“Approximation Methods based on Net-Driven Decompositions”

Bruno Baynat
University P.&M. Curie
Paris, France

Bruno.Baynat@masi.ibp.fr

Javier Campos
University of Zaragoza
Zaragoza, Spain

jcampos@posta.unizar.es

PNPM’97
Saint Malo, France
June 3, 1997

Objective of the tutorial

- To present some…

ideas an **examples** on

approximation techniques that
try to overcome the state
explosion problem within a

divide & conquer strategy and
with a strong exploitation of

structural knowledge of the
underlying Petri net model
for both the

decomposition & solution phases.

Outline

- Principles of approximation techniques based on decomposition
- A technique with non-PF subsystems and PF skeleton
- A technique with non-PF subsystems and non-PF skeleton
- Final comments and forthcoming research efforts

Principles of approximation techniques based on decomposition

- **Basic idea**:

  reduce the complexity of the analysis of a complex system

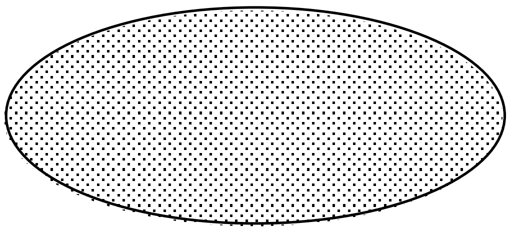
- **when**

  – the system is too complex / big to be solved by any exact analytical technique
  – a simulation is too long (essentially if many different configurations must be tested or it must be included in some optimization procedure)
  – some insights about the internal behaviour of subsystems are wanted (writing equations might help)

Principles of approximation techniques based on decomposition

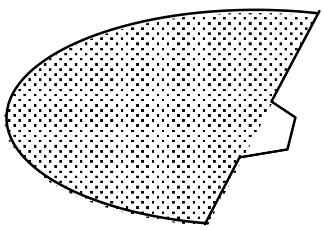
❖ Principle:

– decompose the system into some subsystems

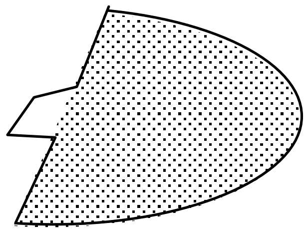


original system

state space size: *n*



two subsystems
state space size of each: $n / 10$
(for example)
(i.e., one order of magnitud less)



– reduce the analysis of the whole system by those of the subsystems in isolation

if the solution technique was, e.g., $O(n^3)$ on the state space size n, the cost of solving the isolated subsystems would be $O(n^3 / 1000)$, i.e. three orders of magnitud less…

Principles of approximation techniques based on decomposition

- Advantages:
  - drastical reduction of complexity and computational requirements
  - enables to extend the class of system that can be solved by analytical techniques

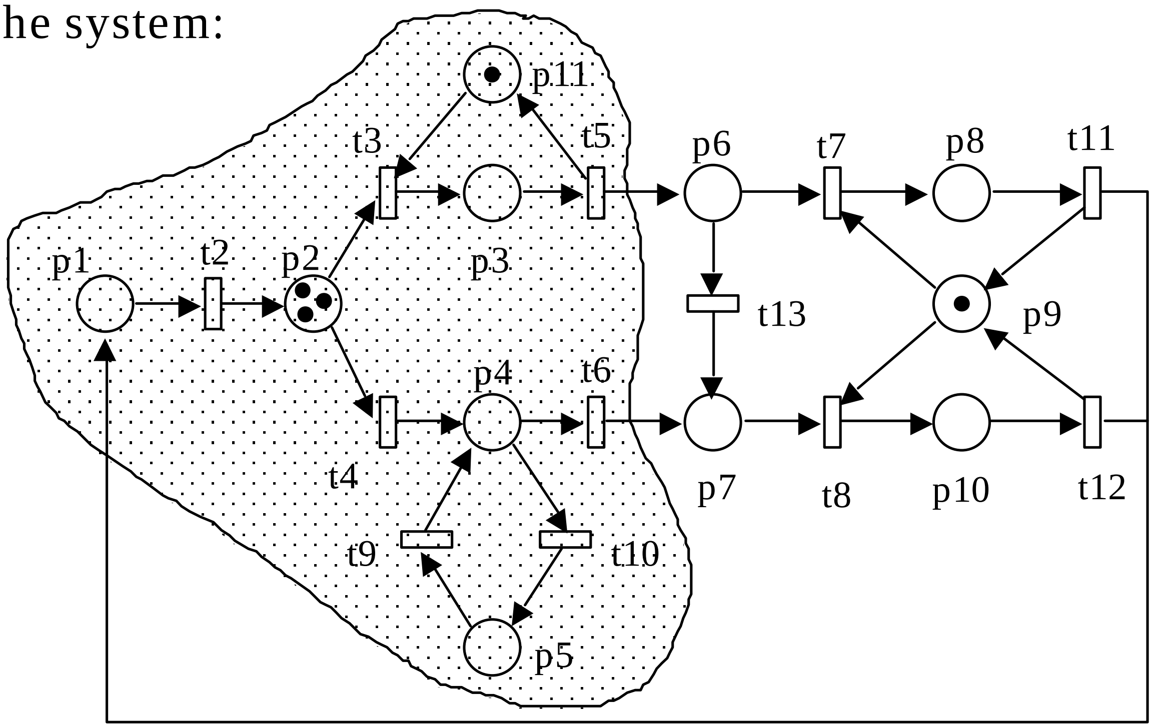
- Problems and limitations
  - Decomposition is not easy!
    - “net-driven” means to use structural information of the net model to assure that “good” qualitative properties are preserved in the isolated subsystems (e.g., liveness, boundedness…)
  - Approximation is not exact!
    - problem of error estimation or at least bounding the error
  - Accurate techniques are usually very especific to particular problems ==> need of expertise to select the adequate technique…

Principles of approximation techniques based on decomposition

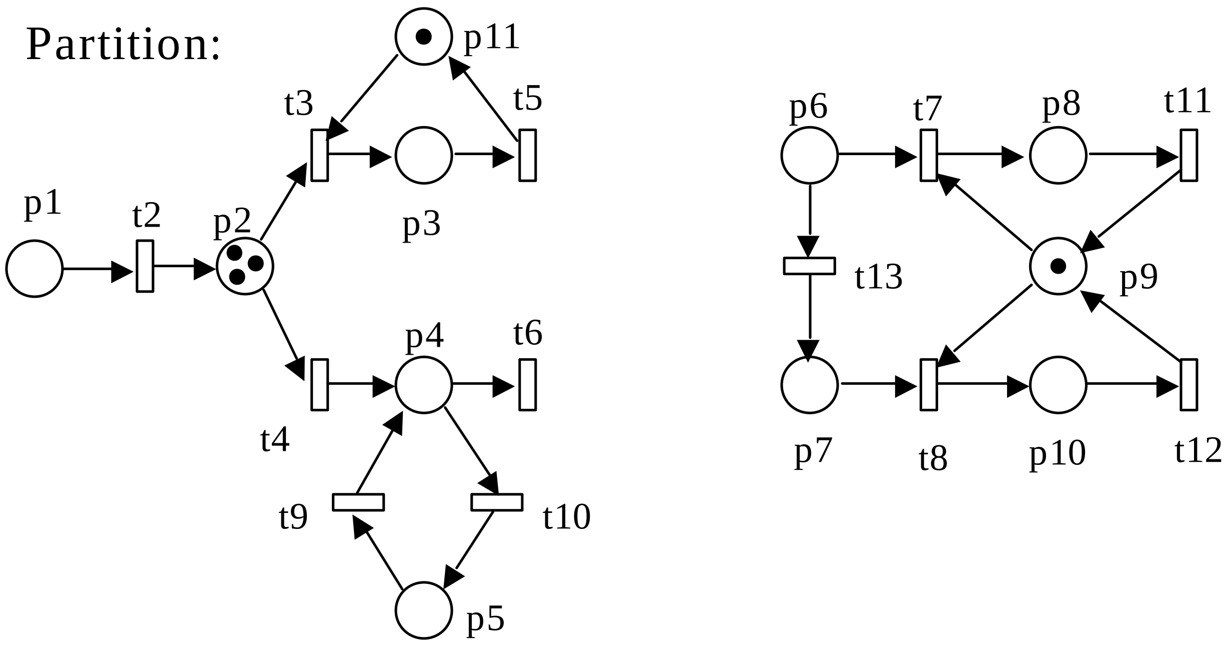
- **Steps** in an approximation technique based on decomposition:
  - Partition of the system into subsystems:
    - definition of rules for decomposition
    - consideration of logical (qualitative) properties that must (or can) be preserved
  - Characterization of subsystems in isolation:
    - definition of unknowns and variables
    - decisions related with consideration of mean variables or higher order moments of involved random variables
    - consideration or not of the “outside world”
    - need of a skeleton (high level view of the model) and characteristics considered in it
  - Estimation of the unknown parameters:
    - writing equations among unknowns
    - direct or iterative technique (in this case, definition of fixed point equations)
    - considerations on existence and uniqueness of solution
    - computational algorithm for solving the fixed point equation (implementation aspects, convergence aspects)

Example: flow equivalent aggregation in GSPN's

The system:



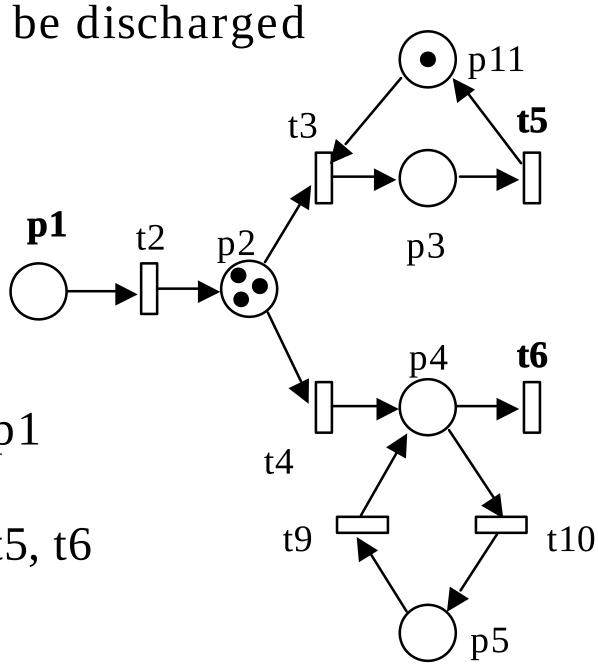
Partition:



Example: flow equivalent aggregation in GSPN's

❖ Characterization of subsystems. Behaviour is characterized by:

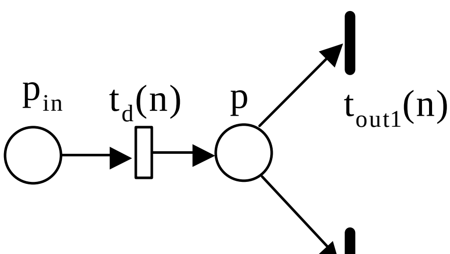
- path a token takes in the Petri net (what percetage leave through t5 and t6)
- time it takes a token to be discharged



• way-in places: p1

• sink transitions: t5, t6

❖ Reduction of the subsystem:



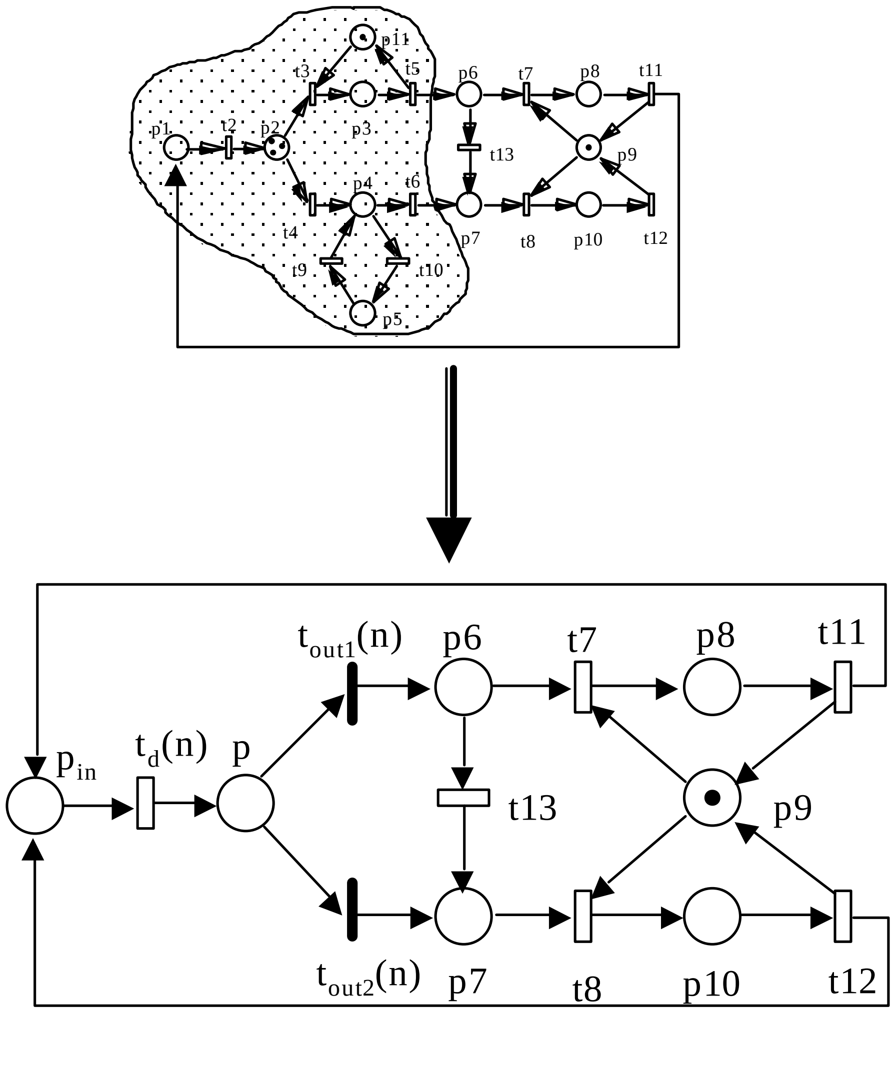
• routing rates of $t_{out1}(n)$ and $t_{out2}(n)$?

• service rate of $t_d(n)$?

(marking dependent: $n=M(p_{in})$

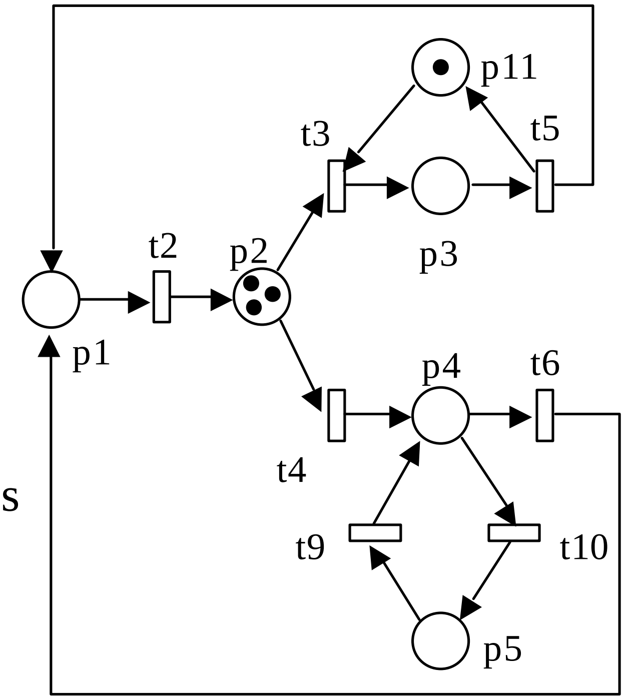
Example: flow equivalent aggregation in GSPN's

❖ Aggregated system:



Example: flow equivalent aggregation in GSPN's

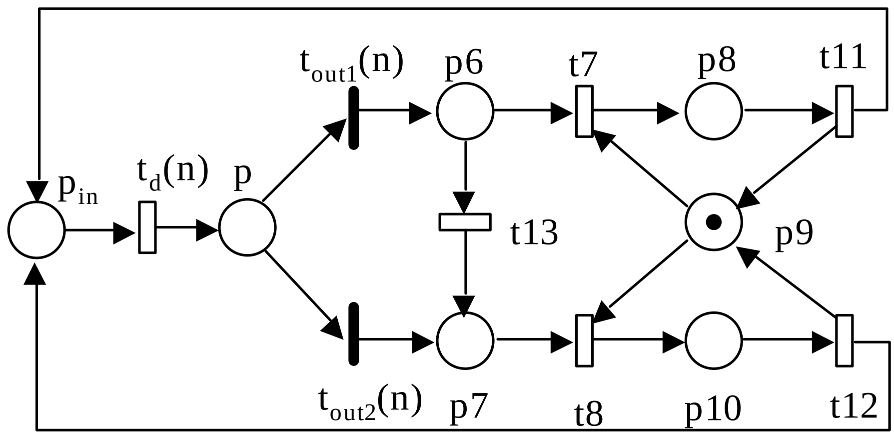
- Estimation of the unknown parameters:
  - Analyze the subnet in isolation with constant number of tokens
    - delay and routing are dependent on the number of tokens in the system
    - compute delay and routing for all possible populations



Parameters of the subsystem in isolation			
# tokens	v_5	v_6	thrput
1	0.500	0.500	0.400
2	0.431	0.569	0.640
3	0.403	0.597	0.780
4	0.389	0.611	0.863
5	0.382	0.618	0.914

Example: flow equivalent aggregation in GSPN's

- When the subnet is substituted back, routing and delay are going to be state dependent ($n=M(p_{in})$)



Comparison of State Spaces & throughput					
#tokens	# states		throughput		%error
	aggregat	original	aggregat	original	
1	5	9	0.232	0.232	0.00
2	12	41	0.381	0.384	0.78
3	22	131	0.470	0.474	0.84
4	35	336	0.521	0.523	0.38
5	51	742	0.548	0.547	<0.10

Example: flow equivalent aggregation in GSPN's

❖ Limitations:

– **Assumption**: the service time depends only on the number of customers which are currently present in the subsystem.

  ◆ The behaviour of the subsystem is assumed independent of the arrival process

– It is exact for product-form queueing networks.

– The error is small if in the original model:

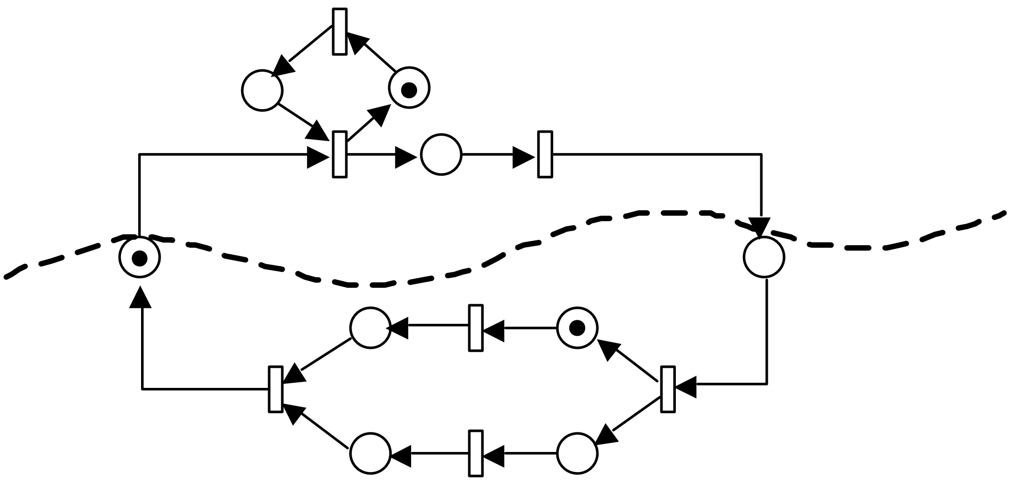
  ◆ the arrivals to the subsystem are "close" to Poisson arrivals and

  ◆ the processing times are approximately exponential

– On the other hand, the error can be very large if

  ◆ there exist internal loops in a subnet or

  ◆ there exist trapped tokens in a fork-join or…



A technique with non-PF subsystems and PF skeleton

- "Syntactical" framework:
  - Closed (multiclass) queueing networks with synchronization mechanisms

- General rule for partition:
  - Include inside each subsystem non-PF primitives
  - The skeleton is a PF network where the subsystems are substituted with "aggregated" exponential stations

- Characterization of subsystems:
  - Service rates of aggregated stations
  - Conditional throughputs of the subsystems in the original system

- Estimation of the parameters:
  - Aggregation technique, Marie's method…

---> Here the slides of Bruno…

A technique with non-PF subsystems and non-PF skeleton

- “Syntactical” framework:
  - Stochastic (exponential) Petri nets
  - Subclasses: marked graphs (≅FJ-QN / B) and some extensions

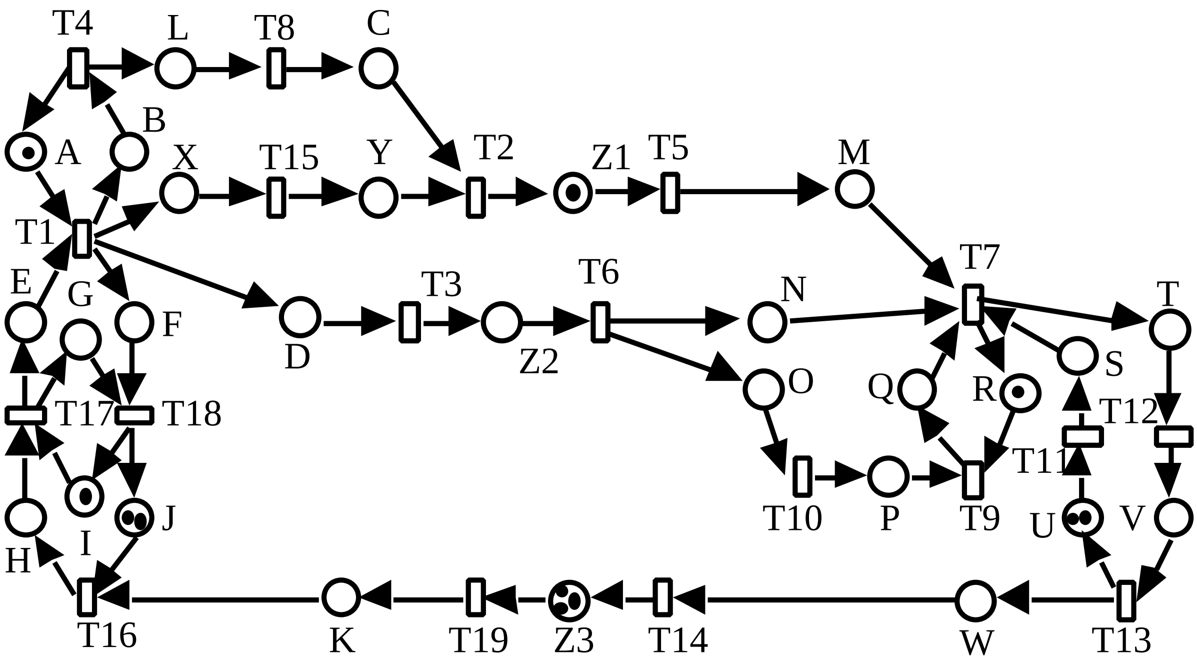
- General rule for partition:
  - Arbitrary cut of the system into pieces
  - The skeleton is a smaller SPN

- Characterization of subsystems:
  - Service times of aggregated stations
  - Response time of the subsystems in the original system

- Estimation of the parameters:
  - Response time approximation…

Stochastic marked graphs case (isomorphous to FJ-QN/B)

❖ An example

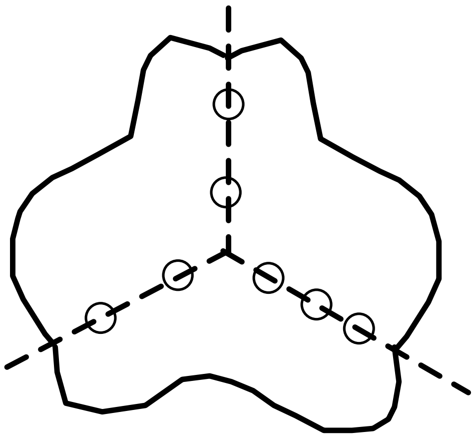


Exact analysis:

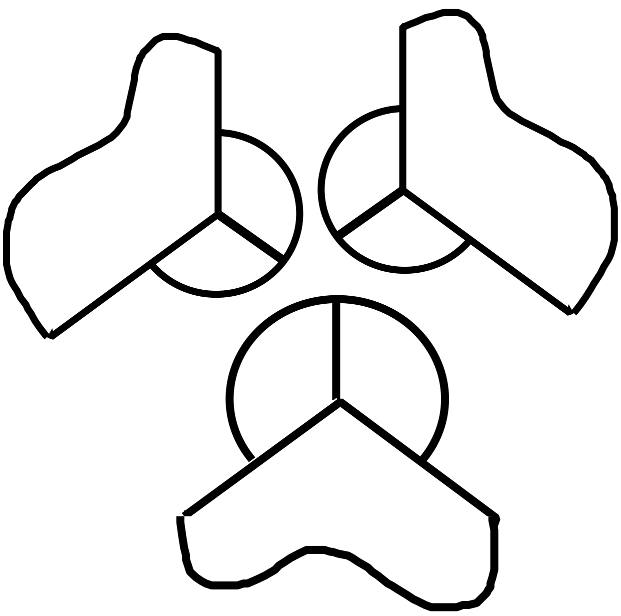
underlying CTMC --> State explosion problem

(89358 states)

Partition of the system



Original system (SMG)
Arbitrary cut through places

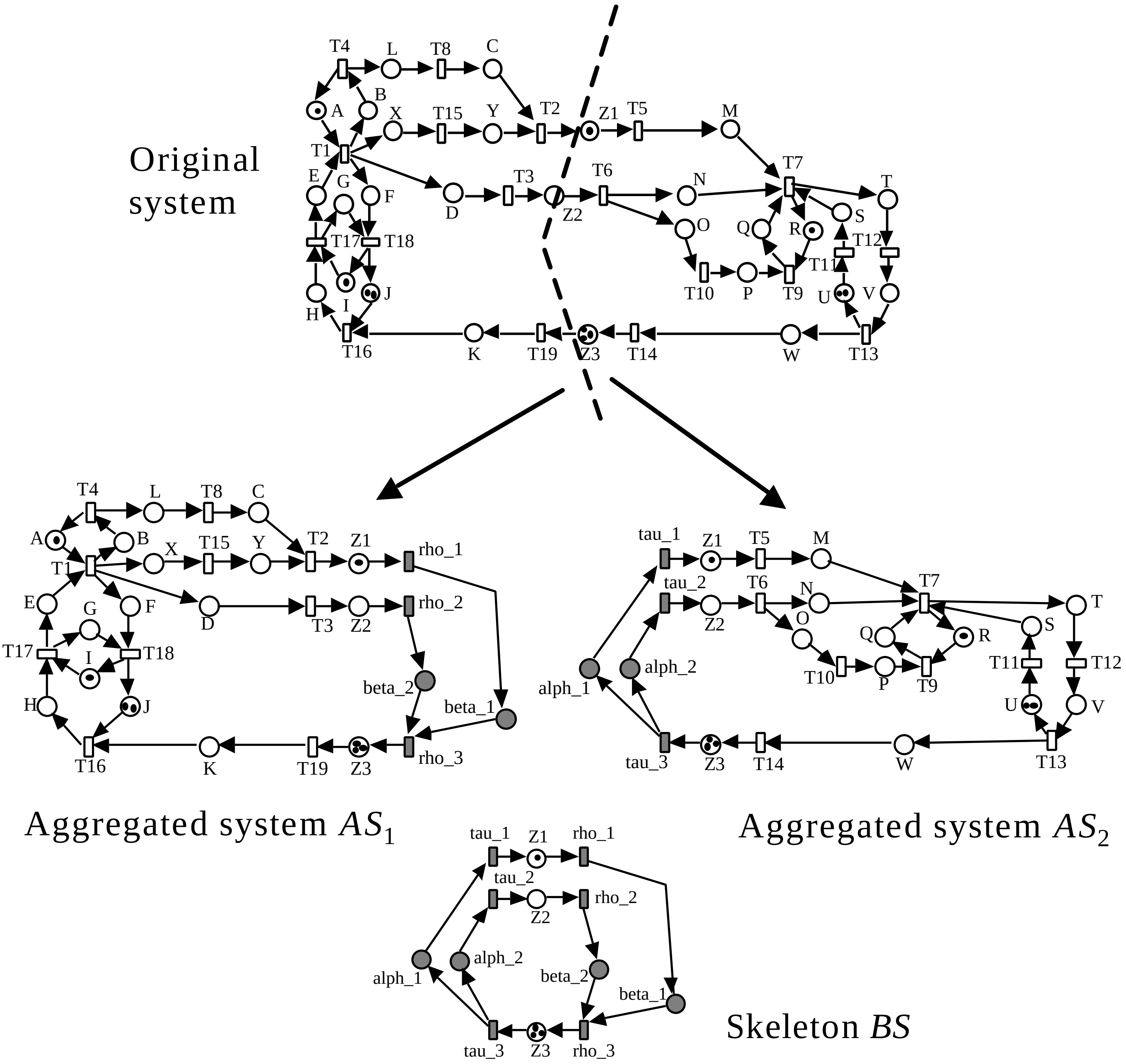


Aggregated subsystems (SMG)

Skeleton (SMG)

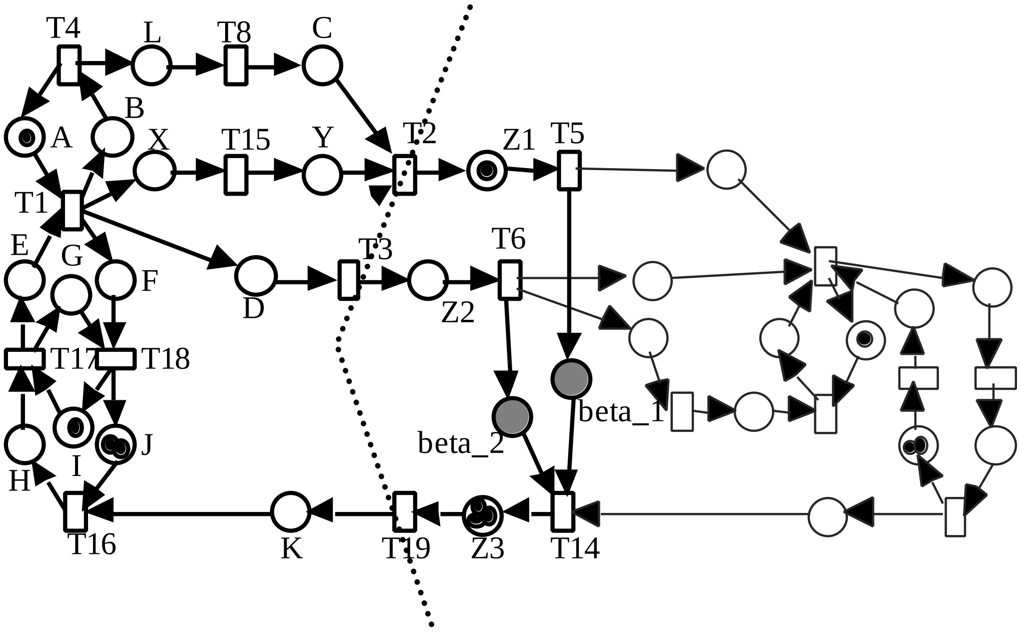
Details of partition: structural decomposition of MG

- Substitute subsystem AS_i by a minimal set IP_i of places



How to compute these places?

Step 1: One place from each input interface transition to each output interface transition



Step 2: Marking is computed using a slight modification of Floyd's *all-pairs shortest paths* algorithm for weighted graphs

transitions --> vertices

places --> arcs

tokens --> weights

Step 3: Delete those added places that are *implicit* in the new system

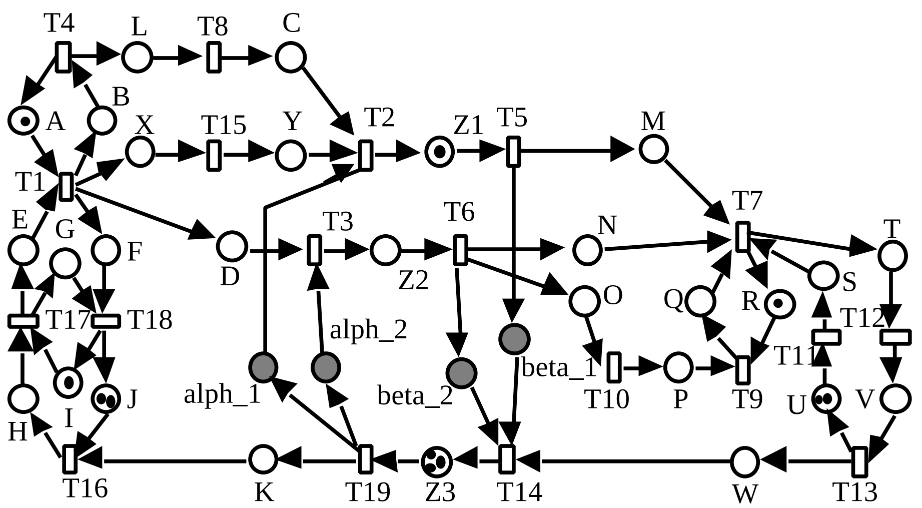
The Decomposition Theorem

Let (N, M_0) be a live and strongly connected MG, $Q1P$ a cut of N and AS_i be the aggregated subsystem obtained from (N, M_0) by substituting all the subnets in N but the ith by the places computed in the previous algorithm.

Then:

(i) The language of firing sequences of the aggregated system is equal to that of the original system projected on the preserved transitions.

(ii) The reachability graph of the aggregated system is isomorphous to that of the original system projected on the preserved places. ♦

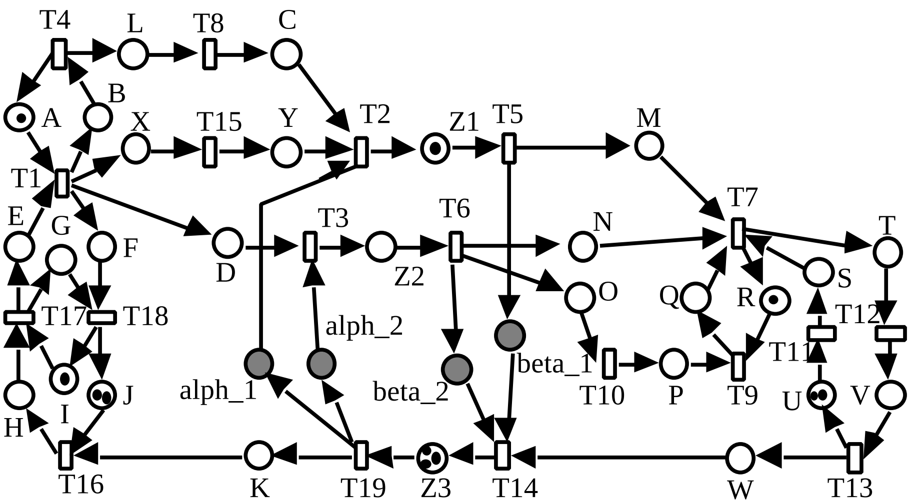


The Decomposition Theorem

- In other words…

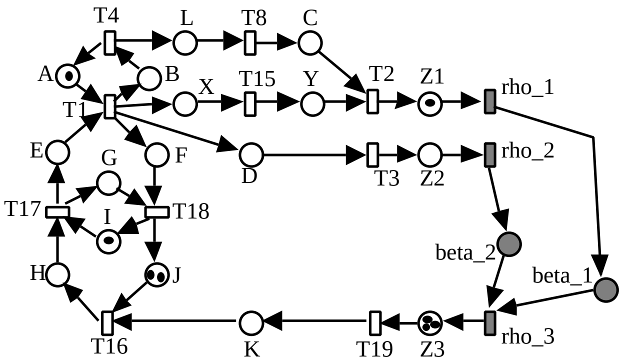
  …the qualitative behaviour of each subsystem is *equivalent* to that of the whole system behaviour projected on the corresponding subset of nodes.

  Equivalence is in terms of language of firing sequences (even more, steps) and reachable markings.

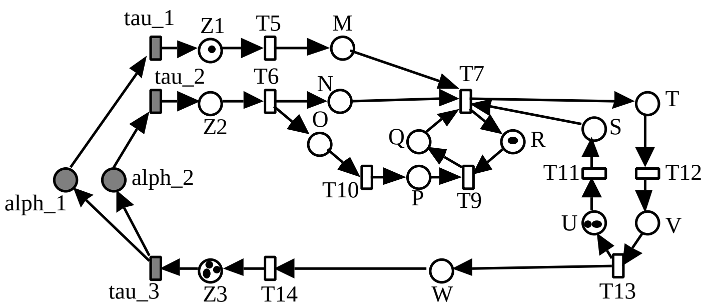


Characterization of subsystems

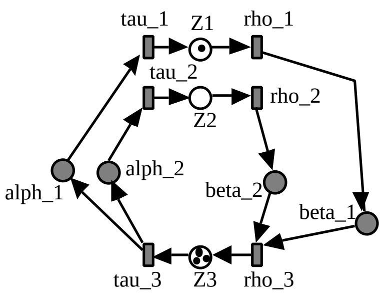
- Definition of unknowns



Service time of *rho_i*



Service time of *tau_j*



Service time of *rho_i* and *tau_j*

Characterization of subsystems

- Additional variables of interest

  – Throughput of:
    - original system
    - first aggregated subsystem
    - second aggregated subsystem
    - skeleton

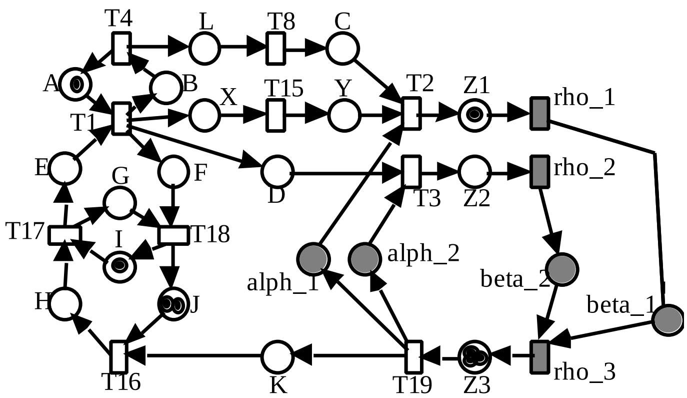
  – Response time of interface transitions at:
    - the original system
    - the first aggregated subsystem
    - the second aggregated subsystem

Estimation of the parameters

❖ Response time approximation:

Response time approximation of the left hand subnet for a token that exits through T2 (Little's law):

$R_2 = M_{alph\_1} / X_2$



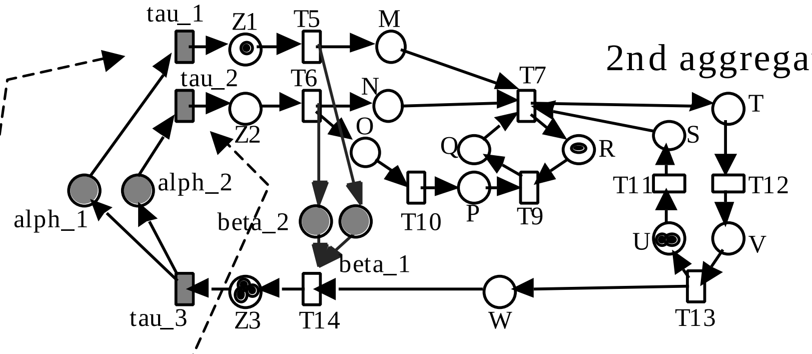
1st aggregated system

and through T3:

$R_3 = M_{alph\_2} / X_3$

where $X_2 = X_3 = X$.

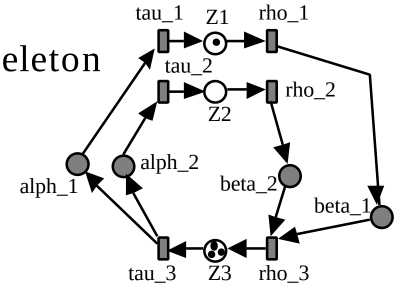
Then $R_2 / R_3 = M_{alph\_1} / M_{alph\_2}$.



2nd aggregated system

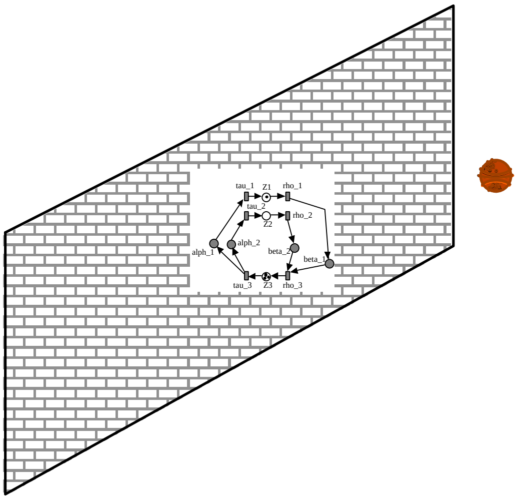
Select tau_1 and tau_2 as:
$tau\_1 = f \cdot R_2$
$tau\_2 = f \cdot R_3$
and compute f such that the throughput of the skeleton is equal to X (linear search of f).

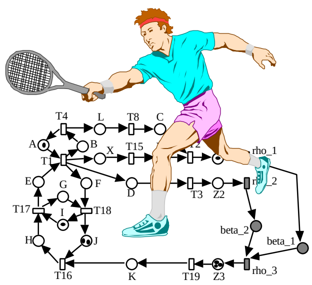
skeleton

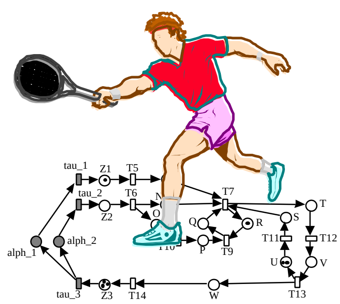


The "pelota" algorithm

select a cut Q;
derive aggregated subsyst. AS_1, AS_2 and skeleton BS;
give initial value $\mu_t^{(0)}$ for each $t \in T_{I2}$;
k:=0; {counter for iteration steps}
repeat
  k:=k+1;
  solve aggregated subsystem AS_1 with
    input: $\mu_t^{(k-1)}$ for each $t \in T_{I2}$,
    output: ratios among $\mu_t^{(k)}$ of $t \in T_{I1}$, and $X_1^{(k)}$;
  solve skeleton BS with
    input: $\mu_t^{(k-1)}$ for each $t \in T_{I2}$,
        ratios among $\mu_t^{(k)}$ of $t \in T_{I1}$, and $X_1^{(k)}$,
    output: scale factor of $\mu_t^{(k)}$ of $t \in T_{I1}$;
  solve aggregated subsystem AS_2 with
    input: $\mu_t^{(k-1)}$ for each $t \in T_{I1}$,
    output: ratios among $\mu_t^{(k)}$ of $t \in T_{I2}$, and $X_2^{(k)}$;
  solve skeleton BS with
    input: $\mu_t^{(k)}$ for each $t \in T_{I1}$,
        ratios among $\mu_t^{(k)}$ of $t \in T_{I2}$, and $X_2^{(k)}$,
    output: scale factor of $\mu_t^{(k)}$ of $t \in T_{I2}$;
until convergence of $X_1^{(k)}$ and $X_2^{(k)}$;







Pelota game

Some results for the example

Service rates (arbitrary):
T2=0.2; T4=0.7; T6=0.3; T8=0.8; T9=0.6; T10=0.5;
Ti=1.0,   i=1,3,5,7,11,12,13,14,15,16,17,18,19

Throughput of the original system: 0.138341
State space of the original system: 89358

Results using the approximation technique:

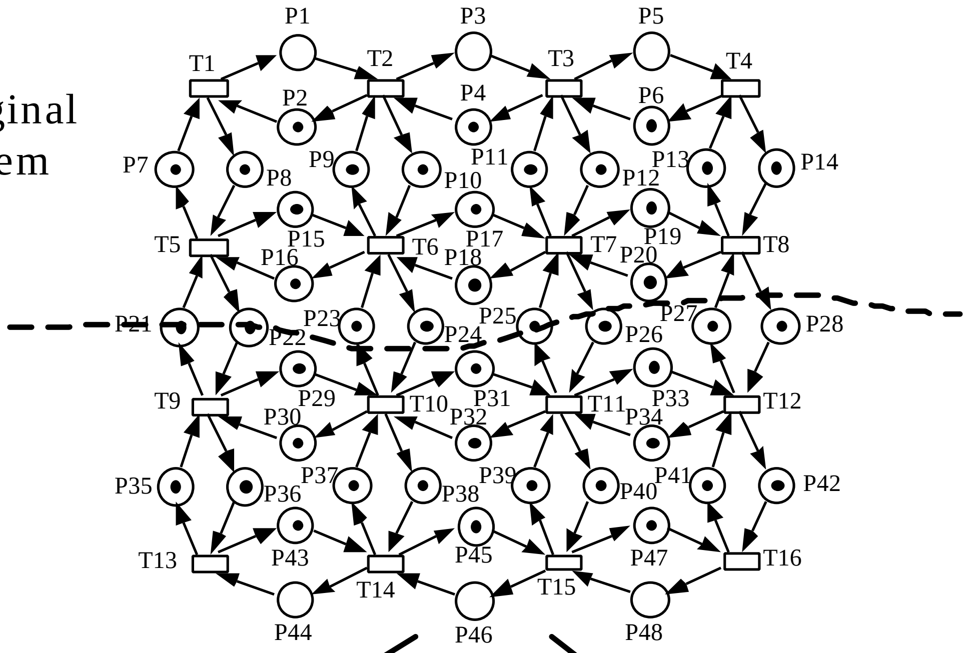
State space AS1: 8288
State space AS2: 3440
State space BS: 231

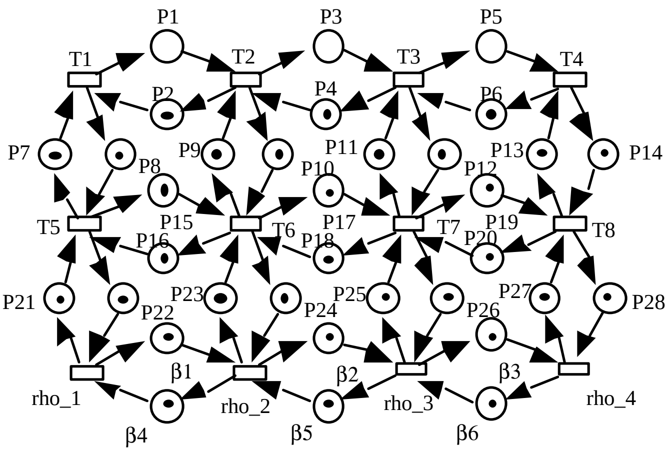
AS1				AS2			
X1	tau_1	tau_2	tau_3	X2	rho_1	rho_2	rho_3
0.17352	0.05170	0.16810	0.88873	0.12714	0.89026	0.21861	0.14354
0.14093	0.06265	0.19707	0.91895	0.13795	0.88267	0.21363	0.13509
0.13856	0.06325	0.19821	0.92054	0.13841	0.88239	0.21343	0.13467
0.13844	0.06328	0.19827	0.92062	0.13843	0.88237	0.21342	0.13465
0.13843	0.06328	0.19827	0.92064	0.13843	0.88238	0.21342	0.13465

Error: -0.064333%

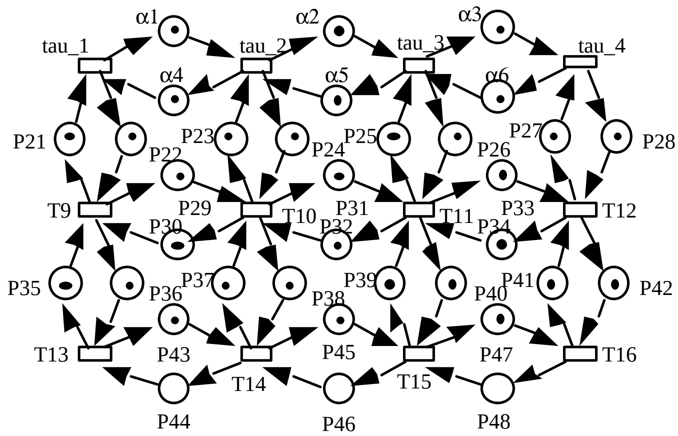
A more complex example

Original system

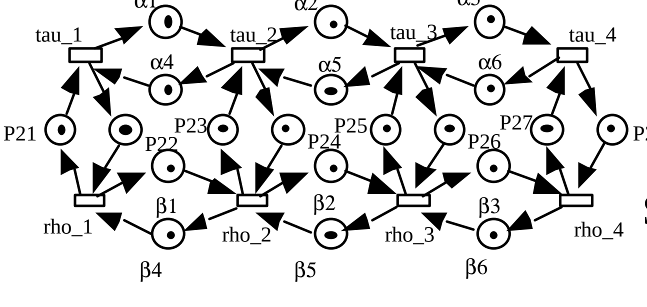




Aggregated system AS_1



Aggregated system AS_2



Skeleton *BS*

A more complex example

Case_1) Firing rates of all transitions (original system) = 1.0

State space original system: 49398
State space AS1 and AS2: 6748 State space BS: 771
Exact Throughput: 0.295945

Initial rates for AS1, 0.1:

AS1					AS2				
X1	tau1	tau2	tau3	tau4	X2	rho1	rho2	rho3	rho4
0.07930	1.02121	1.02452	1.01112	0.80930	0.33294	0.29834	0.50973	0.61599	0.71668
0.29244	0.84574	0.72462	0.55755	0.30802	0.30079	0.29864	0.54035	0.70609	0.83610
0.29710	0.84301	0.71383	0.54364	0.29813	0.29733	0.29758	0.54270	0.71310	0.84299
0.29711	0.84340	0.71354	0.54286	0.29751	0.29711	0.29747	0.54281	0.71352	0.84343

Initial rates for AS1, 1.0:

AS1					AS2				
X1	tau1	tau2	tau3	tau4	X2	rho1	rho2	rho3	rho4
0.33318	0.70982	0.61546	0.51044	0.29917	0.29265	0.30871	0.55771	0.72423	0.84521
0.30095	0.83571	0.70581	0.54034	0.29877	0.29712	0.29817	0.54366	0.71378	0.84293
0.29734	0.84296	0.71307	0.54270	0.29759	0.29712	0.29751	0.54286	0.71354	0.84339
0.29712	0.84343	0.71352	0.54281	0.29747	0.29710	0.29746	0.54282	0.71354	0.84345

Initial rates for AS1, 10.0:

AS1					AS2				
X1	tau1	tau2	tau3	tau4	X2	rho1	rho2	rho3	rho4
0.33419	0.68611	0.59756	0.49474	0.28053	0.28561	0.30812	0.56325	0.73687	0.85741
0.30136	0.83550	0.70455	0.53890	0.29791	0.29679	0.29807	0.54392	0.71447	0.84356
0.29735	0.84299	0.71304	0.54263	0.29753	0.29710	0.29750	0.54287	0.71358	0.84343
0.29711	0.84343	0.71352	0.54281	0.29747	0.29710	0.29746	0.54282	0.71355	0.84346

Error: 0.4% (same for all initial values, fixed point iteration)

A more complex example

Case_2) Firing rates:
T1 = T2 = T3 = T4 = T5 = T6 = T7 = T8 = 1.0;
T9 = T10 = T11 = T12 = T13 = T14 = T15 = T16 = 2.0;
Exact Throughput: 0.333356

Initial rates for AS1: 1.0

AS1					AS2				
X1	tau1	tau2	tau3	tau4	X2	rho1	rho2	rho3	rho4
0.33318	0.70983	0.61546	0.51045	0.29917	0.34424	0.70118	1.49390	1.84123	1.92737
0.33352	0.71500	0.60522	0.49835	0.28554	0.33345	0.68342	1.50320	1.85362	1.93598
0.33345	0.71616	0.60538	0.49834	0.28550	0.33345	0.68281	1.50288	1.85352	1.93592
0.33345	0.71621	0.60539	0.49834	0.28550	0.33345	0.68278	1.50284	1.85348	1.93588

Error: 0.02%

Case_3) Firing rates:
T3 = T4 = T7 = T8 = T11 = T12 = T15 = T16 = 1.0;
T1 = T2 = T5 = T6 = T9 = T10 = T13 = T14 = 2.0;
Exact Throughput: 0.362586

Initial rates for AS1: 1.0

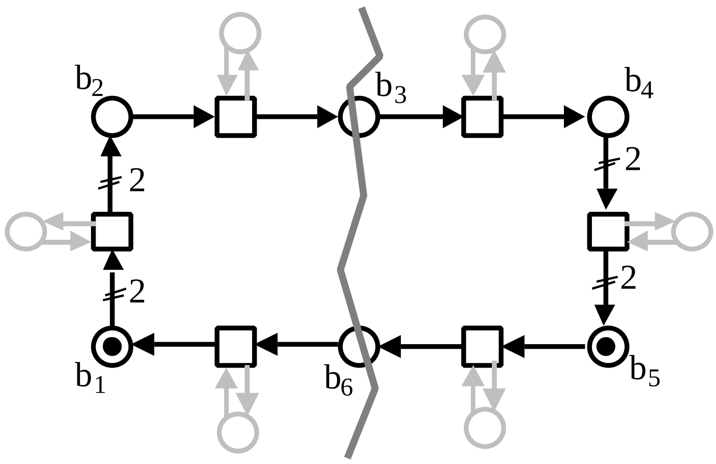
AS1					AS2				
X1	tau1	tau2	tau3	tau4	X2	rho1	rho2	rho3	rho4
0.40526	1.64486	1.58029	0.60759	0.36042	0.35214	0.36948	0.69530	0.61363	0.80667
0.36392	1.81297	1.72253	0.66348	0.38291	0.36239	0.37446	0.68764	0.59809	0.79673
0.36326	1.80988	1.72268	0.66584	0.38570	0.36321	0.37514	0.68748	0.59702	0.79565
0.36328	1.80942	1.72245	0.66596	0.38596	0.36328	0.37520	0.68748	0.59694	0.79556
0.36329	1.80938	1.72243	0.66598	0.38599	0.36329	0.37521	0.68747	0.59693	0.79555

Error: 0.19%
=> Accuracy improves if system is not balanced

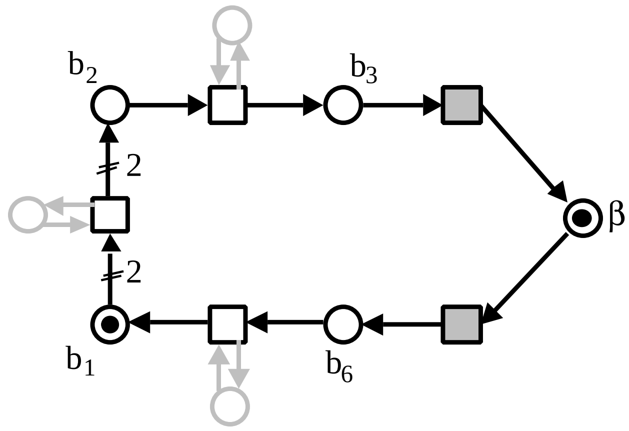
Extension to weighted nets

- In general, the projection of qualitative behaviour cannot be preserved at the aggregated systems

A weighted T-system (MG with weights):



Observe that M(b1)*M(b2) = 0, for all reachable marking M

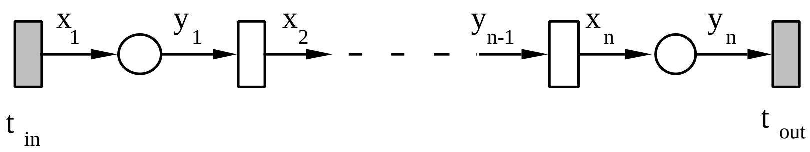


Now, there exists a marking M such that M(b1) = M(b2) = 1

Extension to weighted nets

- Goal: to preserve at least:
  - boundedness (computability) & liveness
  - home state (ergodicity of Markov chain)

- Way: consideration of *gain*, *weighted marking* and *resistance*

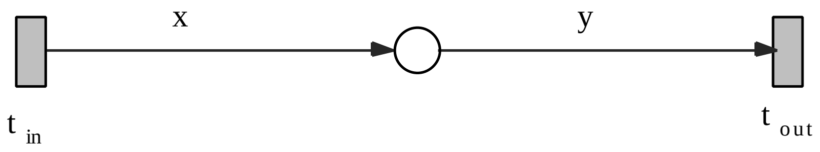


Gain of a weighted path:

$$g(wp) = \prod_{i=1}^{n} \frac{x_i}{y_i}$$

It represents the average number of firings of the final transition per each single firing of the initial one.

To preserve liveness and boundedness it is necessary to preserve the gain.



$g(wp)=x/y$

Extension to weighted nets

And the marking?

Weighted marking of a weighted path:

$$WM(wp) = \sum_{i=1}^{n} \frac{y_1 \cdots y_{i-1}}{x_1 \cdots x_i} \cdot M[P_i]$$

It represents the number of times that the first transition must be fired to achieve the current marking of the path.

Shorting a weighted path we introduce spurious states. To reduce the number of spurious states we introduce the concept of ...

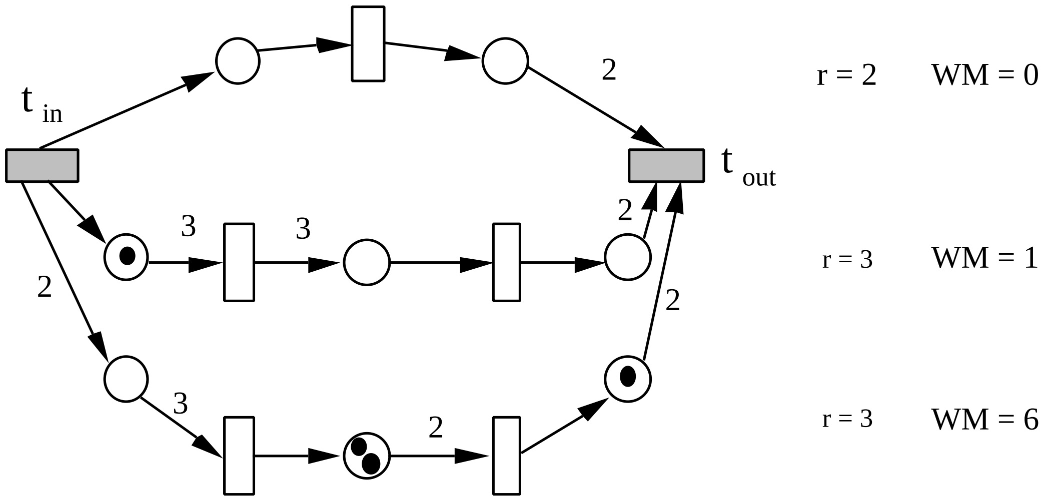
…*resistance* of a weighted path:

$$R(wp) = \max_{i=1,\ldots,n} \prod_{j=1}^{i} \frac{x_j}{y_j} - WM(wp)$$

Resistance is related to the number of firings of the first transition needed to fire the last transition.

Extension to weighted nets

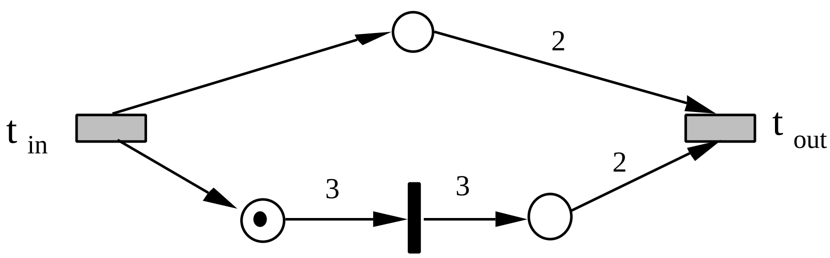
- Between the input & output transitions of an aggregable subsystem



summarize:

– the path with minimum weighted marking

– the path with maximum resistance



This reduction technique preserves the desired properties (liveness, boundedness and existence of home states).

Extension to weighted nets

- Numerical example:

--> here paste a piece of paper
(with pictures and numbers)

Extension to weighted nets

--> here paste a piece of paper
(with pictures and numbers)

Extension to weighted nets

--> here paste a piece of paper
(with pictures and numbers)

Extension to nets with choices

--> here paste a piece of paper
(with pictures and numbers)

Extension to nets with choices

--> here paste a piece of paper
(with pictures and numbers)

Extension to nets with choices

--> here paste a piece of paper
(with pictures and numbers)

Final comments and forthcoming research efforts

- Classical trade-off accuracy / complexity
  - accuracy decreases with the number of subsystems (one subsystem ==> exact!)
  - complexity decreases with the number of subsystems

- The problem of the bad quality of temporal abstraction

  (in comparison to qualitative or logical abstraction)

- The need of a hierarchical approach with the possibility of using different techniques at each abstraction level