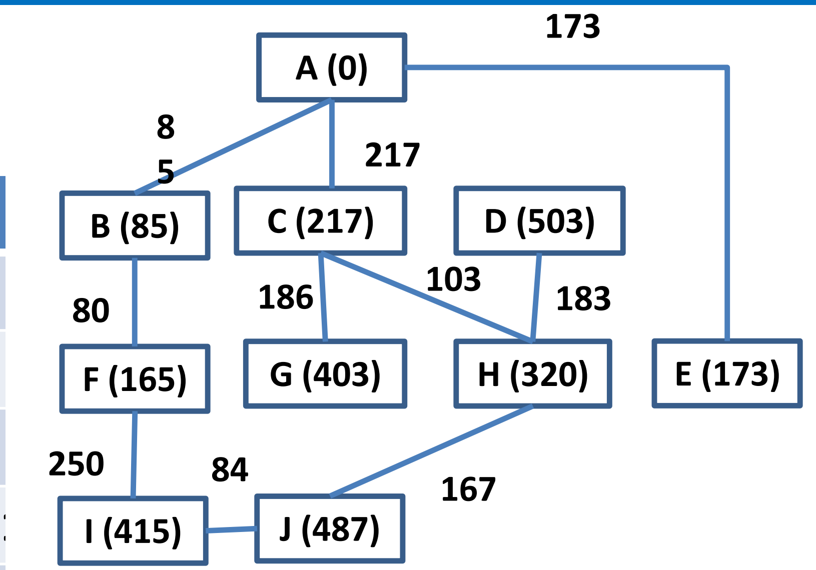
Dijkstra algorithm

	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4	X	X	217		173	X			4[illegible]	
5	X	X	217		X	X			415	675
6	X	X	X		X	X	403	320	415	675
7	X	X	X	503	X	X	403	X	415	487
8	X	X	X	503	X	X	X	X	415	487
9	X	X	X	503	X	X	X	X	X	487
10	X	X	X	503	X	X	X	X	X	X
11										
pre	A	A	A	H	A	B	C	C	F	H



Dijkstra algorithm

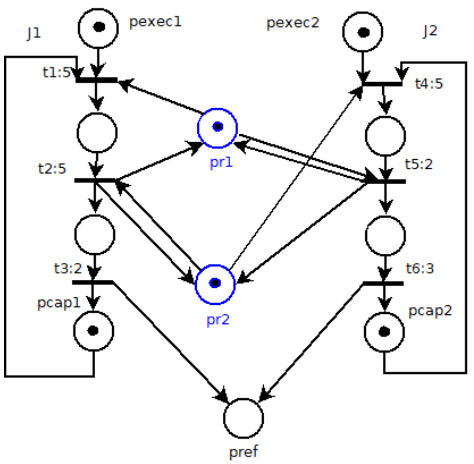
	A	B	C	D	E	F	G	H	I	
1	0									
2	X	85	217		173					
3	X	X	217		173	165				
4	X	X	217		173	X			41	
5	X	X	217		X	X			415	675
6	X	X	X		X	X	403	320	415	675
7	X	X	X	503	X	X	403	X	415	487
8	X	X	X	503	X	X	X	X	415	487
9	X	X	X	503	X	X	X	X	X	487
10	X	X	X	503	X	X	X	X	X	X
11	X	X	X	X	X	X	X	X	X	X
pre	A	A	A	H	A	B	C	C	F	H



MATLAB with example 2

>> [Wpr,Wpo,delta,MI,Pjob] = Exemple2_MACS(1,1)

>>[RG,S,RG_time]=RG_MACS(Wpr,Wpo,delta,MI)



Transition labels

RG =

0	1	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	2	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	1	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	3	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	1	6	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	4	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	3	5	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	2	6	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	1	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	3	6	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	6
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	3
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

```
delta= [5   5   2   5   2   3]';
```

RG_time =

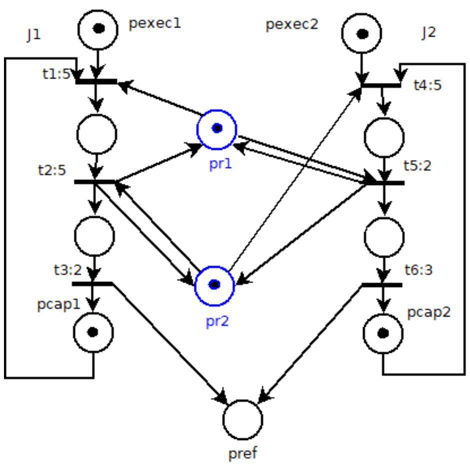
Transition firing duration

0	5	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	5	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	5	2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	2	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	3	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	2	2	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	3	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	2	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2	3	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	5	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	3
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	2
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

MATLAB
>>[predecesseur] = Dijkstra_predecessor_MACS([1:16],RG_time,1)
>>MS=16
>>[seq,cost] = Dijkstra_seq_MACS(TERG,TERG_time,predecesseur,1,MS)

seq =[1 2 3 4 5 6]
duration = 22 !!!

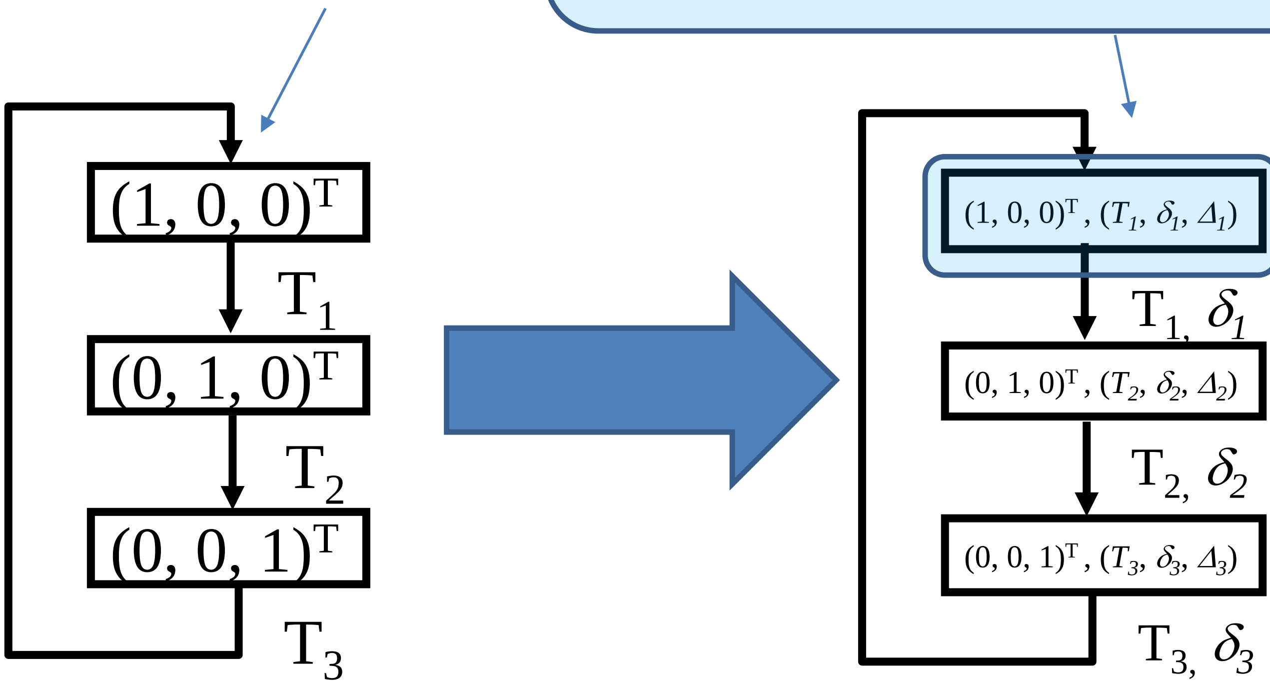
```
delta= [5  5  2  5  2  3]';
```

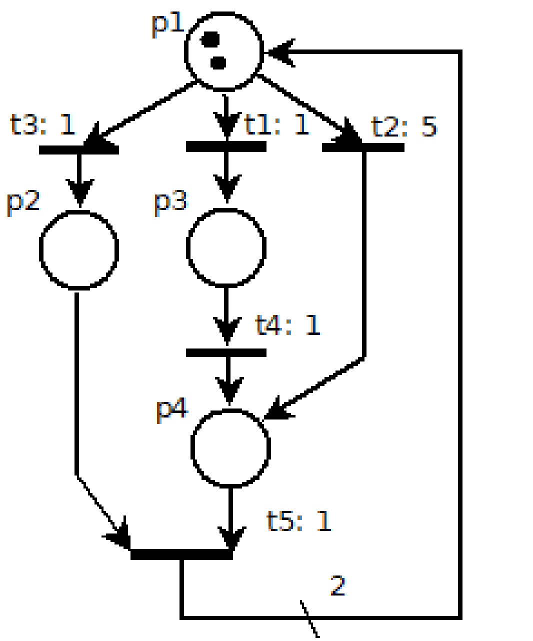


Timed Extended RG versus Untimed RG

State: Marking

State
=
Marking = information about product and resources
+
Information about time: δ or [δ, Δ]



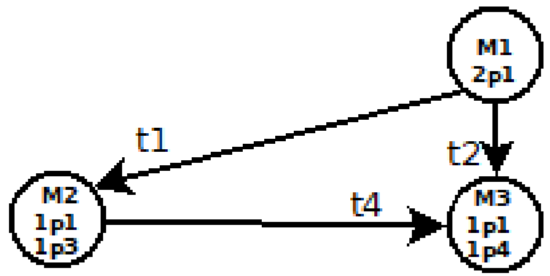


$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

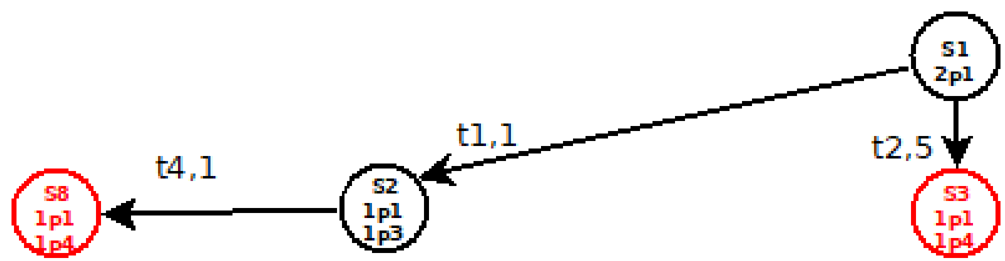
$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$



Untimed RG



t1 : 0
t2 : 3
t3 : 0

TERG

t1 : 0
t2 : 0
t3 : 0

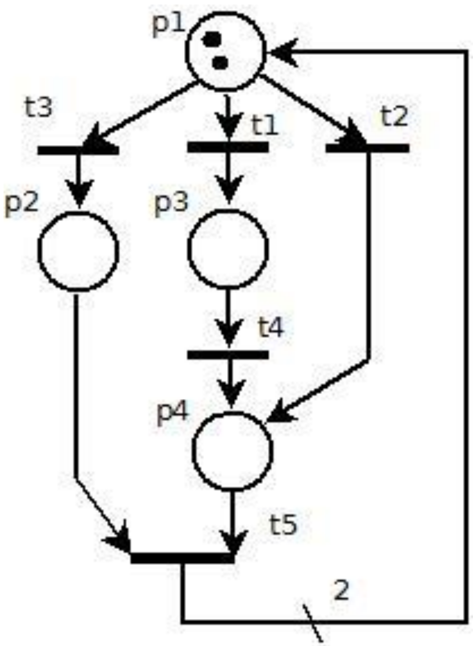
Timed Extended RG (TERG)

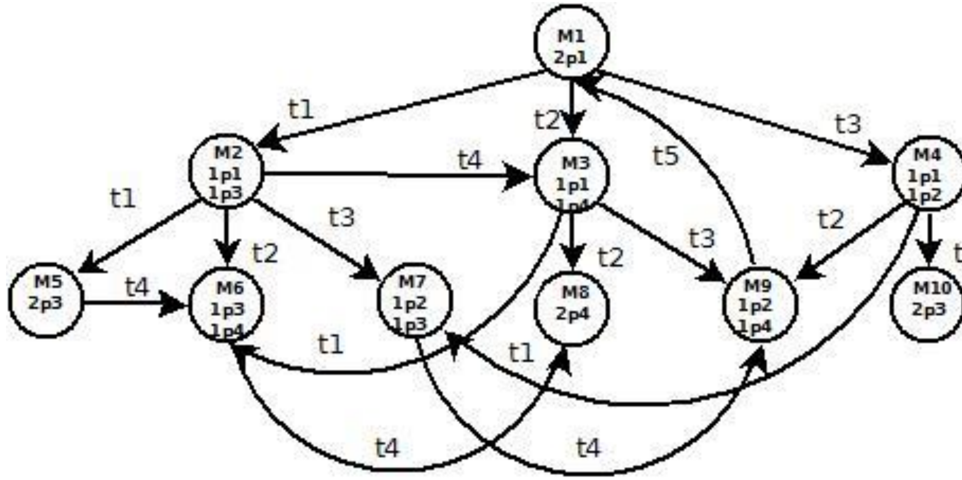
Proposition: Let <G, STI, M_I, A> be a controlled TPN system that behaves under earliest firing policy and its corresponding TERG <$S_E(M_I), \Omega_E, B_E, S_0$>.

(a) If the timed trajectory $(\sigma, M_I) = M(0)\ [(t_{j1}, \tau_1) > M(1) \ldots. > M(h-1)\ [(t_{jh}, \tau_h) > M(h)$ with $M(0) = M_I$ and $t_{jk} \in T$, $k = 1,\ldots h$, is feasible in TPN system, then a path $S_0\ S_1 \ldots S_h$ exists in <$S_E(M_I), \Omega_E, B_E, S_0$> st (1) $M(S_k) = M(k)$ for $k = 0,\ldots,h$; (2) $\Omega_E(S_{k-1}, S_k) = t_{jk}$ and $B_E(S_{k-1}, S_k).dt = \tau_k - \tau_{k-1}$ for $k = 1,\ldots,h$.

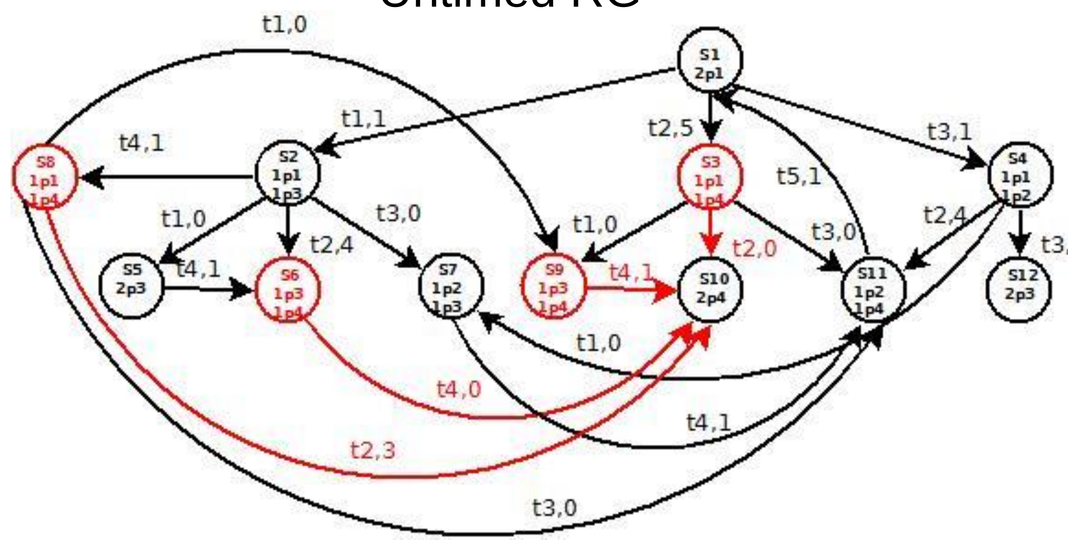
(b) If $S_0\ S_1 \ldots S_h$ is a path in TERG <$S_E(M_I), \Omega_E, B_E, S_0$> with S_0 the root node of the TERG then the timed trajectory $(\sigma, M_I) = M(0)\ [(t_{j1}, \tau_1) > M(1) \ldots. > M(h-1)\ [(t_{jh}, \tau_h) > M(h)$ with (1) $M(k) = M(S_k)$, $k = 0,\ldots,h$; (2) $t_{jk} = \Omega_E(S_{k-1}, S_k)$, $k = 1,\ldots,h$; (3) $\tau_1 = B_E(S_0, S_1).dt$; (4) $\tau_k = \tau_{k-1} + B_E(S_{k-1}, S_k).dt$, $k = 1,\ldots,h$; is feasible in TPN

All feasible timed trajectories in time PN starting from M_I and executed under earliest firing policy are EXACTLY encoded in Timed Extended Reachability Graph





Untimed RG



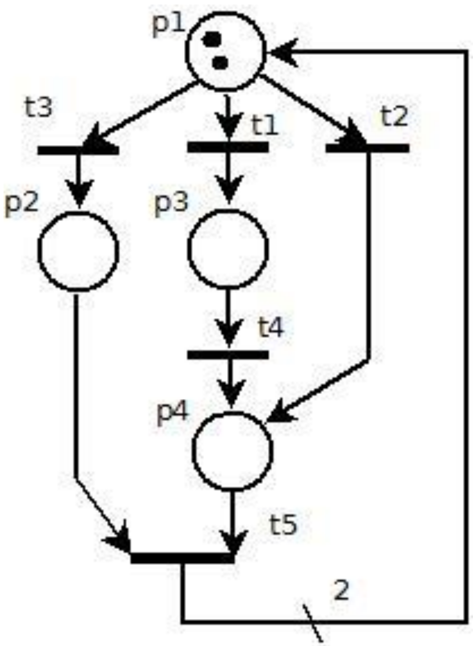
TERG

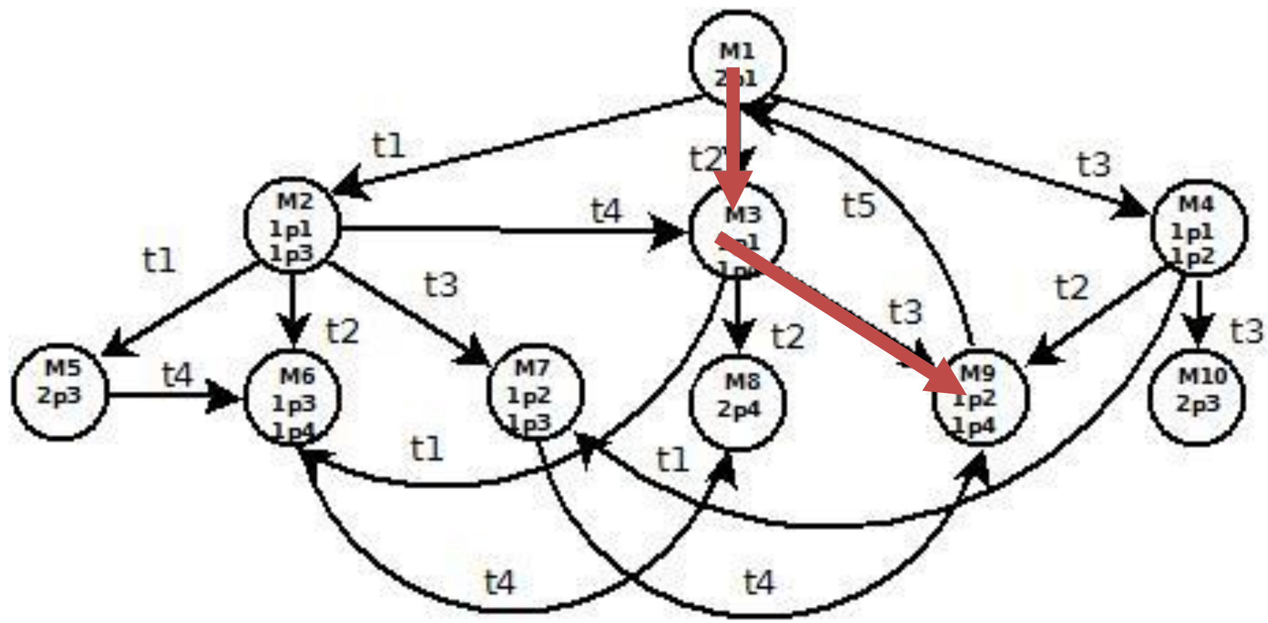
$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$





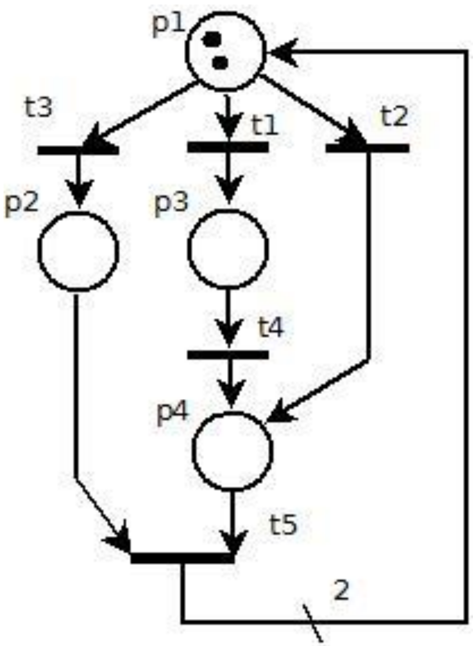
$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

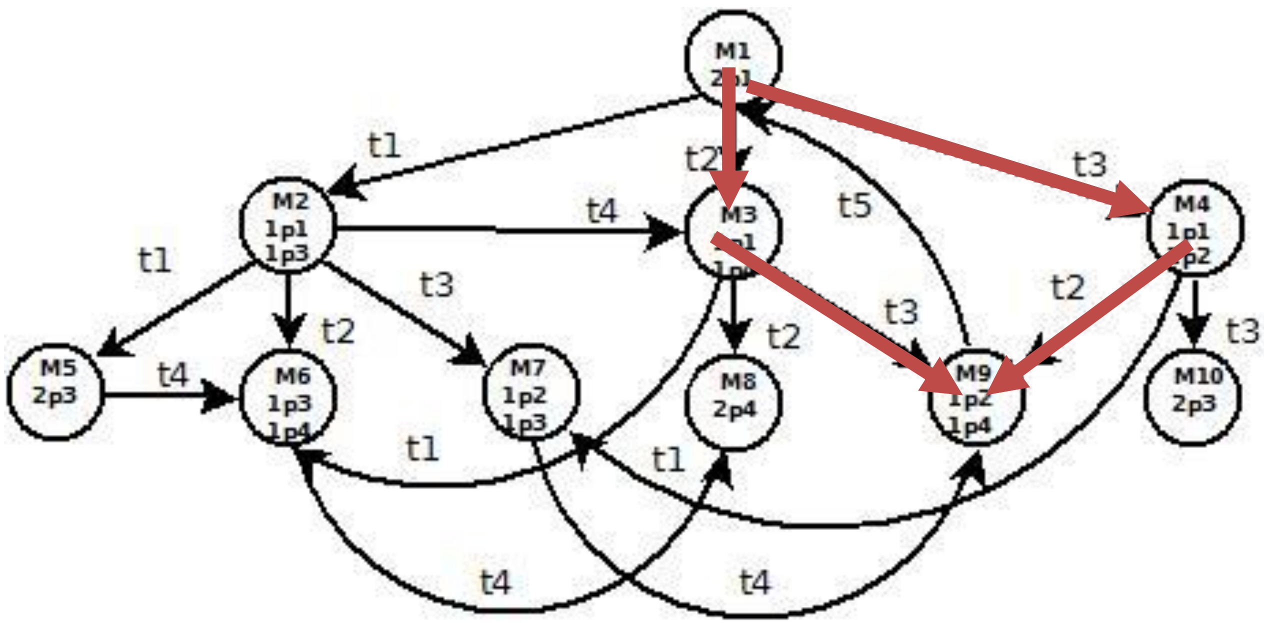
$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$

Untimed RG





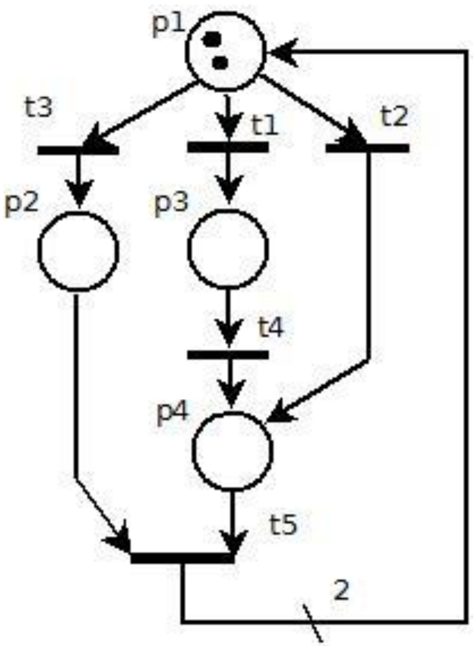
$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

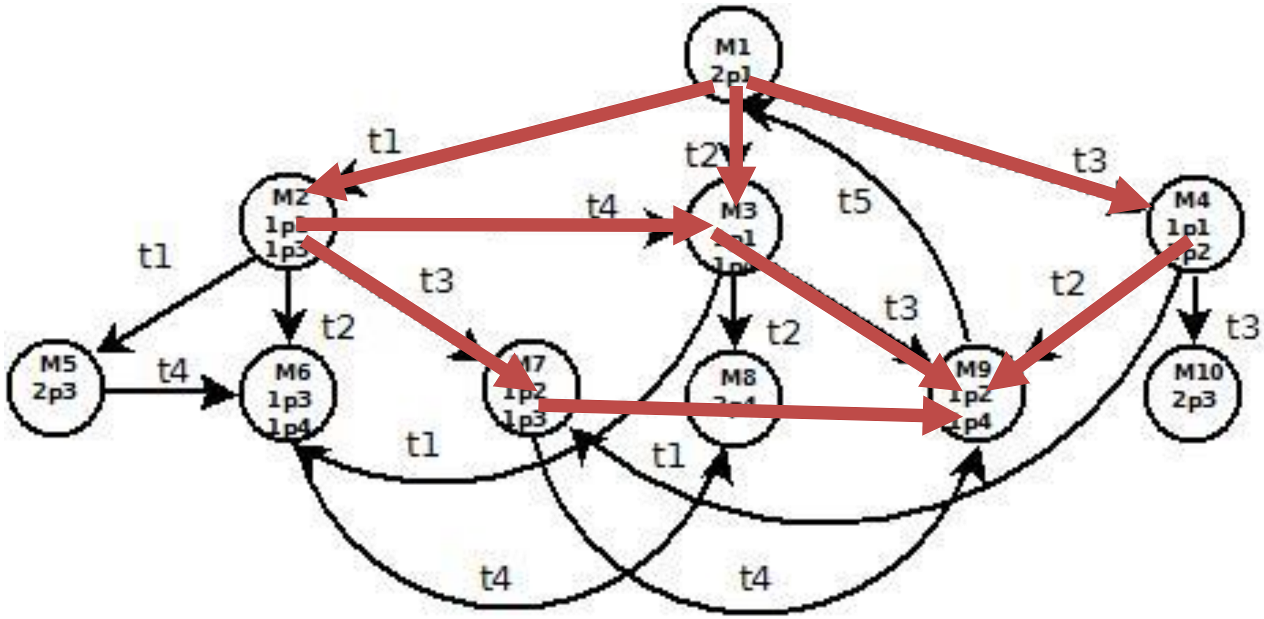
$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$

Untimed RG





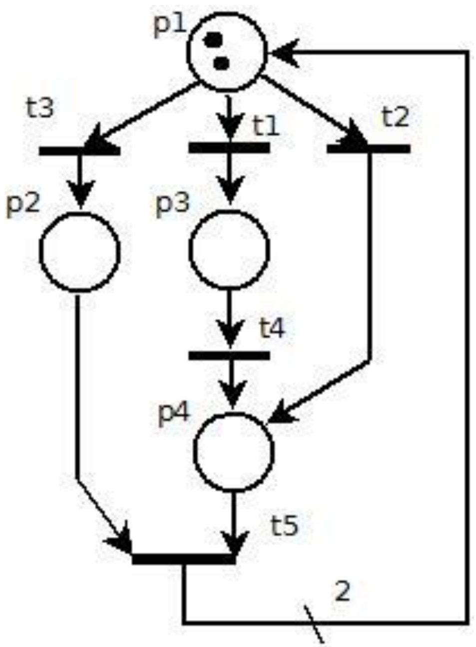
$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

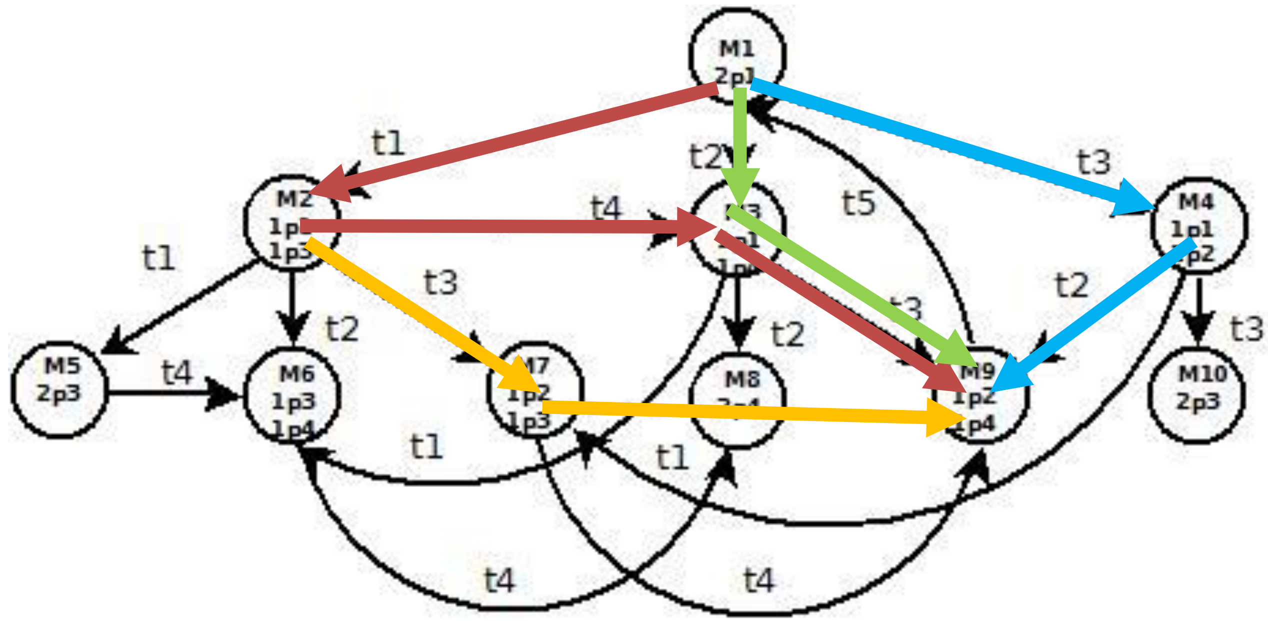
$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$

Untimed RG : what is the path of minimal duration ?
=> exhaustive path enumeration
+ path duration evaluation





$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\boldsymbol{\delta = (1\ 5\ 1\ 1\ 1)^T}$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$

Exercice

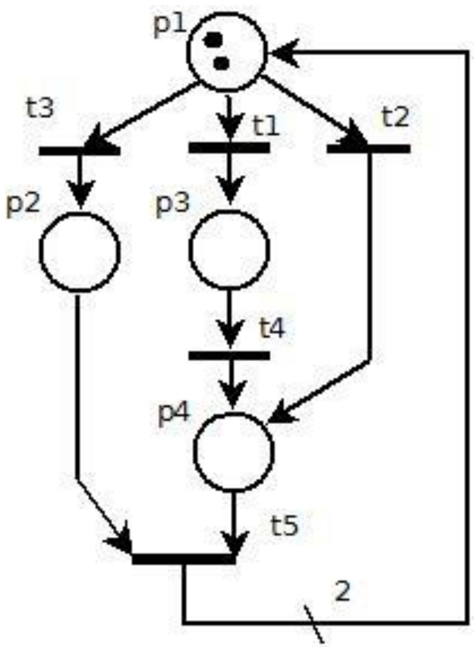
Path durations

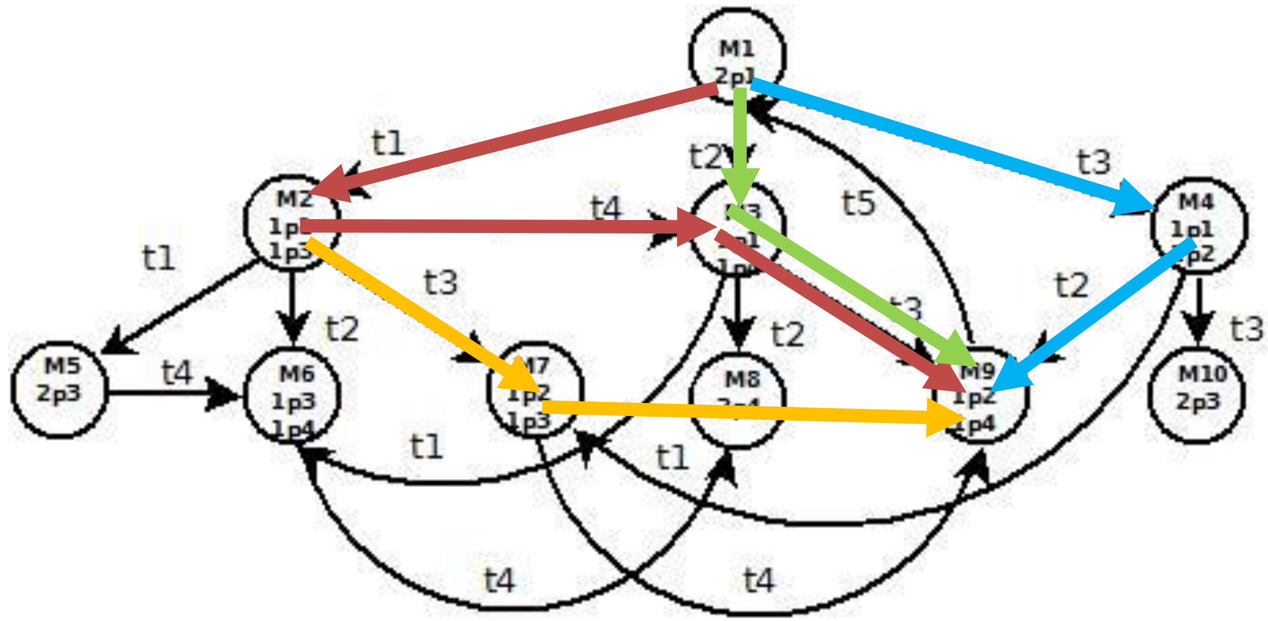
Path 1 : ???

Path 2 : ???

Path 3 : ???

Path 4 : ???





$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$

Exercice

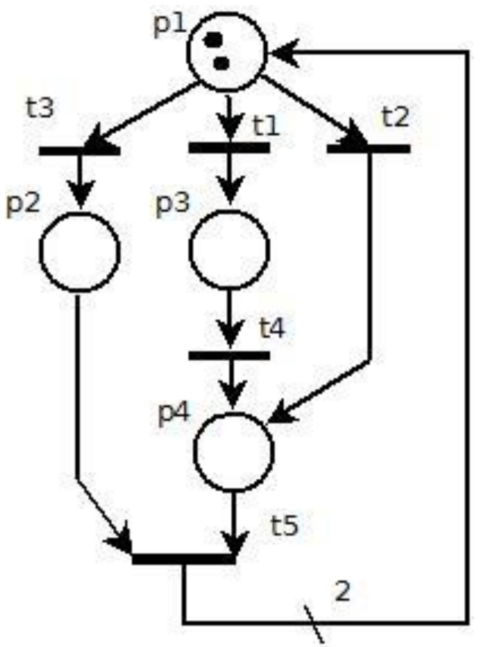
Path durations

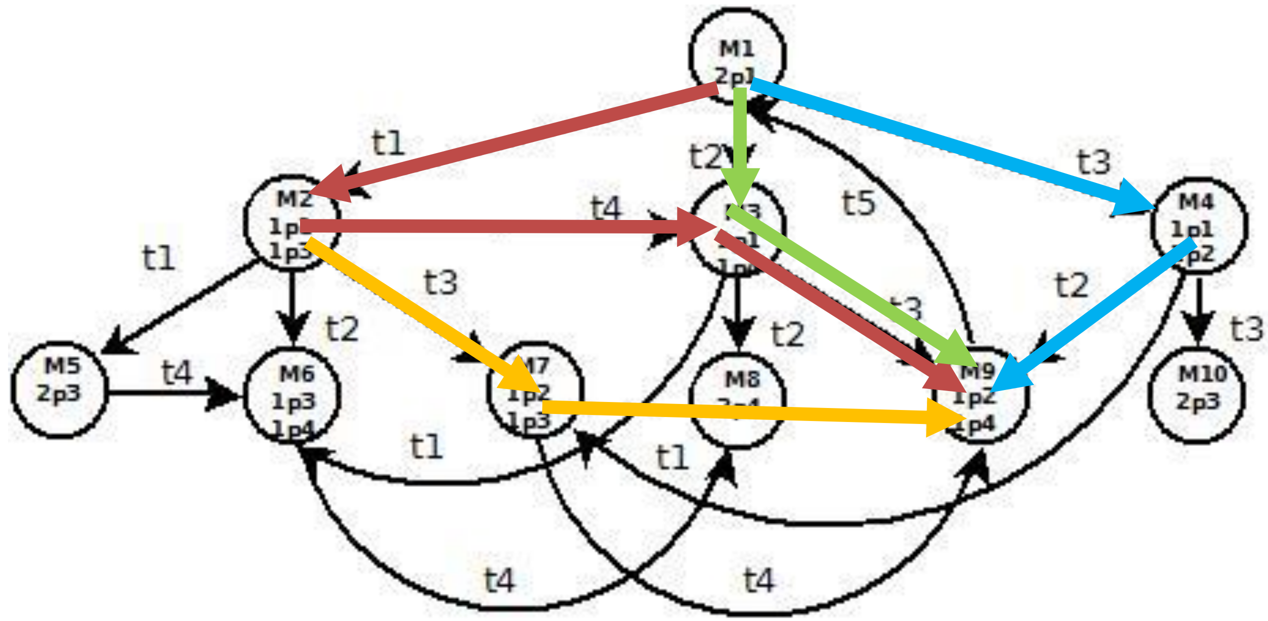
Path 1 : T3(1)T2(5), duration = 5

Path 2 : ???

Path 3 : ???

Path 4 : ???





$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$

Exercice

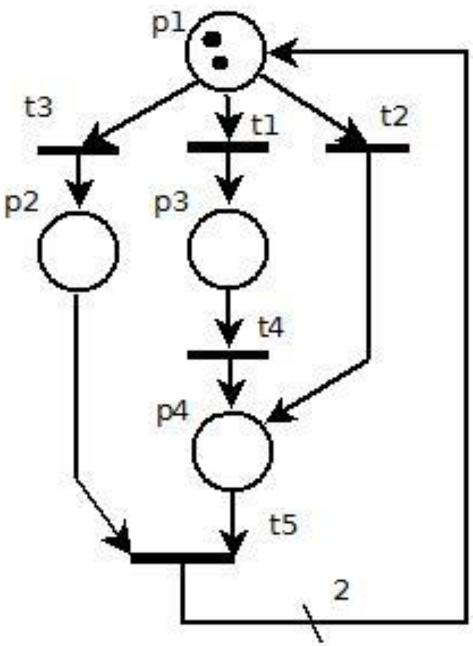
Path durations

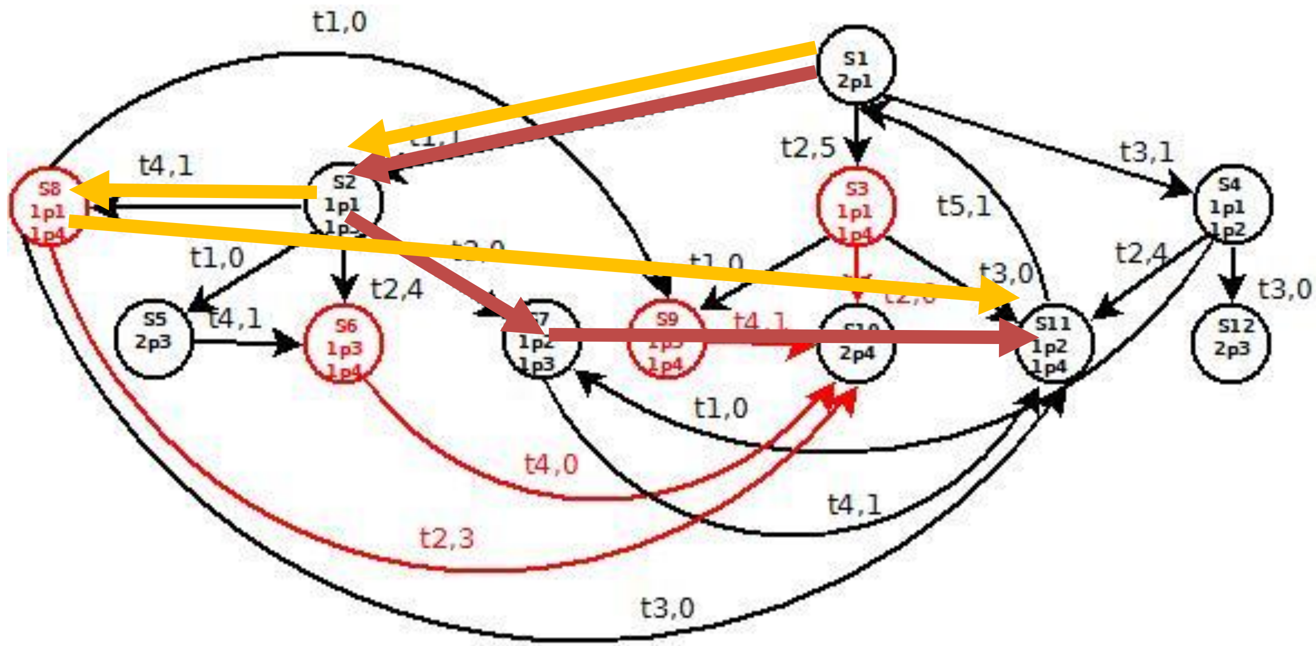
Path 1 : T3(1)T2(5), duration = 5TU

Path 2 : T2(5)T3(5), duration = 5TU

Path 3 : T1(1)T4(2)T3(2), duration = 2 TU

Path 4 : T1(1)T3(1)T4(2), duration = 2 TU





$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$

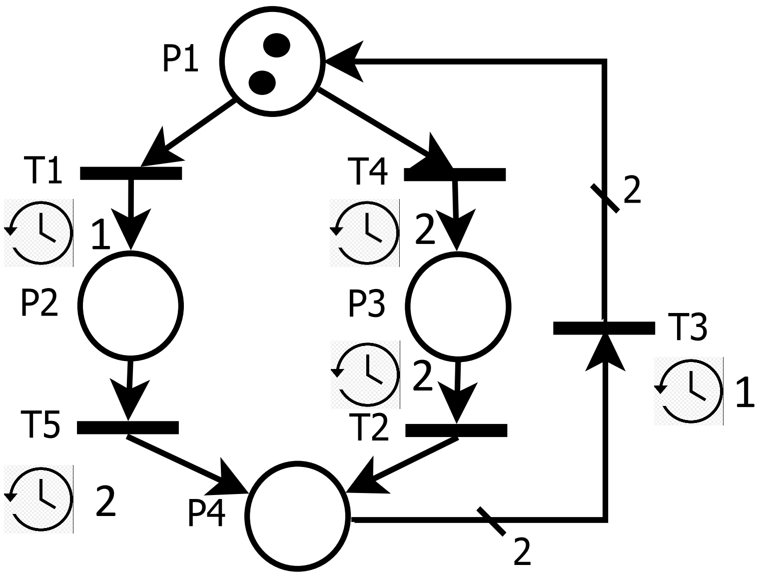
TERG + Dijkstra algorithm

$\Rightarrow$ Paths of minimal duration

Path 3 : T1(1)T4(2)T3(2), duration = 2 TU

Path 4 : T1(1)T3(1)T4(2), duration = 2 TU

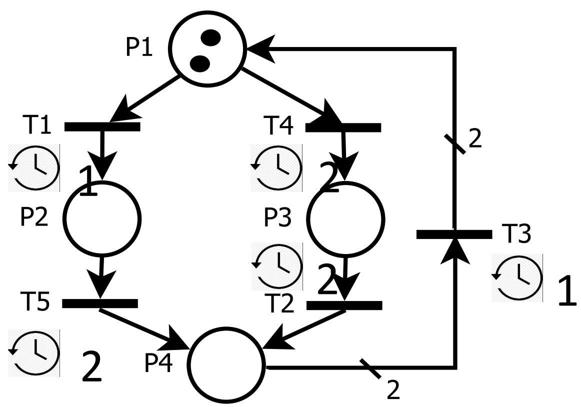
Example 1

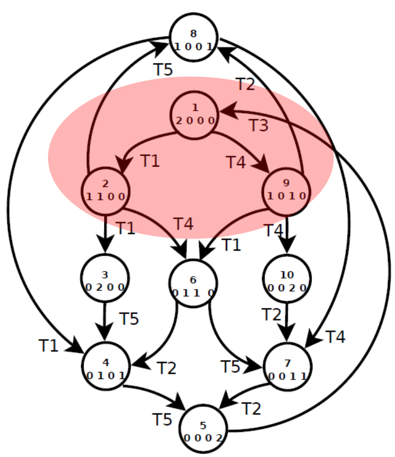


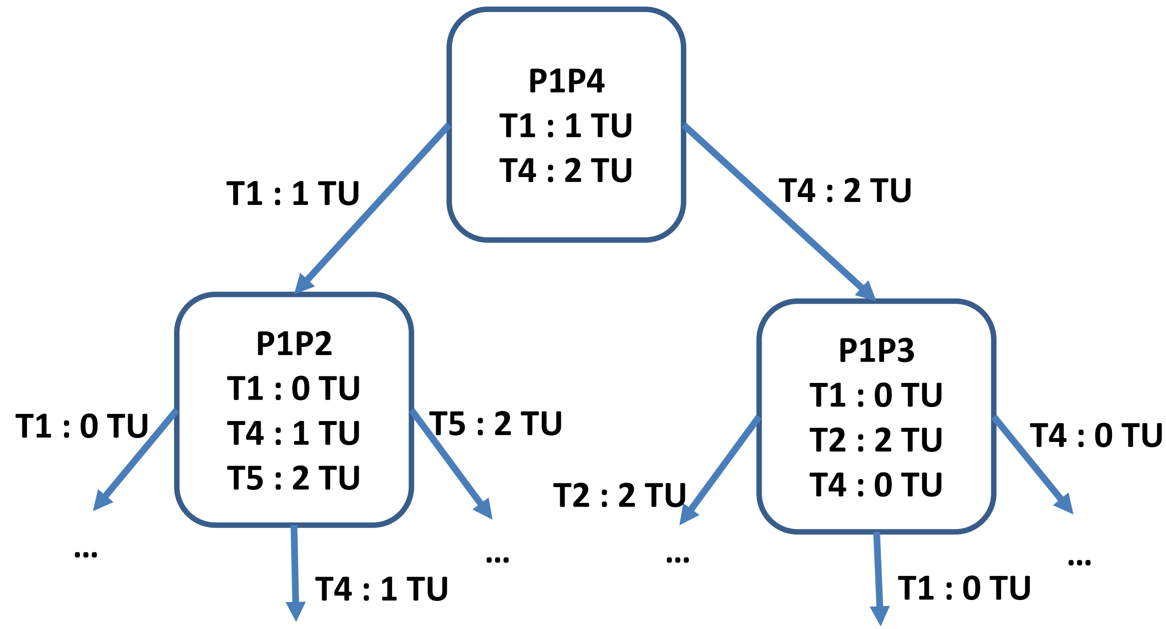
Exercice

TERG ???

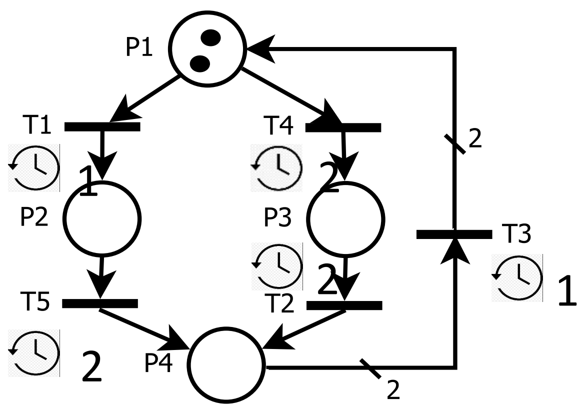
Exercice

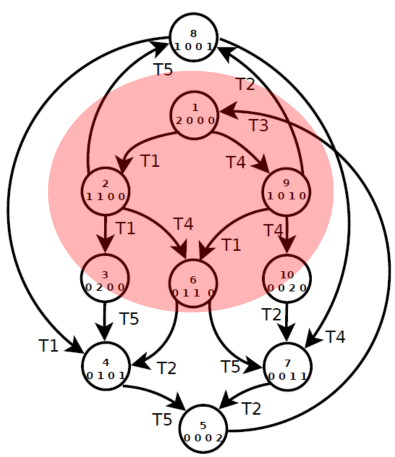

P1
T1
T4
1
2
P2
P3
2
T3
2
1
T5
T2
2
P4
2

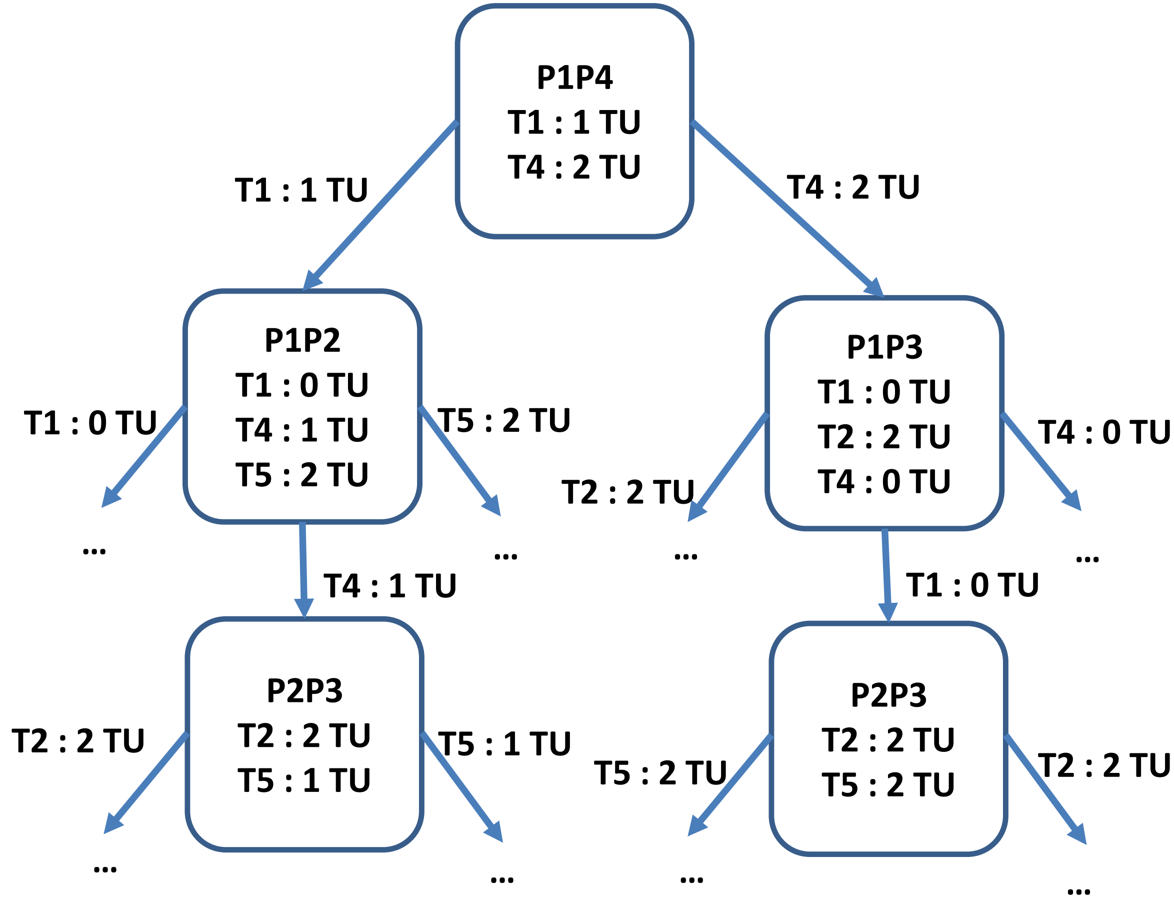



P1P4
T1 : 1 TU
T4 : 2 TU
T1 : 1 TU
T4 : 2 TU
P1P2
T1 : 0 TU
T4 : 1 TU
T5 : 2 TU
P1P3
T1 : 0 TU
T2 : 2 TU
T4 : 0 TU
T1 : 0 TU
T5 : 2 TU
T2 : 2 TU
T4 : 0 TU
T4 : 1 TU
T1 : 0 TU
…
…
…
…

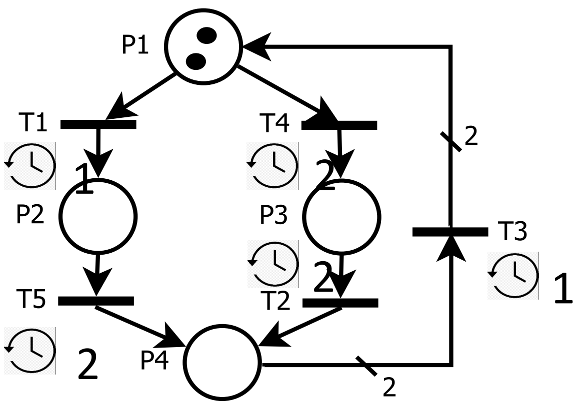
Exercice

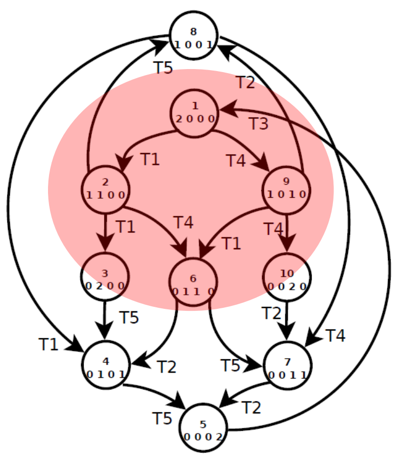

P1
T1
T4
2
1
2
P2
P3
T3
1
2
T5
T2
2
P4
2


T5
T2
T3
T1
T4
T1
T4
T1
T4
T5
T2
T1
T2
T5
T4
T5
T2

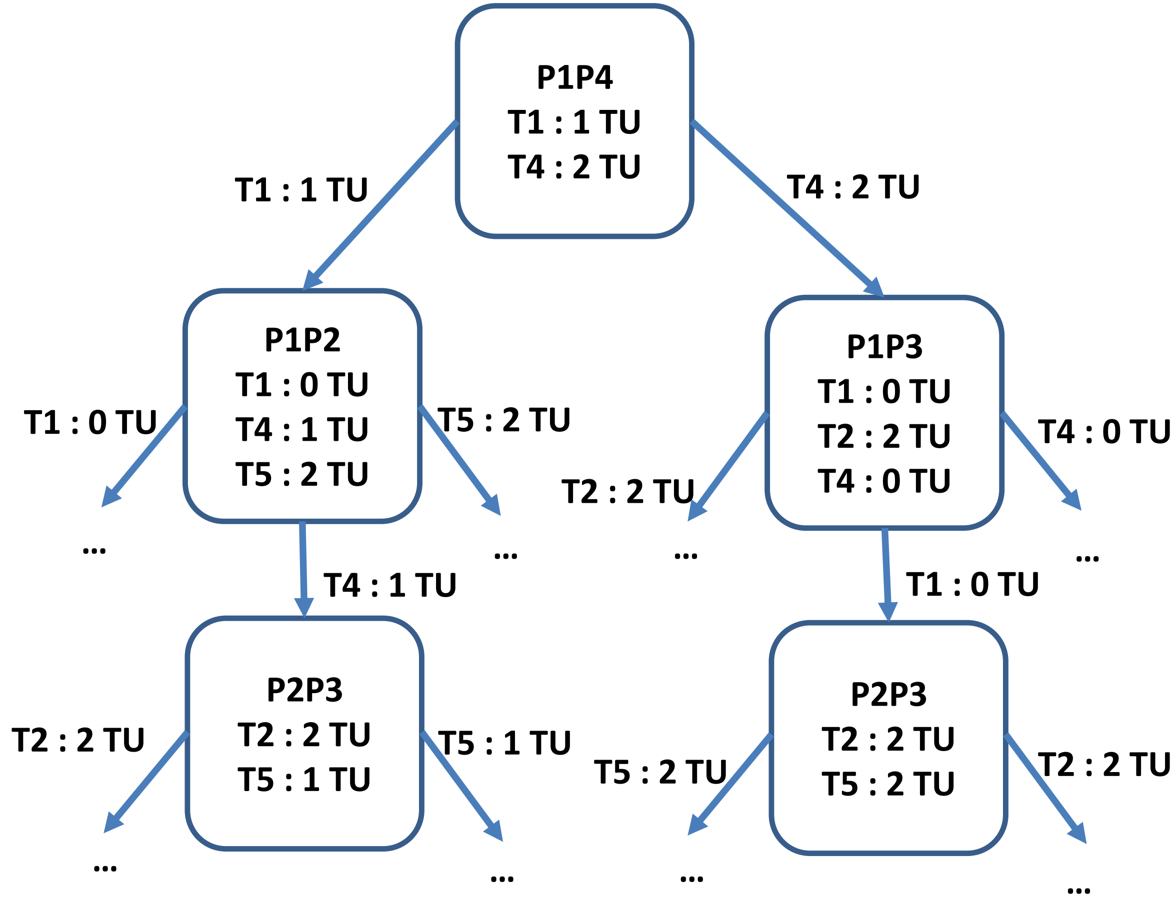

P1P4
T1 : 1 TU
T4 : 2 TU
T1 : 1 TU
T4 : 2 TU
P1P2
T1 : 0 TU
T4 : 1 TU
T5 : 2 TU
P1P3
T1 : 0 TU
T2 : 2 TU
T4 : 0 TU
T1 : 0 TU
T5 : 2 TU
T2 : 2 TU
T4 : 0 TU
...
...
...
...
T4 : 1 TU
T1 : 0 TU
P2P3
T2 : 2 TU
T5 : 1 TU
P2P3
T2 : 2 TU
T5 : 2 TU
T2 : 2 TU
T5 : 1 TU
T5 : 2 TU
T2 : 2 TU
...
...
...
...

Exercice 2





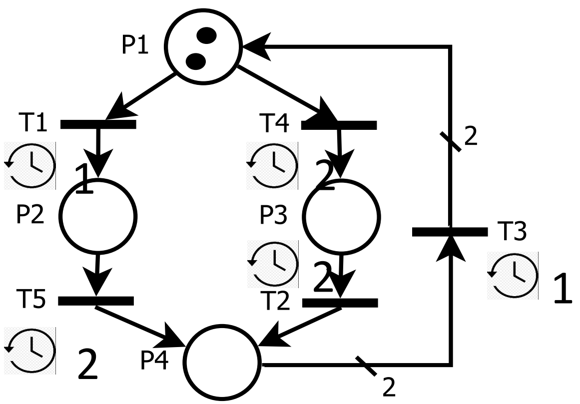
10 states



14 states

MATLAB

>> [TERG,TERG_time,S]=ApTERG_MACS(Wpr,Wpo,MI,delta,0)



Label of the transitions

TERG =

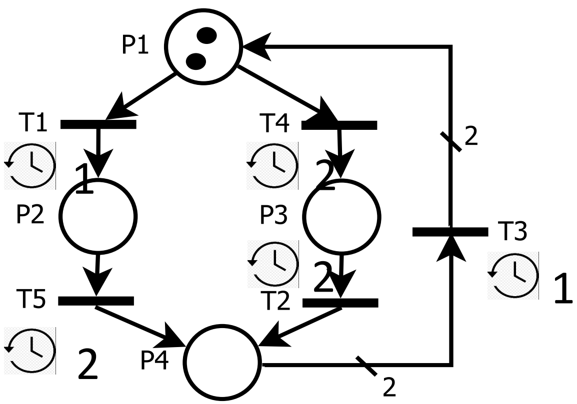
0	1	4	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	1	4	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	Inf	2	1	4	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	2	5	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	1	4	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2	Inf	Inf	Inf	5	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	2	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	5
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	2
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	2
3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

S =

2	1	1	0	0	1	0	0	0	0	0	0	0	0
0	1	0	2	1	0	1	0	1	0	1	0	0	0
0	0	1	0	1	0	1	2	0	1	0	1	1	0
0	0	0	0	0	1	0	0	1	1	1	1	1	2

MATLAB

>> [TERG,TERG_time,S]=ApTERG_MACS(Wpr,Wpo,MI,delta,0)



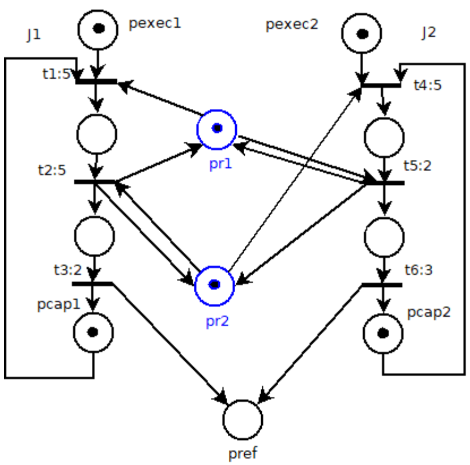
Transition firing duration

TERG_time =

0	1	2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	0	1	2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	Inf	2	0	0	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	2	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	2	1	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	0	0	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2	Inf	Inf	Inf	2	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	2	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	0
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	1
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	2
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	0
1	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

S =

2	1	1	0	0	1	0	0	0	0	0	0	0	0
0	1	0	2	1	0	1	0	1	0	1	0	0	0
0	0	1	0	1	0	1	2	0	1	0	1	1	0
0	0	0	0	0	1	0	0	1	1	1	1	1	2



MATLAB

>> [TERG,TERG_time,S]=ApTERG_MACS(Wpr,Wpo,MI,delta,0)

TERG_timed =

0	5	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	5	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	0	2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	2	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	3	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	0	2	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	5	0	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	2	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	2	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	0	3	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	2	0	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	5	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	3
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	0
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	0
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	2
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

RG_time =

0	5	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	0	Inf	5	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	0	Inf	5	2	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	0	Inf	Inf	2	5	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	3	Inf	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	Inf	5	Inf	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	2	2	Inf	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	3	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	5	Inf	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	Inf	2	Inf	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	2	3	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	5	Inf
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	Inf	3
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0	2
Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

16 x 16

19 x 19

Timed Extended Reachability Graph

Proposition: If $\boldsymbol{S(M_I)}$ is of finite cardinality N, then $\boldsymbol{S_E(M_I)}$ is of finite cardinality N_E that satisfies:

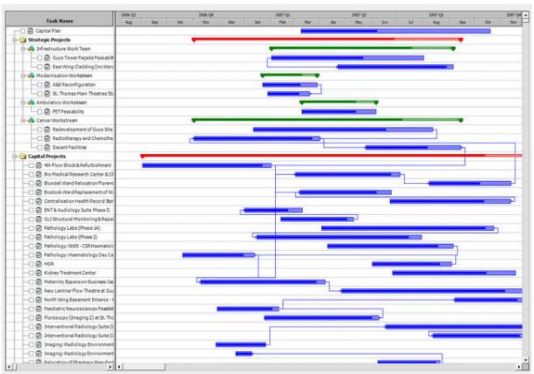
$N \leq N_E \leq N.((D_m + 1).(D_M + 1))^{k.q}$

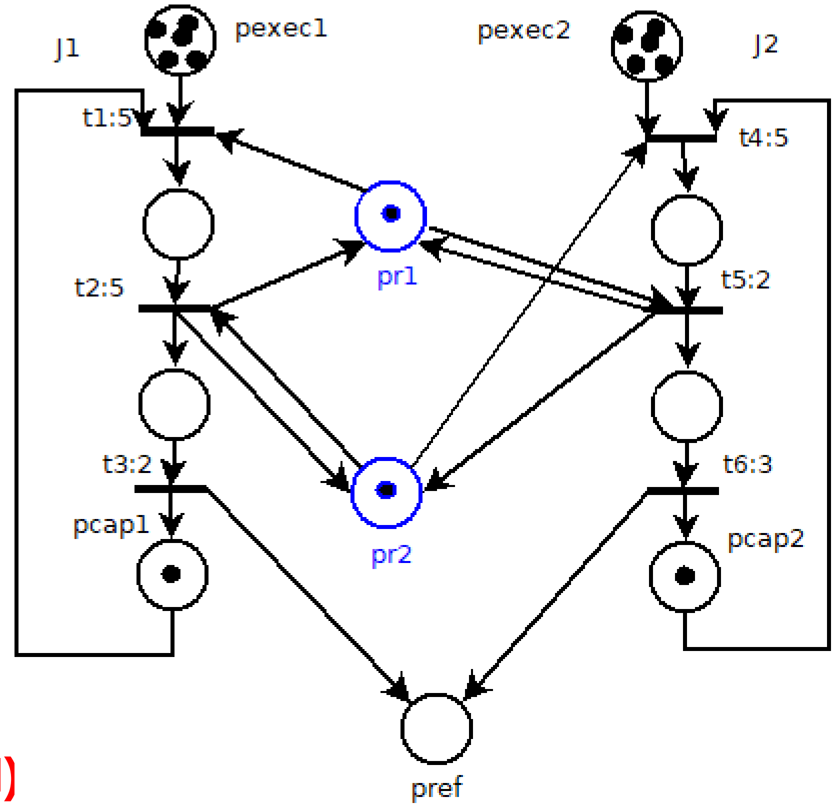
with $D_m = \max\{d_{min\,j} : t_j \in \boldsymbol{T}\}$ and $D_M = \max\{d_{MAX\,j} : t_j \in \boldsymbol{T} \text{ and } d_{MAX\,j} < \infty\}$

The number of states of the Timed Extended Reachability Graph INCREASES EXPONENTIALLY with respect to the time specifications

D. Lefebvre, Approximated Timed Reachability Graphs for the robust control of discrete event systems, *Discrete Event Dynamic Systems: theory and applications*, 29(1), 31-56, 2019

Scheduling problems



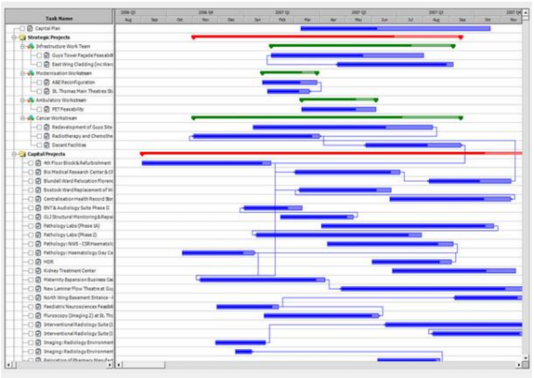


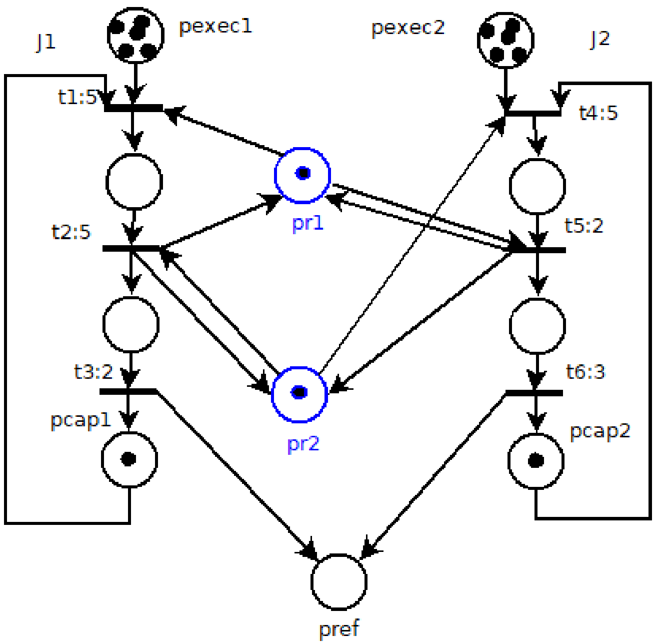
MATLAB
>> [Wpr,Wpo,delta,MI,Pjob]
= Exemple2_MACS(5,5)

>>[RG,S]=RG_MACS(Wpr,Wpo,MI)
RG: 256 states

>>[TERG,TERG_time,S]=ApTERG_MACS(Wpr,Wpo,MI,delta,0)
TERG : 443 states

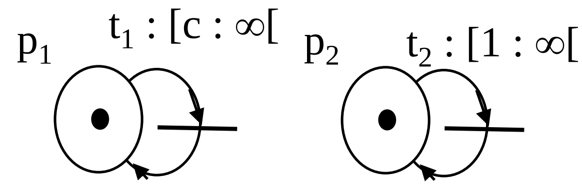
Scheduling problems

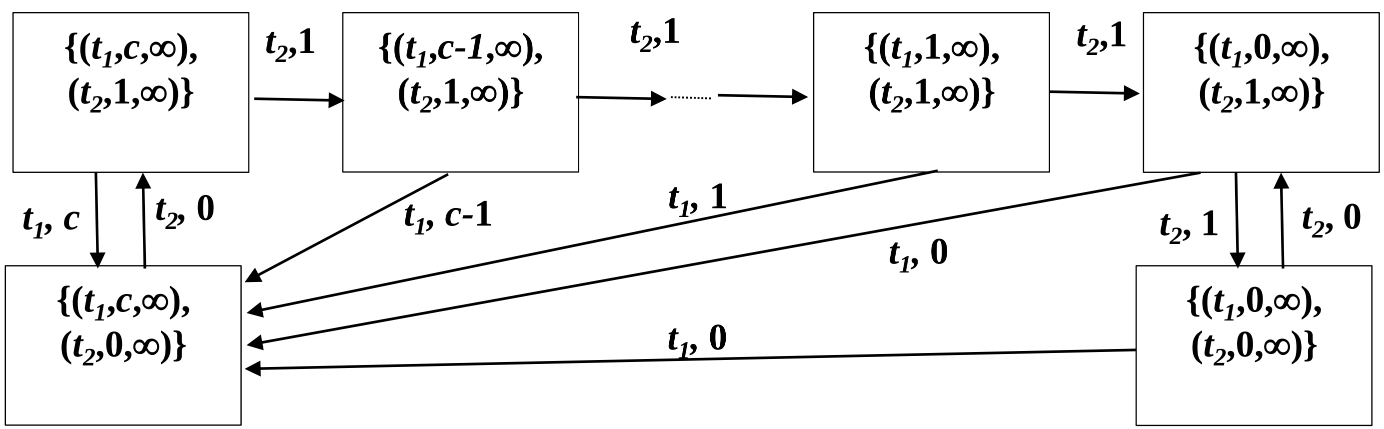




MATLAB
>>[predecesseur] = Dijkstra_predecessor_MACS([1:443],TERG_time,1)
>>MS=443
>>[seq,cost] = Dijkstra_seq_MACS(TERG,TERG_time,predecesseur,1,MS)

seq =[4 5 1 6 2 4 3 5 1 6 2 4 3 5 1 6 2
4 3 5 1 6 2 4 3 5 6 1 2 3]
duration = 87

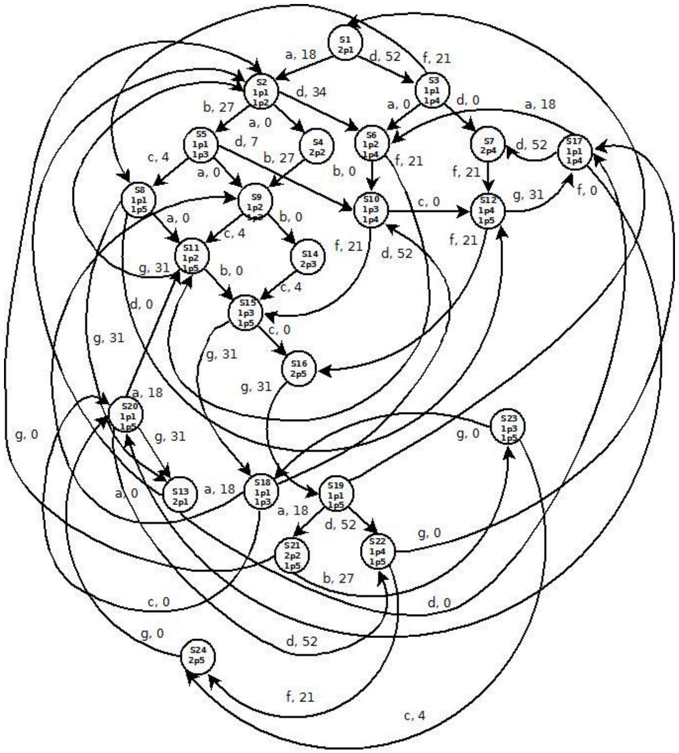




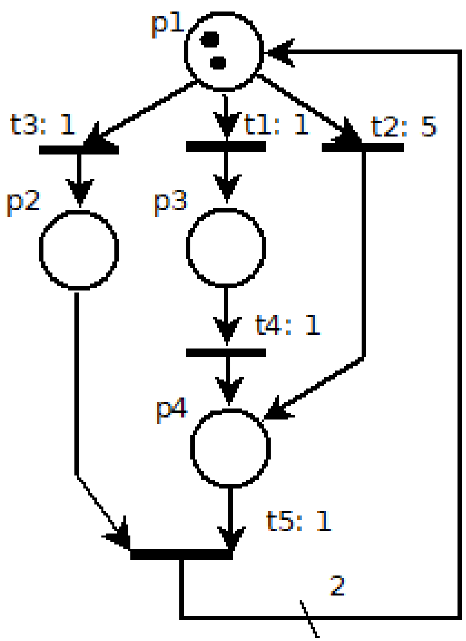
TERG tends to infinite size with respect to c

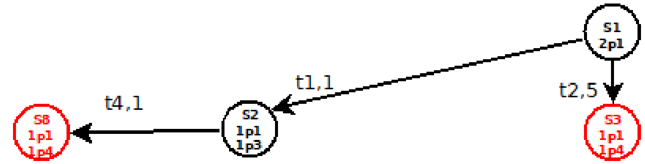
Outline

1. Problems and objectives
2. Modeling scheduling problems with T-TPN
3. Timed Extended Reachability Graph
4. **Approximated Timed Extended Reachability Graph**
5. Beam Search
6. Conclusion and future works



ApTERG

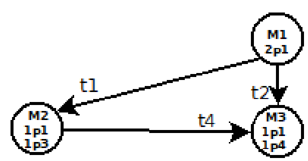


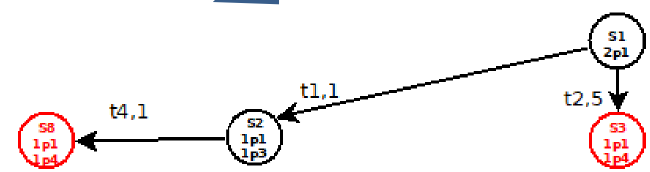


t1 : 0
t2 : 3
t3 : 0

t1 : 0
t2 : 0
t3 : 0

ApTERG(0)





$M_I = (2\ 0\ 0\ 0)^T = 2p_1$

$M_{ref} = (0\ 1\ 0\ 1)^T = 1p_2\ 1p_4$

$\delta = (1\ 5\ 1\ 1\ 1)^T$

$\Delta = (\infty\ \infty\ \infty\ \infty\ \infty)^T$

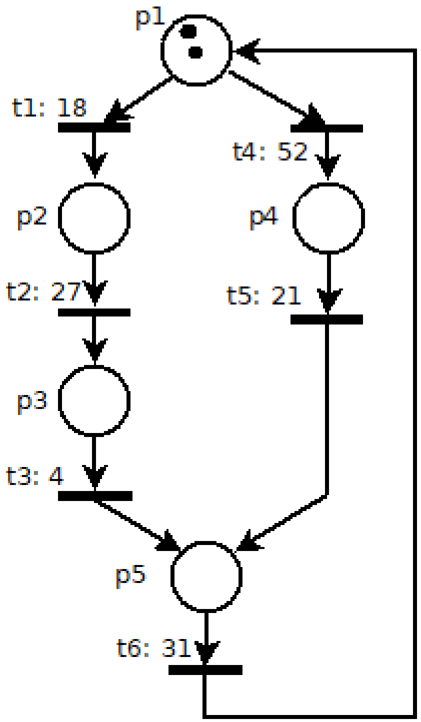
ApTERG(3.01)
to
ApTERG(∞)

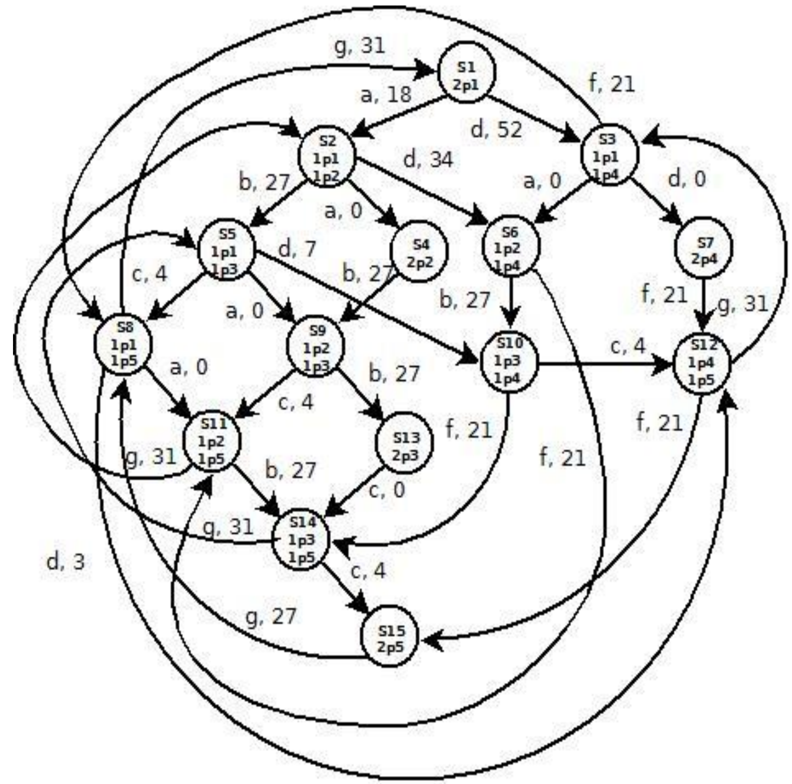
ApTERG(0)
to
ApTERG(2.99)

Approximated TERG (ApTERG)

Proposition: Let <G, STI, M_I, A> be a TPN system that behaves under earliest firing policy and its corresponding ApTERG(λ) = < $S_{ApE}(M_I)$, Ω_{ApE}, B_{ApE}, S_0 >.
If the timed trajectory $(\sigma, M_I) = M(0)$ [(t_{j1}, τ_1) > $M(1)$…. > $M(h-1)$ [(t_{jh}, τ_h) > $M(h)$ with $M(0) = M_I$ and $t_{jk} \in T$, k = 1,…,h, is feasible in TPN system, then a path S_0 S_1…S_h exists in <$S_{ApE}(M_I)$, Ω_{ApE}, B_{ApE}, S_0> st (1) $M(S_k) = M(k)$ for k = 0,…,h; (2) $\Omega_{ApE}(S_{k-1}, S_k) = t_{jk}$ and $|B_{ApE}(S_{k-1}, S_k).dt - (\tau_k - \tau_{k-1})| \leq \lambda.dt$ for k = 1,…,h.
If S_0 S_1…S_h is a path in ApTERG(λ) = <$S_{ApE}(M_I)$, Ω_{ApE}, B_{ApE}, S_0 > with S_0 the root node of the TERG then a timed trajectory $(\sigma, M_I) = M(0)$ [(t_{j1}, τ_1) > $M(1)$…. > $M(h-1)$ [(t_{jh}, τ_h) > $M(h)$ with (1) $M(k) = M(S_k)$, k = 0,…,h; (2) $t_{jk} = \Omega_{ApE}(S_{k-1}, S_k)$, k = 1,…,h; (3) $|\tau_1 - B_{ApE}(S_0, S_1).dt| \leq \lambda.dt$; (4) $|\tau_k - \tau_{k-1} - B_{ApE}(S_{k-1}, S_k).dt| \leq \lambda.dt$, k = 1,…,h; is feasible in TPN.

All feasible timed trajectories in time PN starting from M_I and executed under earliest firing policy are APPROXIMATED in ApTERG(λ) and the error is bounded depending on the granularity parameter λ





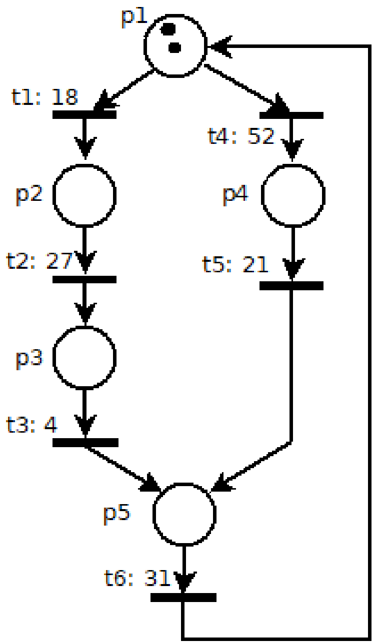
$M_I = (2\ 0\ 0\ 0\ 0)^T = 2p_1$

$M_{Ref} = (0\ 0\ 0\ 0\ 2)^T = 2p_5$

t_i	t_1 (a)	t_2(b)	t_3(c)	t_4(d)	t_5(f)	t_6(g)
δ_j	18	27	4	52	21	31
Δ_j	∞	∞	∞	∞	∞	∞

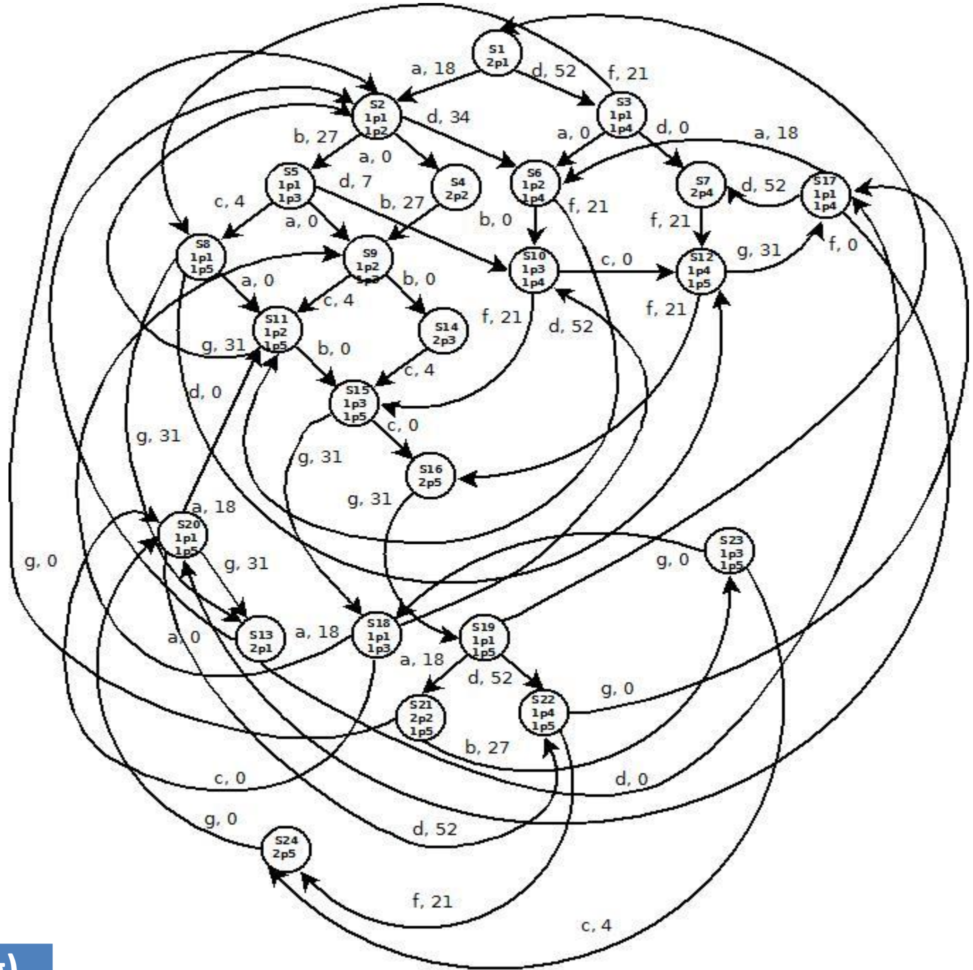
ApTERG(∞) = Untimed RG: 15 states

$$\sigma = T_1(18)\ T_2(45)\ T_3(49) T_4(52) T_4(73)$$



$M_I = (2\ 0\ 0\ 0\ 0)^T = 2p_1$

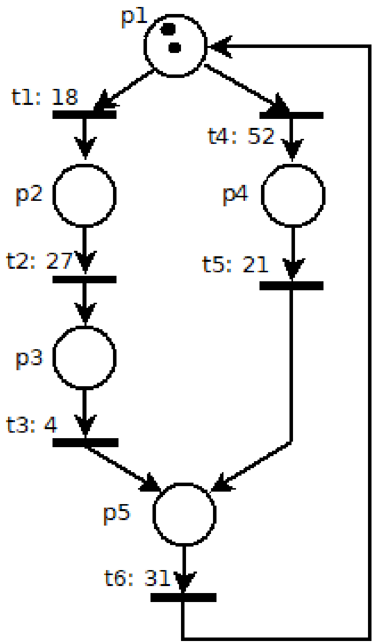
$M_{Ref} = (0\ 0\ 0\ 0\ 2)^T = 2p_5$



t_i	t_1 (a)	t_2(b)	t_3(c)	t_4(d)	t_5(f)	t_6(g)
δ_j	18	27	4	52	21	31
Δ_j	∞	∞	∞	∞	∞	∞

ApTERG(30) : 24 states

$\sigma = T_1(18)\ T_2(45)\ T_3(49) T_4(52) T_4(73)$



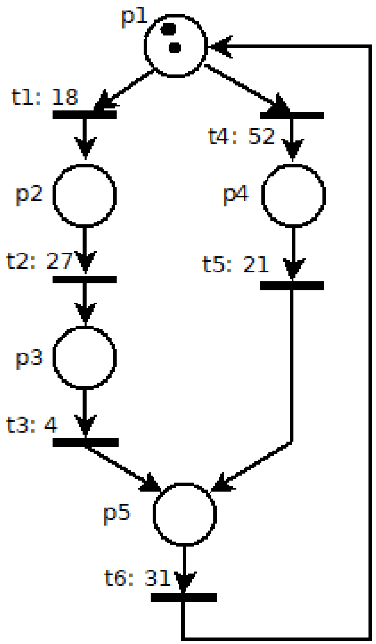
$M_I = (2\ 0\ 0\ 0\ 0)^T = 2p_1$

$M_{Ref} = (0\ 0\ 0\ 0\ 2)^T = 2p_5$

t_i	t_1 (a)	t_2(b)	t_3(c)	t_4(d)	t_5(f)	t_6(g)
δ_j	18	27	4	52	21	31
Δ_i	∞	∞	∞	∞	∞	∞

ApTERG(0) =) : 327 states

$$\sigma = T_1(18)T_1(18)T_2(45)$$
$$T_2(45)\ \ T_3(49)T_3(49)$$



Number of states in ApTERG(λ)

λ	∞	50	40	30	20	10	1	0
N_{ApE}	15	18	19	24	37	42	214	327

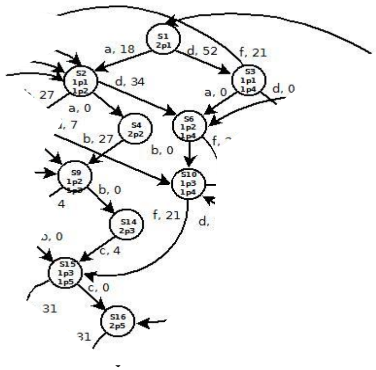
The number of states of ApTERG(λ) IS BOUNDED depending on λ

D. Lefebvre, Approximated Timed Reachability Graphs for the robust control of discrete event systems, *Discrete Event Dynamic Systems: theory and applications*, 29(1), 31-56, 2019

t_j	t_1	t_2	t_3	t_4	t_5	t_6
δ_j	18	27	4	52	21	31
Δ_j	∞	∞	∞	∞	∞	∞

Outline

1. Problems and objectives
2. Modeling scheduling problems with T-TPN
3. Timed Extended Reachability Graph
4. Approximated Timed Extended Reachability Graph
5. **Beam Search**
6. Conclusion and future works

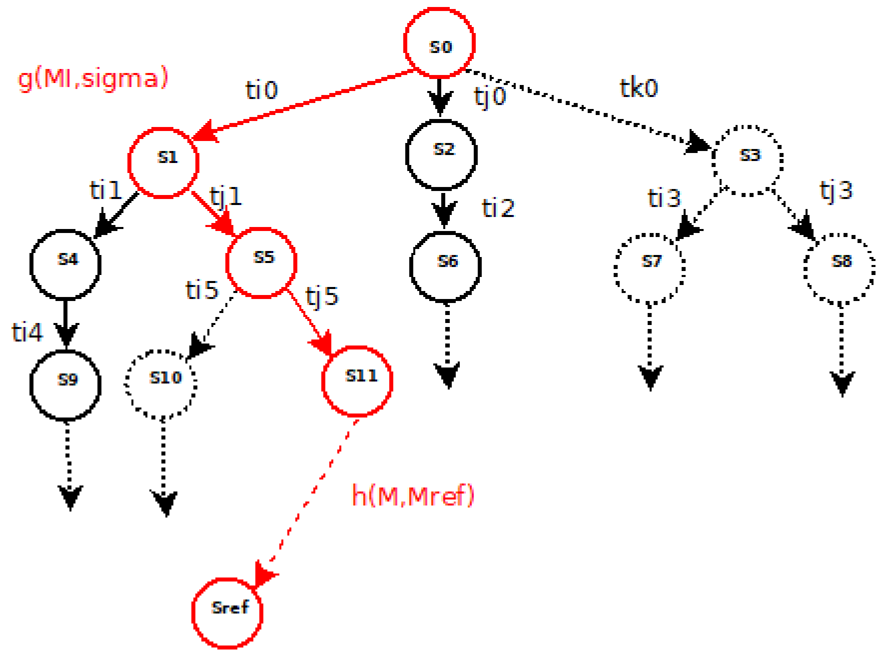


Local exploration in the untimed reachability graph driven by a cost function

$g(\sigma_1, M_I)$ = the cumulative duration of the already computed trajectory

$h(M, M_{ref})$ = estimation of the residual duration of (σ_2, M)

If $h(M, M_{ref})$ is a lower bound of the actual residual duration ($h(M, M_{ref})$ is said to be admissible), then the search find a least-coast trajectory



$$f(M_I, M, M_{ref}) = g(\sigma_1, M_I) + h(M, M_{ref})$$

Local exploration in the untimed reachability graph driven by a cost function

$g(\sigma_1, M_I)$ = the cumulative duration of the already computed trajectory

$h(M, M_{ref})$ = estimation of the residual duration of (σ_2, M)

$$f(M_I, M, M_{ref}) = g(\sigma_1, M_I) + h(M, M_{ref})$$

$$g(\sigma_1, M_I) = \tau_1 - \tau_0$$

$$(M_I, \tau_0) \xrightarrow{\sigma_1} (M, \tau)$$

DFS and WFS : $h(M, M_{ref}) = 0$

Another strategy : $h(M, M_{ref}) = \max\{ \chi^*(p_i, p_{ref}) : p_i \in P(M)\}$

….

A* algorithm

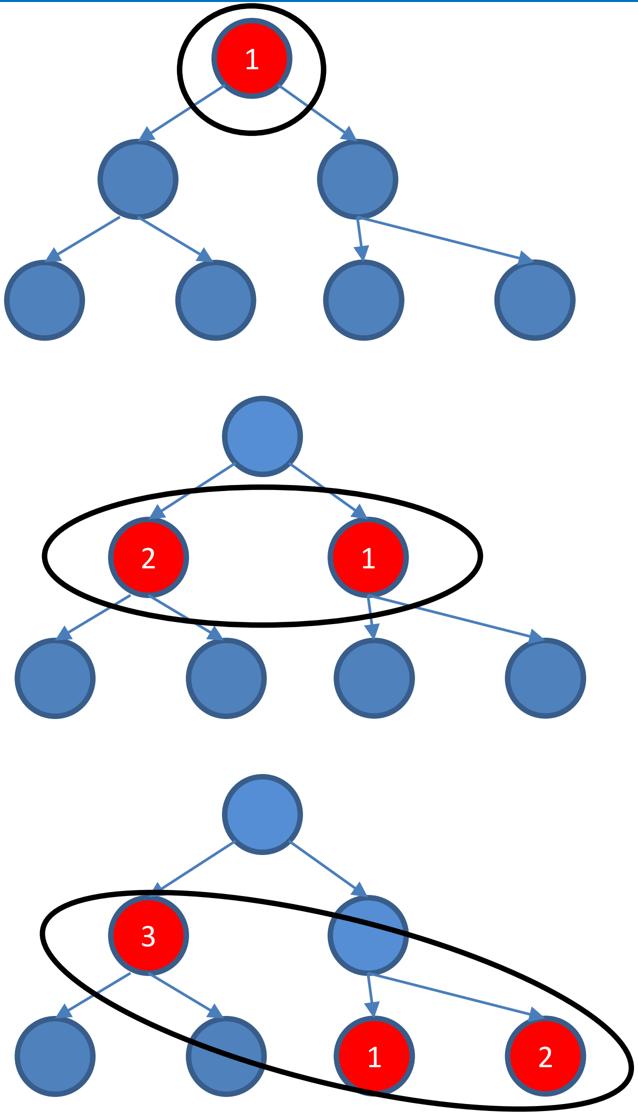
At each iteration:

(1) Rank the list of candidates

(2) Expand the best candidate of the list of candidates

(3) Store the new candidate(s)

Problem: the size of the list of candidates increases



P. E. Hart, N. J. Nilsson et B. Raphael, « A Formal Basis for the Heuristic Determination of Minimum Cost Paths », *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no 2, 1968, p. 100–107

Beam Search Algorithm

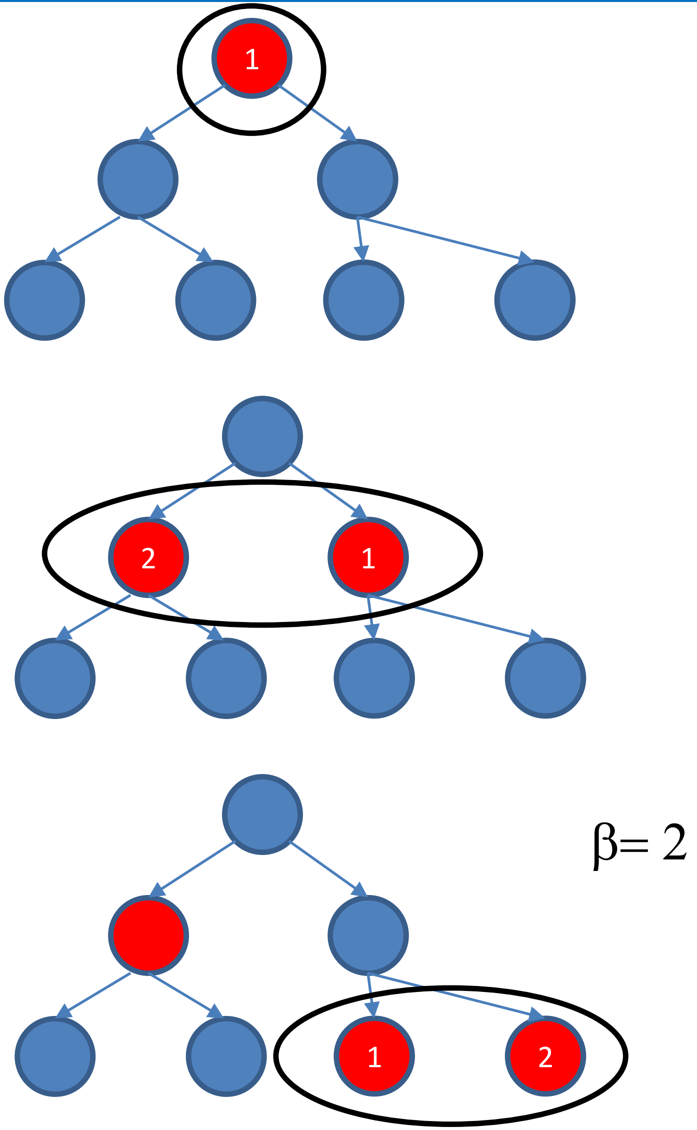
At each iteration:

(1) Rank the list of candidates

(2) Expand the best candidate of the list of candidates

(3) Store only the best β candidates selected with a global filter

Problem: Lack of diversity : Concentrate the candidates issued from the same parent



Russell, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ (USA): Prentice Hall.

Filtered Beam Search

At each iteration:

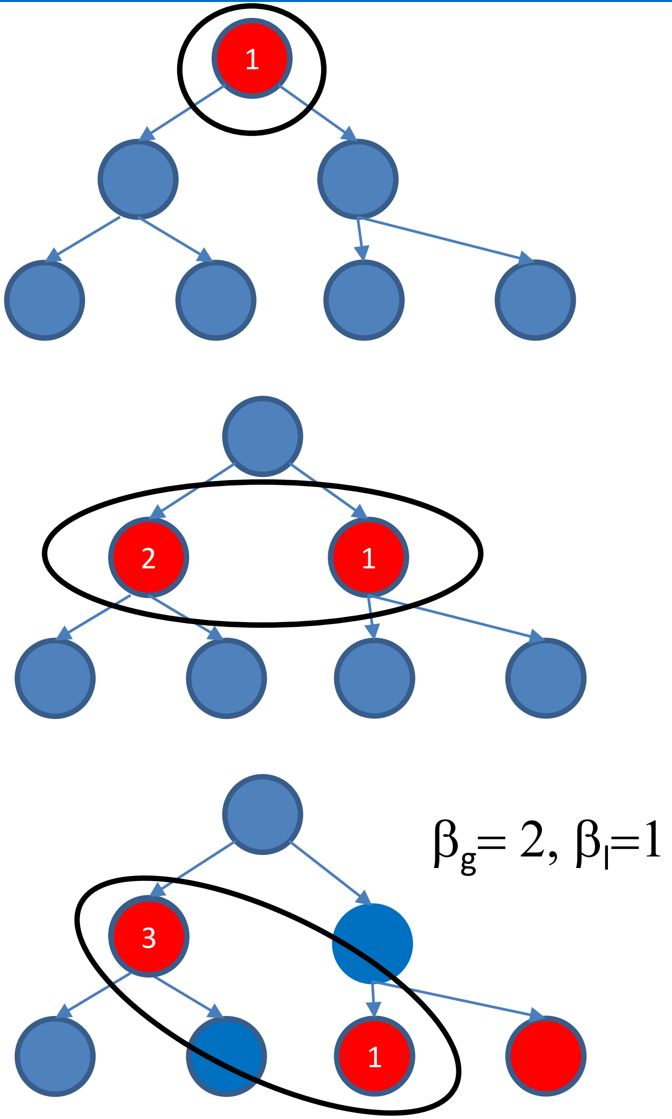
(1) Rank the list of candidates

(2) Expand the best candidate of the list of candidates

(3) A local filter limits the number β_l of childrens of a given parent

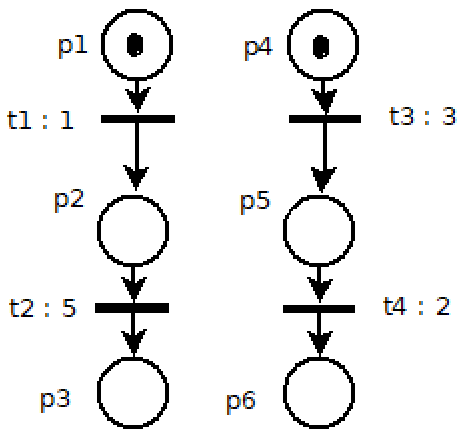
(4) A global filter selects the best β_g nodes at each iteration

Explores different regions of the RG



Mejía, G., & Niño, K. (2017). A new Hybrid Filtered Beam Search algorithm for deadlock-free scheduling of flexible manufacturing systems using Petri Nets. *Computers & Industrial Engineering*, 108, 165–176.

Exercice 3

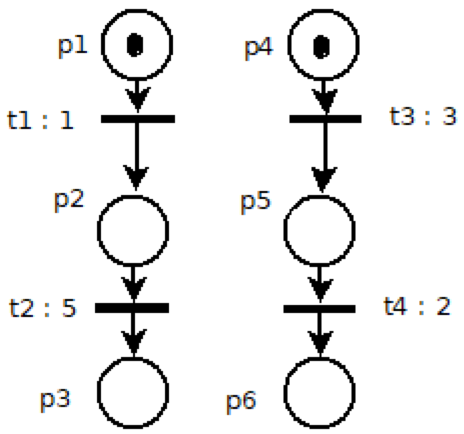


$M_I = p_1p_4$
$M_D = p_3p_6$

σ	t1	t3					
g/h/f	1/0/1	3/0/3					
σ	…	…	…	…	…	…	
g/h/f	…	…	…	…	…	…	
σ							
g/h/f							
σ							
g/h/f							
σ							
g/h/f							

A* with $h(M,M_{ref}) = 0$

Exercice 3

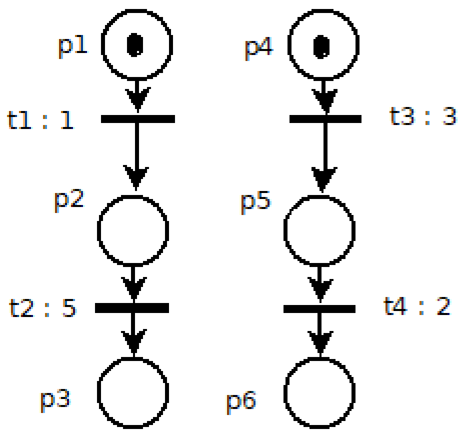


$M_I = p_1p_4$
$M_D = p_3p_6$

σ	t1	t3					
g/h/f	1/0/1	3/0/3					
σ	t3	t1t3	t1t2				
g/h/f	3/0/3	3/0/3	6/0/6				
σ	…	…	…	…	…	…	
g/h/f	…	…	…	…	…	…	
σ							
g/h/f							
σ							
g/h/f							

A* with *h(M,M$_{ref}$) = 0*

Exercice 3

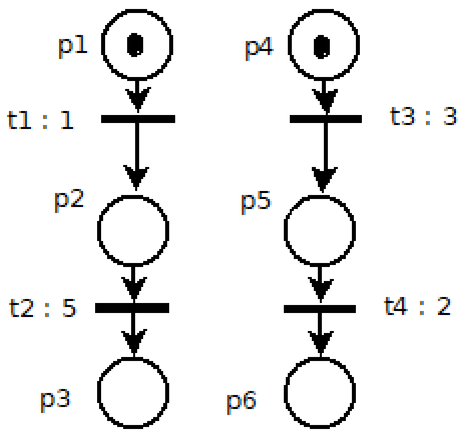


$M_I = p_1p_4$
$M_{MD} = p_3p_6$

σ	t1	t3					
g/h/f	1/0/1	3/0/3					
σ	t3	t1t3	t1t2				
g/h/f	3/0/3	3/0/3	6/0/6				
σ	t3t1	t1t3	t3t4	t1t2			
g/h/f	3/0/3	3/0/3	5/0/5	6/0/6			
σ	t1t3	t3t4	t3t1t4	t1t2	t3t1t2		
g/h/f	3/0/3	5/0/5	5/0/5	6/0/6	8/0/8		
σ	…	…	…	…	…	…	
g/h/f	…	…	…	…	…	…	

A* with $h(M,M_{ref}) = 0$

Exercice 3



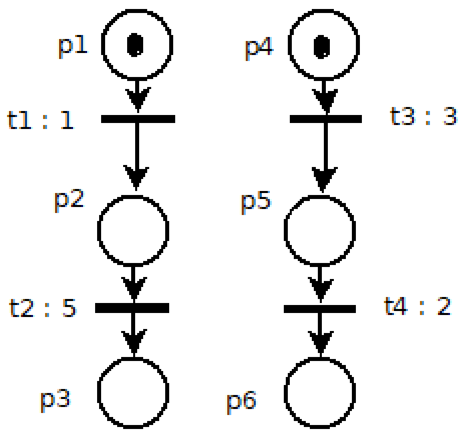
$M_I = p_1p_4$

$M_D = p_3p_6$

σ	t1	t3					
g/h/f	1/5/6	3/6/9					
σ	t1t3	t3	t1t2				
g/h/f	3/5/8	3/6/9	6/5/11				
σ	t1t3t2	t3	t1t3t4	t1t2			
g/h/f	6/2/8	3/6/9	5/5/10	6/5/11			
σ	t1t3t2t4	t3	t1t3t4	t1t2			
g/h/f	6/0/6	3/6/9	5/5/10	6/5/11			
σ							
g/h/f							

A* with $h(M,M_{ref}) = max\{ \chi^*(p_i, p_{ref}) : p_i \in P(M)\}$

Exercice 3



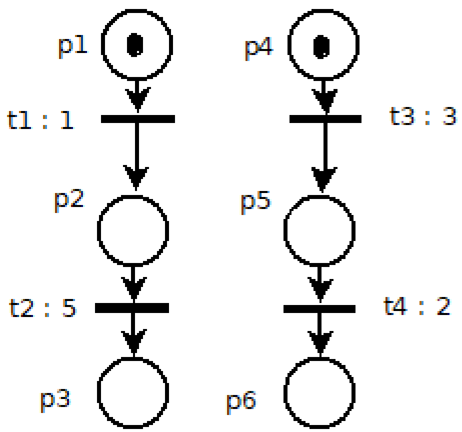
$M_I = p_1 p_4$

$M_D = p_3 p_6$

σ	t1	t3		X	X	X	X
g/h/f	1/5/6	3/6/9		X	X	X	X
σ	t3	t1t3	t1t2	X	X	X	X
g/h/f	3/6/9	3/5/8	6/5/11	X	X	X	X
σ	t1t3t2	t3	t1t3t4	X	X	X	X
g/h/f	6/2/8	3/6/9	5/5/10	X	X	X	X
σ	t1t3t2t4	t3	t1t3t4	X	X	X	X
g/h/f	6/0/6	3/6/9	5/5/10	X	X	X	X
σ				X	X	X	X
g/h/f				X	X	X	X

Beam with $\beta = 3$ and with $h(M, M_{ref}) = \max\{ \chi^*(p_i, p_{ref}) : p_i \in P(M)\}$

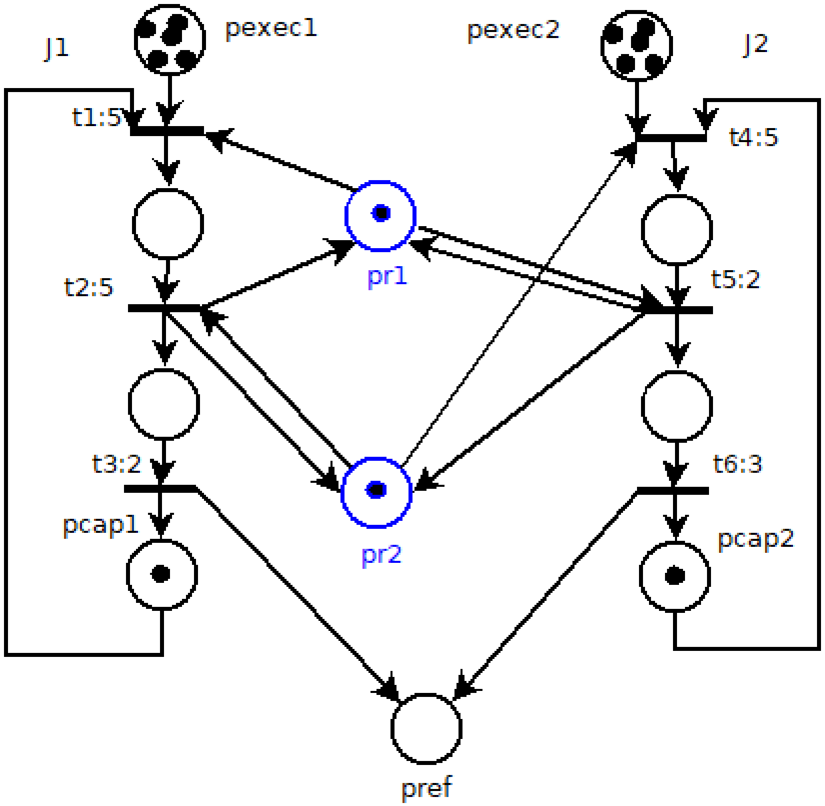
Exercice 3



$M_I = p_1p_4$
$M_D = p_3p_6$

σ	t1	t3		X	X	X	X
g/h/f	1/5/6	3/6/9		X	X	X	X
σ	t1t3	t3	t3	X	X	X	X
g/h/f	3/5/8	3/6/9	3/6/9	X	X	X	X
σ	t1t3t2	t3		X	X	X	X
g/h/f	6/2/8	3/6/9		X	X	X	X
σ	t1t3t2t4	t3		X	X	X	X
g/h/f	6/0/6	3/6/9		X	X	X	X
σ				X	X	X	X
g/h/f				X	X	X	X

Beam with $\beta_g = 3$ and $\beta_l = 1$
and with $h(M,M_{ref}) = max\{\ \chi^*(p_i, p_{ref}) : p_i \in P(M)\}$



MATLAB

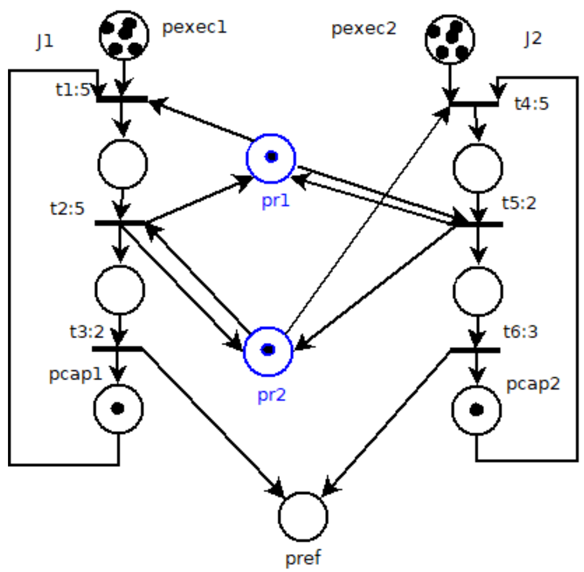
A* ???

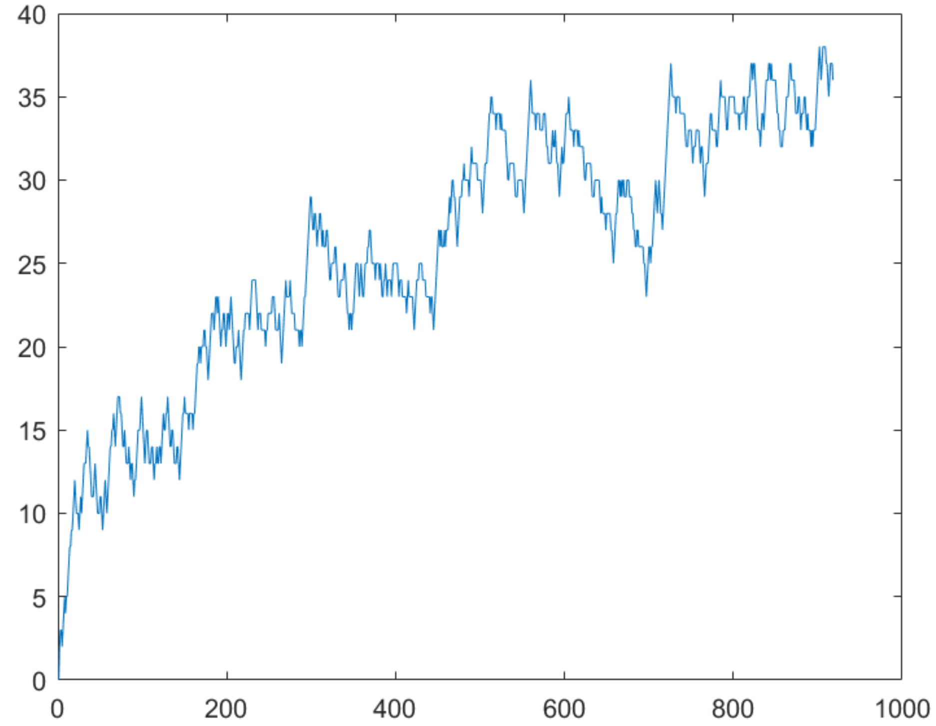
Beam with $\beta = 5$???

Beam with $\beta_g = 5$ and $\beta_l = 2$???

Beam with $\beta_g = 5$ and $\beta_l = 1$???

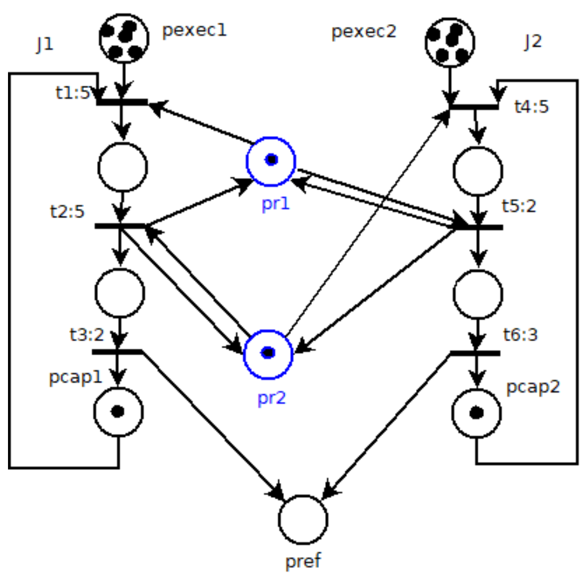
MATLAB : A*
>>MD = [0 1 0 0 0 1 0 0 1 1 10]'
>> [seq,criteria]=Astar_MACS(Wpr,Wpo,Pjob,MI,MD,delta)
Duration of the obtained sequence = 87
917 explored candidates

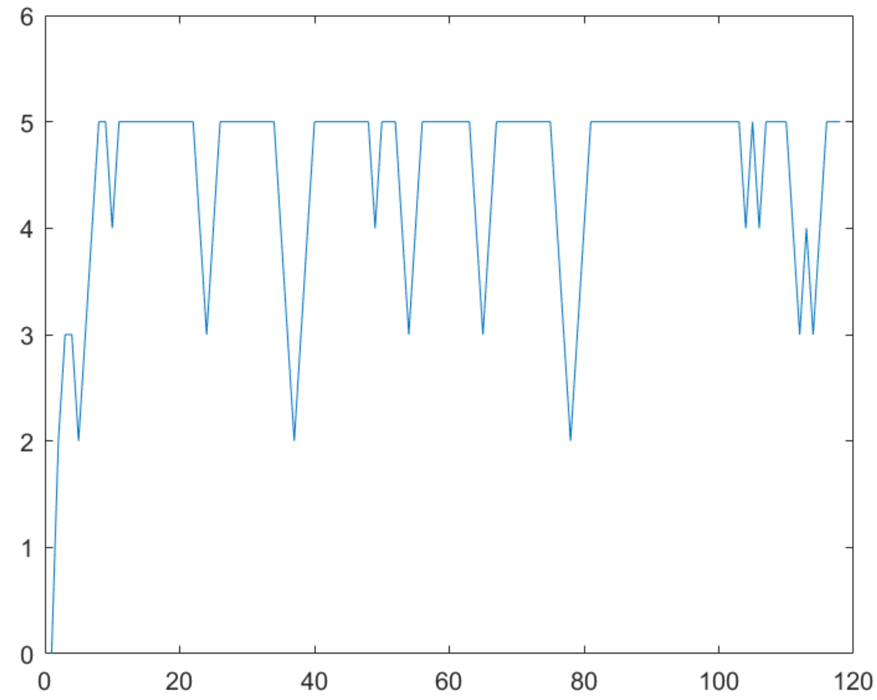




Size of list OPEN

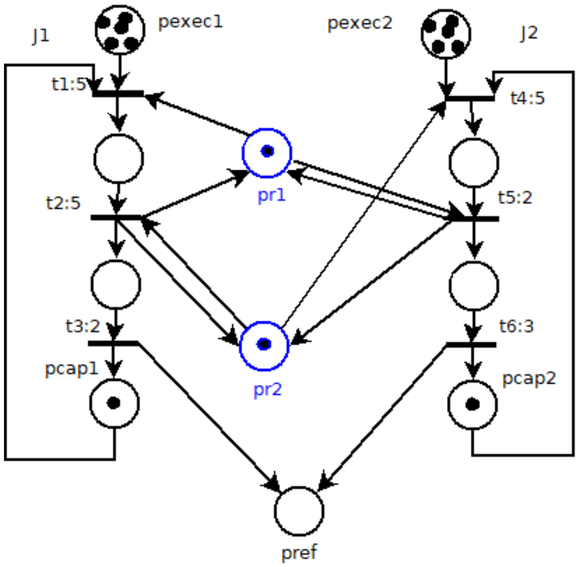
MATLAB : Beam with β_g = 5 and β_l = 2
>>MD = [0 1 0 0 0 1 0 0 1 1 10]’
>> [seq,criteria]=beam_MACS(Wpr,Wpo,Pjob,MI,MD,delta,5,2)
Duration of the obtained sequence = 88
118 explored candidates





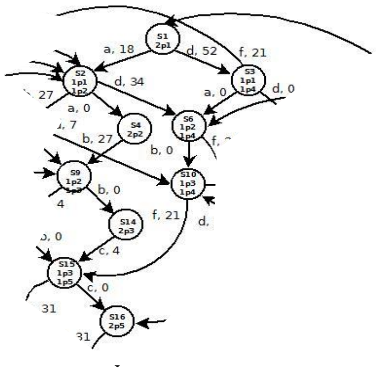
Size of list OPEN

MATLAB : Beam with β_g = 5 and β_l = 1
>>MD = [0 1 0 0 0 1 0 0 1 1 10]'
>> [seq,criteria]=beam_MACS(Wpr,Wpo,Pjob,MI,MD,delta,5,1)
Duration = X
4 explored candidates
Algorithm fails



Outline

1. Problems and objectives
2. Modeling scheduling problems with T-TPN
3. Timed Extended Reachability Graph
4. Approximated Timed Extended Reachability Graph
5. Beam Search
6. **Conclusion and future works**

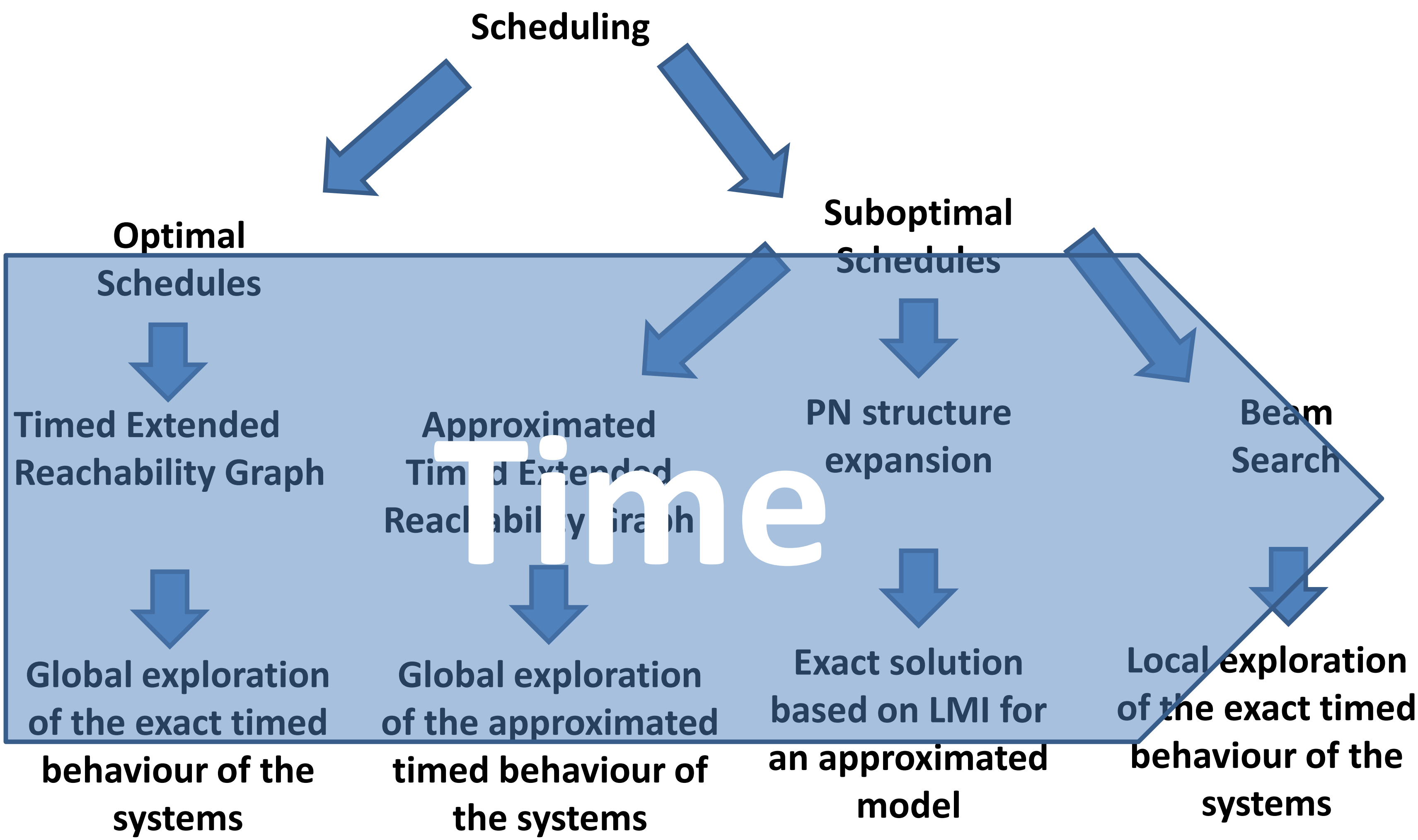


Conclusions and future works

Scheduling

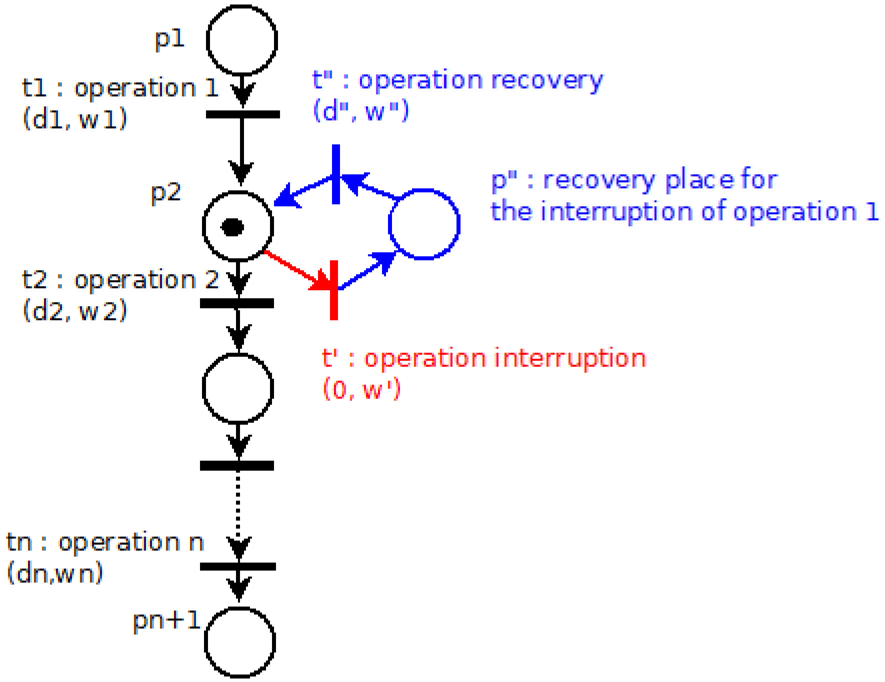
- **Optimal Schedules**
  - **Timed Extended Reachability Graph**
    - **Global exploration of the exact timed behaviour of the systems**
- **Suboptimal Schedules**
  - **Approximated Timed Extended Reachability Graph**
    - **Global exploration of the approximated timed behaviour of the systems**
  - **PN structure expansion**
    - **Exact solution based on LMI for an approximated model**
  - **Beam Search**
    - **Local exploration of the exact timed behaviour of the systems**

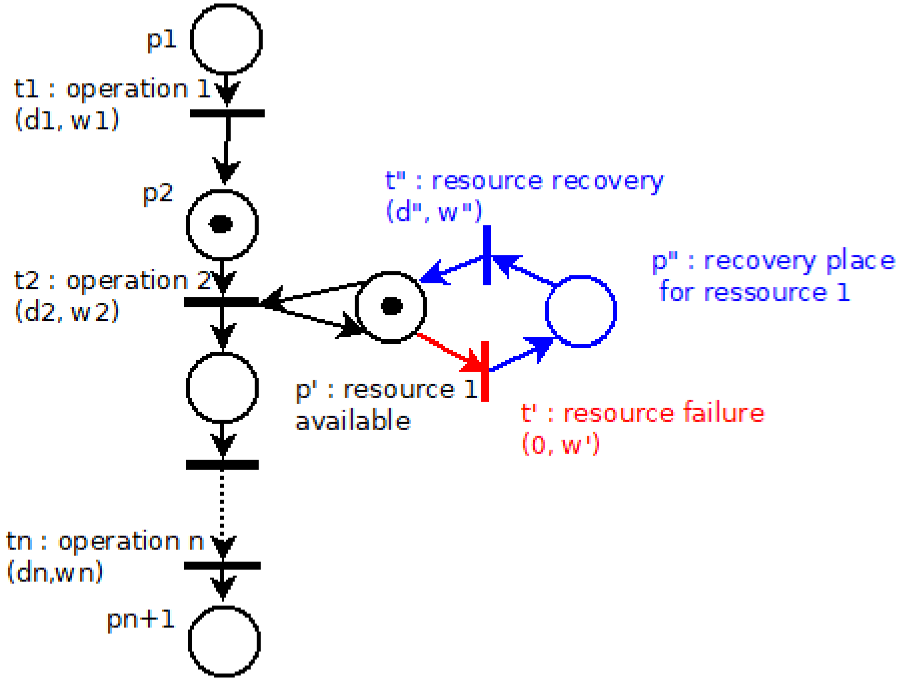
Conclusions and future works



Future works

Robust scheduling in uncertain environments





Operation interruption **Unreliable resource**

Future works

Robust scheduling in uncertain environments

2) Local exploration

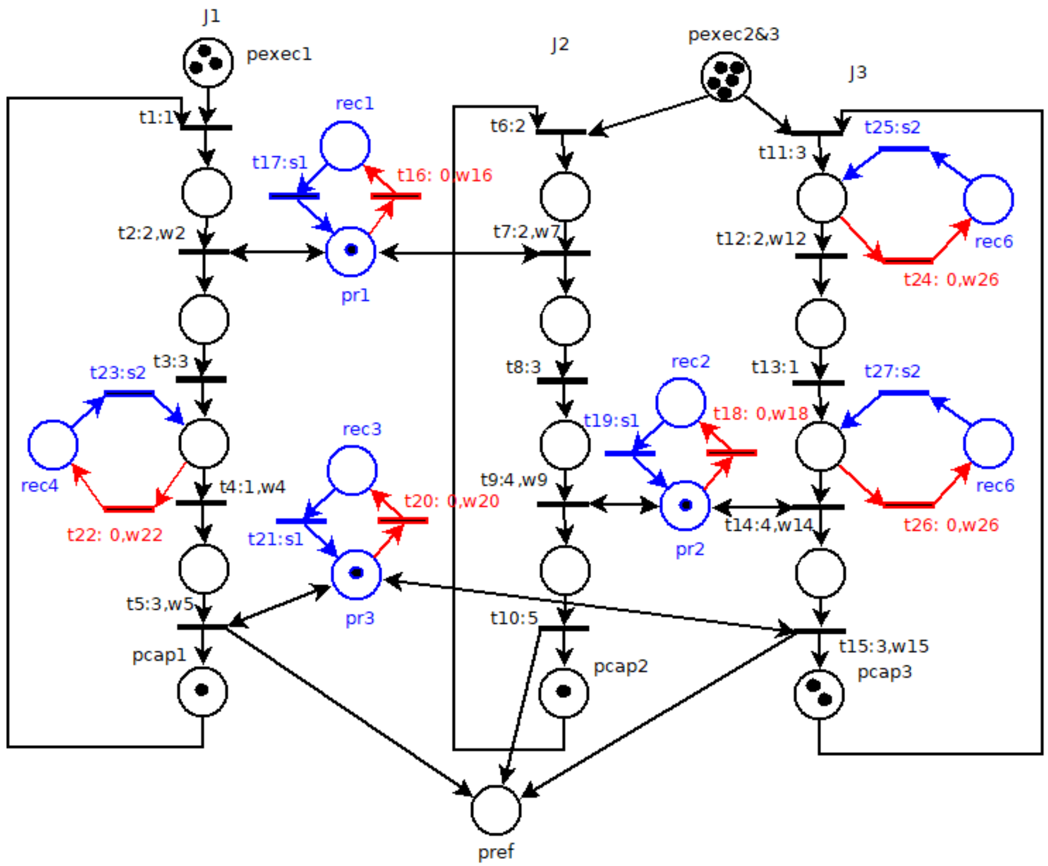
$f(M_I, M, M_{ref}) =$

$+ \ (\sigma_1, M_I)$ **duration**

$+ g_r(\sigma_1, M_I)$ **risk**

$+ h_d(M, M_{ref})$ **duration**

$+ h_r(M, M_{ref})$ **risk**



Thanks to my colleagues and students

Rabah Ammour
Ghassen Cherif
Edouard Leclercq
Alioune Mbaye
Yoann Pigné
Sara Rachidi
Eric Sanlaville
Marwa Taleb
…

Some of our references in control theory

Lefebvre D. and Leclercq E. Control design for trajectory tracking with untimed Petri nets, *IEEE Trans. Aut. Contr.*, vol. 60(7), pp. 1921-1926, 2015

Lefebvre D., Approaching minimal time control sequences for timed Petri nets, *IEEE Trans. Automation Science and Engineering*, vol. 13, no. 2, pp. 1215-1221, 2016.

C. Daoui, D. Lefebvre, Control design for untimed Petri nets using Markov Decision Processes, *Journal Operations Research and Decisions*, 27(4): 28-43, 2017.

Taleb M., Leclercq E., Lefebvre D., Control Design of Timed Continuous Petri Nets via Model Predictive Constant Control, *International Journal of Control*, 91(8): 1962-1978, 2018.

Lefebvre D., Near-optimal scheduling for Petri net models with forbidden markings, *IEEE Trans. On Automatic Control*, 63(8): 2550-2557, 2018.

D. Lefebvre, Algorithms of reduced complexity to design control sequences for untimed Petri nets in varying and uncertain environments, *Journal of Systems and Control Engineering*, Vol. 232, issue 6, pp: 638-651, 2018.

Lefebvre D., Daoui C., Control design for bounded Partially Controlled TPNs using Timed Extended Reachability Graphs and MDP, *IEEE Trans. Man, Cybernetics and Systems*, early access, DOI: 10.1109/TSMC.2018.2817492, 2018.

Thank you !!!

dimitri.lefebvre@univ-lehavre.fr

http://www.univ-lehavre.fr/recherche/greah/

RÉGION NORMANDIE

UNION EUROPEENNE

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE