EE249 Discussion: Petri Net

David Burnett
Wei Yang Tan

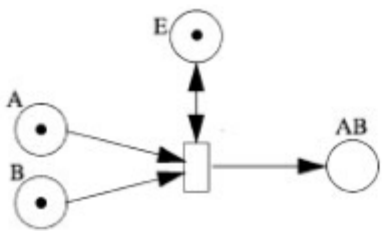
Outline

- Basics of Petri Net
- Applications of Petri Net
  - Supply Chain Modeling
  - Biological Network
- Petri Net Synthesis

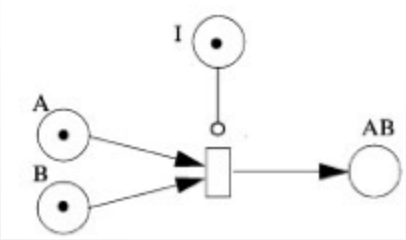
Basics of Petri Net

Petri Net Recap

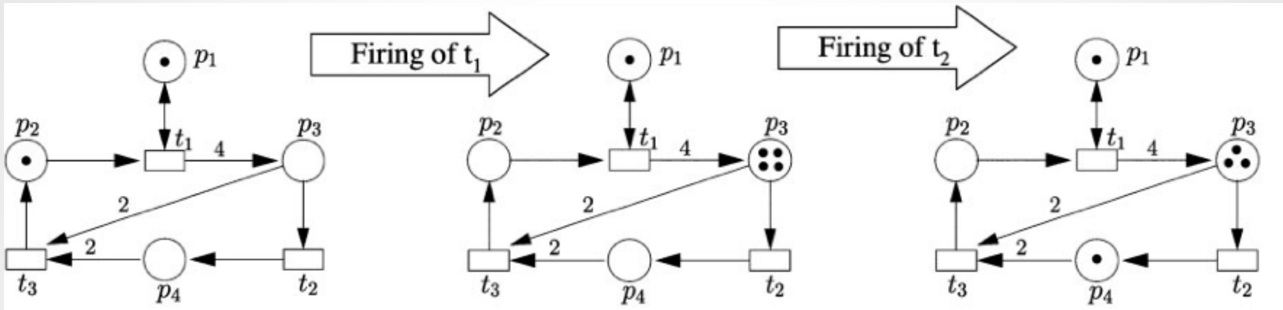
- A directed-bipartite graph
- Two kinds of nodes:
  a. places (-> resources)
  b. transitions (-> events)
- Test arc



- Inhibitor arc



Analysis of Petri Net



pre - condition matrix

Pre =	t_1	t_2	t_3
p_1	1	0	0
p_2	1	0	0
p_3	0	1	2
p_4	0	0	2

post - condition matrix

Post =	t_1	t_2	t_3
p_1	1	0	0
p_2	0	0	1
p_3	4	0	0
p_4	0	1	0

initial marking

$$M_0 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

state equation

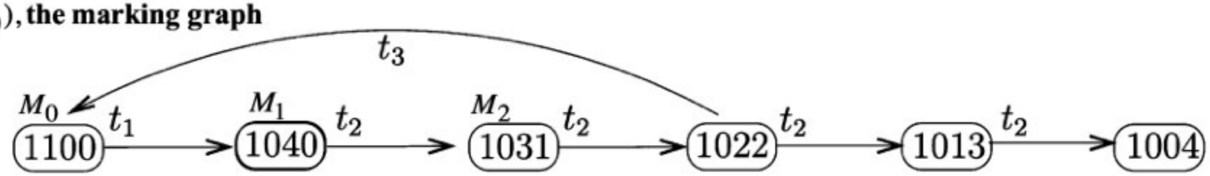
$$M' = M + C.\sigma,$$

incidence matrix

C = Post − Pre =	t_1	t_2	t_3
p_1	0	0	0
p_2	−1	0	1
p_3	4	−1	−2
p_4	0	1	−2

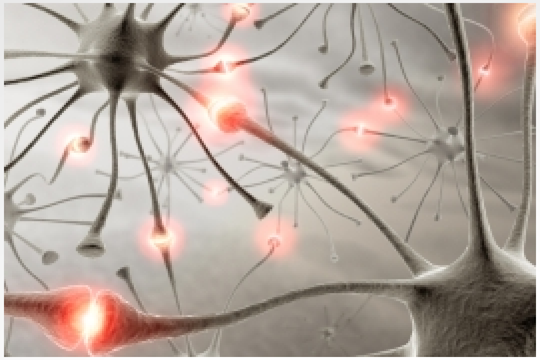
$$M_2 = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 1 \end{bmatrix} = M_0 + C.\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

$R(M_0)$, **the marking graph**



Petri Net Recap

- Boundedness: # of tokens is always bounded
- S/P-invariant: set of places which total token count remains constant
- T-invariant: sequence of transitions that bring back to original marking
- Reachability: able to evolve from initial marking to targeted marking
- Liveness: always able to fire any transition

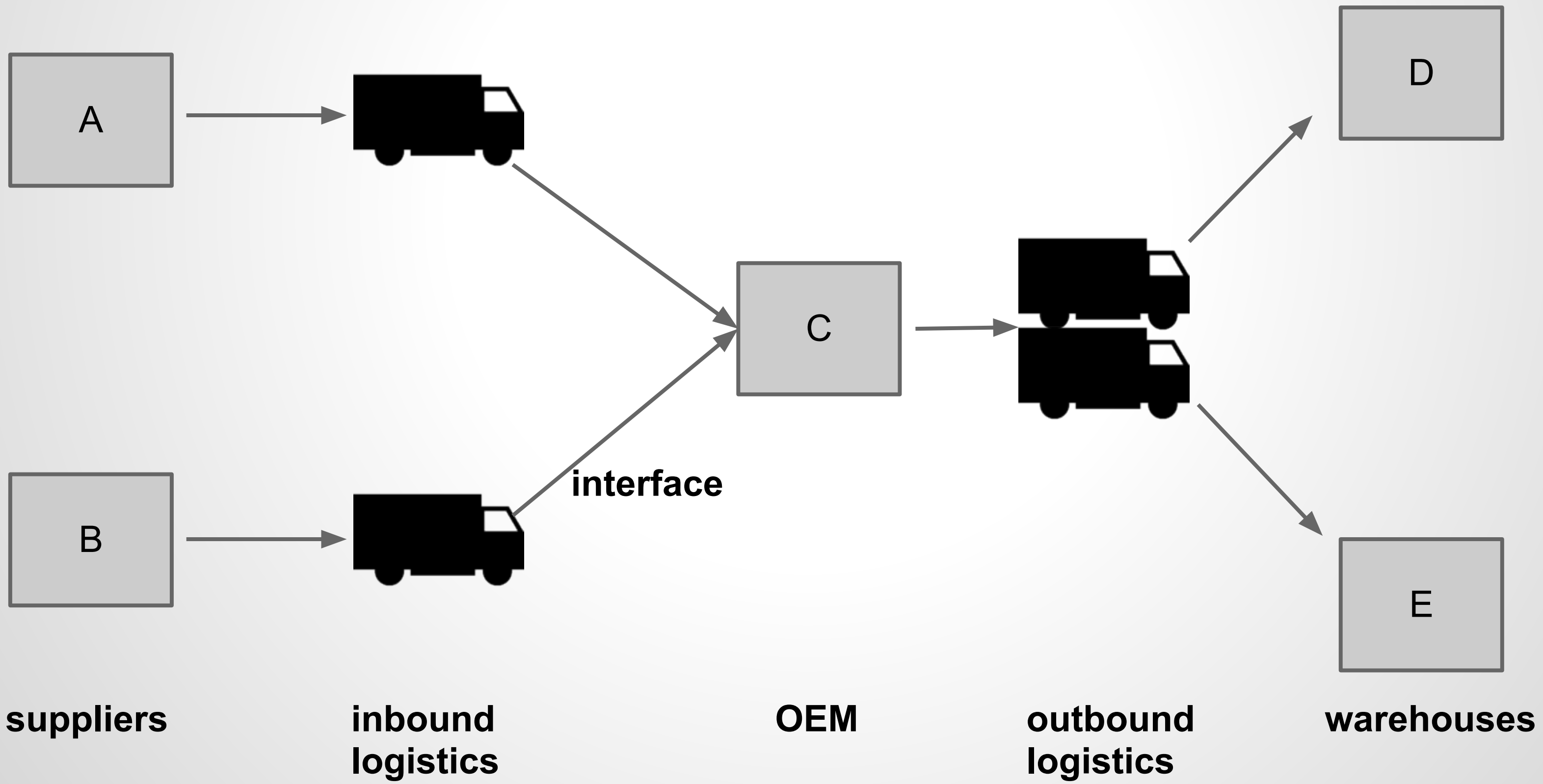


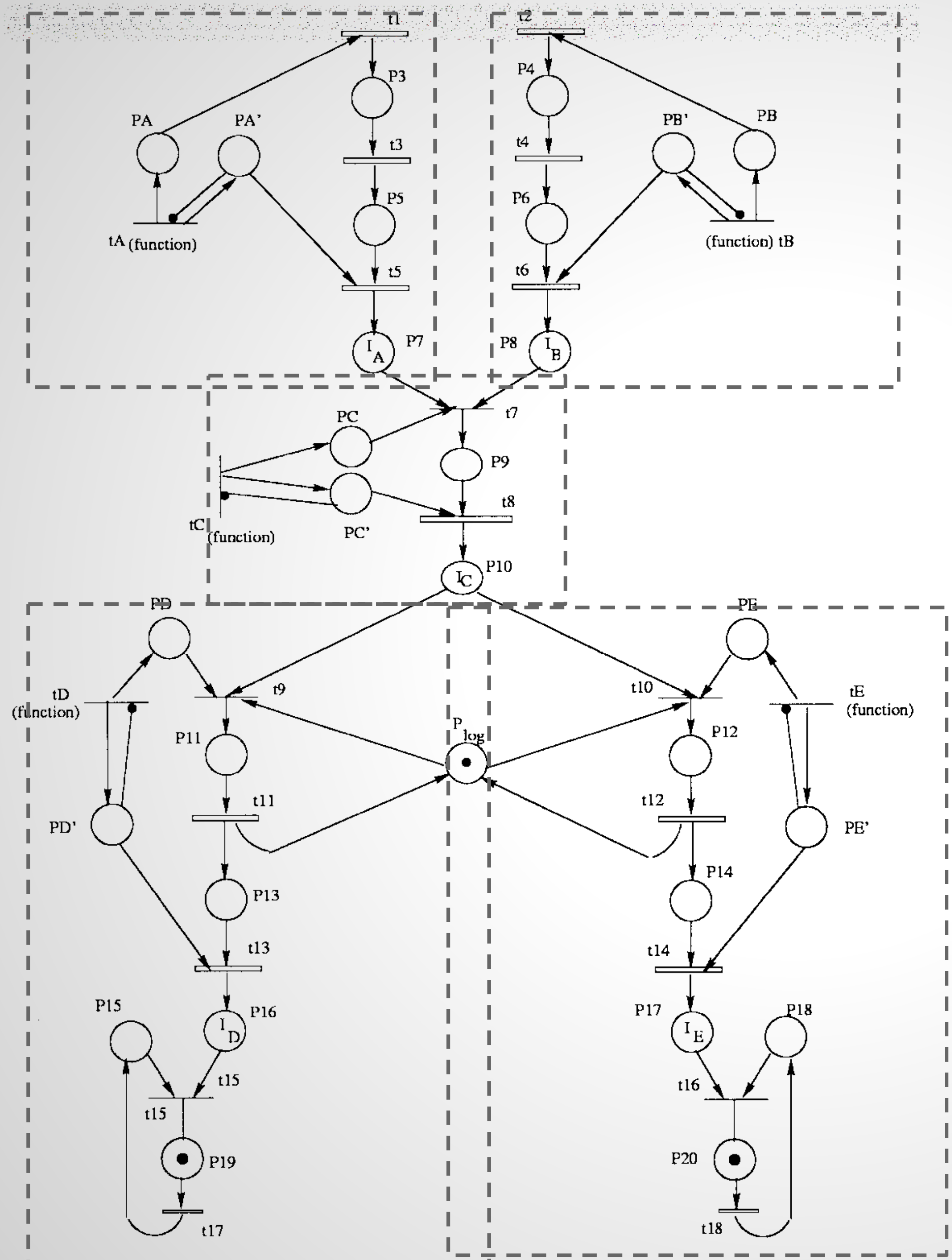
Petri Net Applications

Petri Net on Supply Chains

- Used generalised stochastic Petri nets (GSPN) models for analysis
- Solve two cost minimisation problems:
  a. comparison of make-to-stock (MTS) and assemble-to-order (ATO) policies
  b. locating decoupling point in supply chain
- Software used: stochastic petri net package (SPNP)

Supply Chain





Place name	*Description*
PA′	Material on order to supplier of A
PB′	Material on order to supplier of B
PA	Manufacturing at supplier of A
PB	Manufacturing at supplier of B
P3	Logistics from supplier of A
P4	Logistics from supplier of B
P5	Interface between supplier A logistics and OEM
P6	Interface between supplier B logistics and OEM
P7	Available inventory of A
P8	Available inventory of B
PC	Order receipt for production of C
PC′	Material on order for production of C
P9	Production of C
P10	Inventory of C available
PD	Order receipt for production of D
PD′	Material on order for production of D
PE	Order receipt for production of E
PE′	Material on order for production of E
P11	Outbound logistics of D from plant to warehouse
P12	Outbound logistics of E from plant to warehouse
Plog	Logistics carriers available
P13	Assembling of D from inventory of C
P14	Assembling of E from inventory of C
P16	Finished goods inventory of D at warehouse
P17	Finished goods inventory of E at warehouse
P15	Back order for D ready
P18	Back order for E ready
P19	Customer order for D ready
P20	Customer order for E ready

Transition name	*Description*
tA	Start of manufacturing of A
tB	Start of manufacturing of B
t1	Processing by supplier of A
t2	Processing by supplier of B
t3	Transportation from supplier of A
t4	Transportation from supplier of B
t5	Paper work or interfaces with supplier of A
t6	Paper work or interfaces with supplier of B
tC	Trigger for production of C
t7	Manufacturer of C starts production
t8	Processing of C
tD	Trigger for assembling of D
tE	Trigger for assembling of E
t9	End of assembling of D from C
t10	End of assembling of E from C
t11	Outbound logistics of D
t12	Outbound logistics of E
t13	Assembling of D
t14	Assembling of E
t15	Customer order for D served
t16	Customer order for E served
t17	Arrival of order for D
t18	Arrival of order for E

MTS vs ATO: Setup

- H_I: incurred holding cost for inventories of first stage (i.e. *A* and *B*)
- H_D: cost per hour of delayed delivery

MTS vs ATO: Setup

- Assumptions:
  - H_D of E = 150% H_D of D
  - H_I of E or D = 120% H_I of C;
    H_I of C = 120% H_I of A or B
- I_A, I_B, I_C = 6 units; I_D, I_E = 3 units (for MTS only)
- Inventory level that triggers reorder:
  - A, B = 1 unit; C = 10 units; D, E = 1 unit (for MTS only)

MTS vs ATO: Setup

- Transition firing rate:

Transition name	*Firing rate (units/h)*
t1	1.00
t2	1.00
t3	3.00
t4	2.00
t5	6.00
t6	4.00
t8	2.00
t11	4.00
t12	2.00
t13	4.00
t14	3.00
t17	0.80
t18	0.60

MTS vs ATO: Effect of Arrival Rates

- Variation of total cost with arrival rates of D

λ_D units/h	Total cost			
	$H_D/H_I = 1.5$		$H_D/H_I = 40.0$	
	MTS system	ATO system	MTS system	ATO system
0.8	22.421	19.815	26.001	257.437
1.0	21.237	18.610	25.818	237.559
1.2	20.012	17.714	25.961	224.228
1.4	18.774	17.016	26.339	214.675

MTS vs ATO: Effect of Inventory *C*

- FGI: finished goods inventory

FGI_C	Total cost: $H_D/H_I = 1.5$	Total cost: $H_D/H_I = 40.0$
6	18.54	28.01
9	27.53	29.34
12	35.553	42.175
15	43.403	49.929

134% 78%

MTS

FGI_C	Total cost: $H_D/H_I = 1.5$	Total cost: $H_D/H_I = 40.0$
5	15.64	197.40
6	18.37	201.52
7	21.07	204.87
8	23.73	207.92

ATO

MTS vs ATO: Effect of Interface Times of B

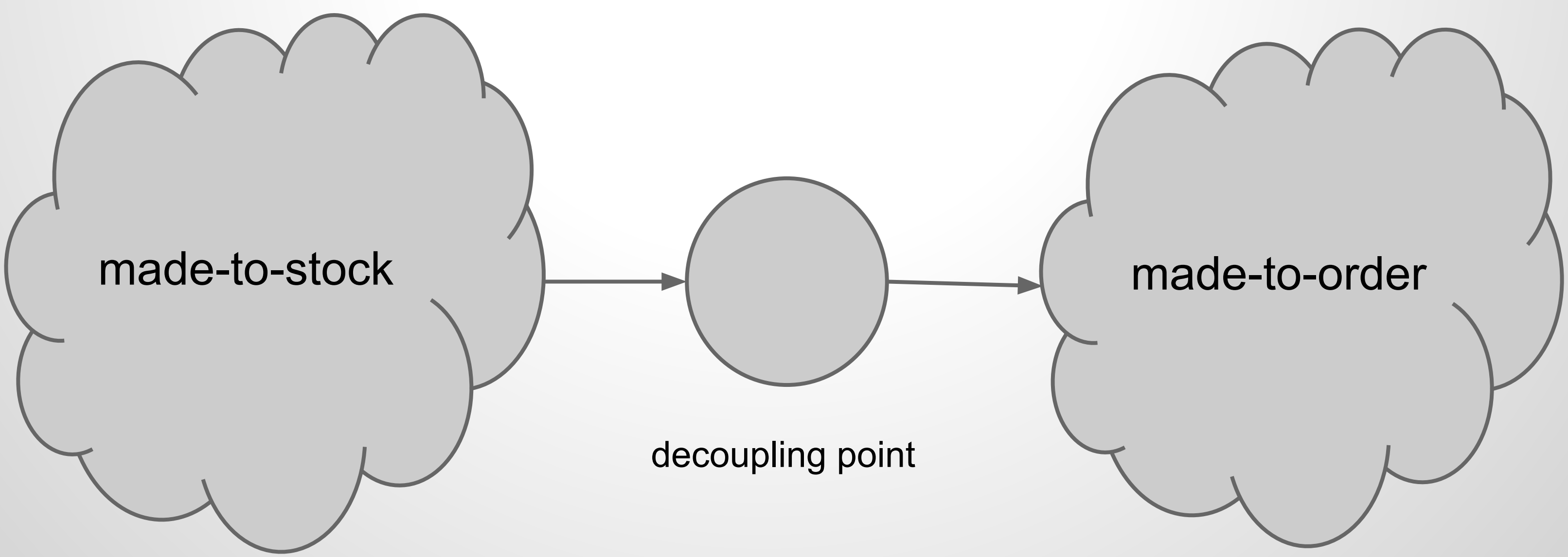
- Total costs increase marginally for both policies as interface times decreases
- Inventory held at C increases, as adds more inventory at greater pace -> more holding costs
- Conclusion: need to consider the whole supply chain (for e.g., need to reduce interface times at A too to derive benefits)

Table 7 Variation of total cost with interface rates of supplier of B

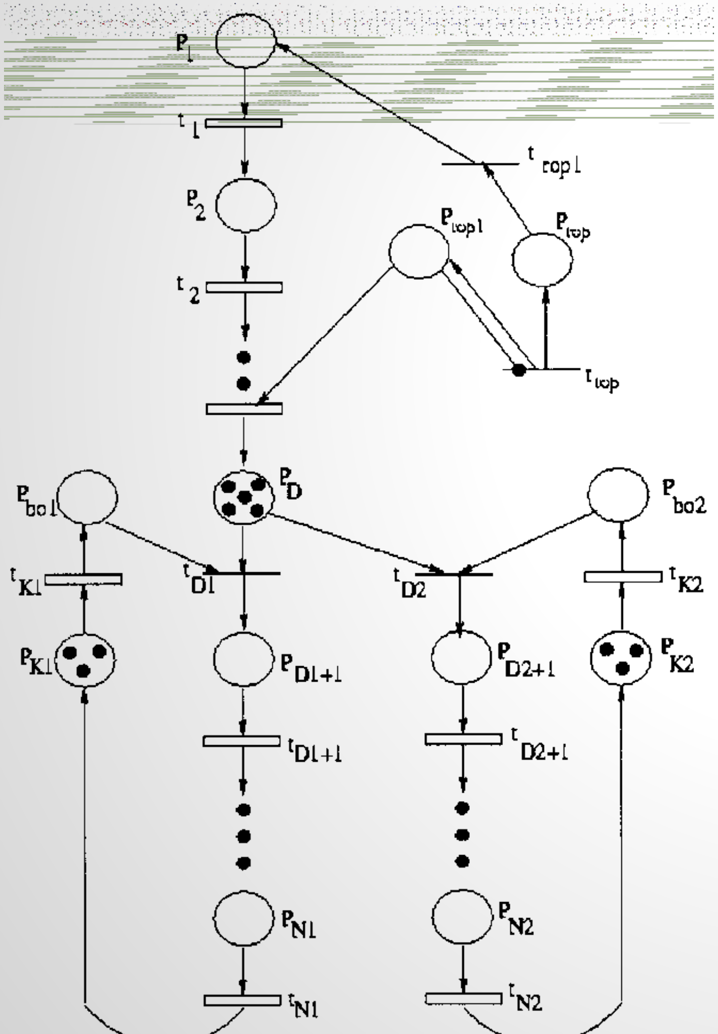
Interface rates with S_2 units/h	*Total cost* $H_D/H_I = 1.5$		$H_D/H_I = 40.0$	
	MTS system	*ATO system*	*MTS system*	*ATO system*
4.0	22.566	15.542	24.934	197.185
5.0	22.651	15.640	24.981	197.360
6.0	22.709	15.705	25.038	197.502
8.0	22.780	15.785	25.109	197.659

Decoupling Point Location Problem (DPLP)

- Decoupling point (DP): point in space where order is assigned to the customer
- Aim: to find minimal cost DP
    a. cost of holding inventory before DP
    b. cost of excess lead time from DP onwards



Decoupling Point Location Problem (DPLP)



after aggregation

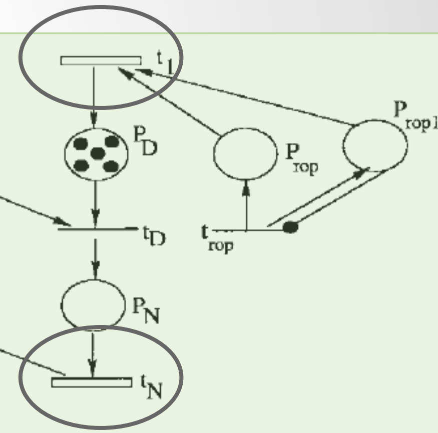


Figure 7 The aggregated GSPN model of the supply chain considered for DPLP.

DPLP Setup

- Use an integrated queuing-GSPN model
- Holding costs increase by 20% each stage as move away to distribution centre

Table 10 Input parameters for the DPLP considered

Facility	*Average service rate (jobs/h)* μ	*SCV of service time*
Supplier 1	10.0	0.7
Supplier 2	15.0	0.7
Initial assembly plant	25.0	0.8
Final assembly plant	10.0	0.8
Distribution	20.0	0.6

DPLP Analysis

Table 12 The total costs for general service times and base stock policy

Decoupling point	Total cost			
	$H_D/H_I = 10$	$H_D/H_I = 30$	$H_D/H_I = 40$	$H_D/H_I = 50$
1	3.596	5.940	8.285	10.630
2	5.214	6.963	8.712	10.461
3	8.967	10.237	11.508	12.779
4	12.071	12.429	12.788	13.147

Table 13 The total costs for general service times and reorder point policy

Decoupling point	Total cost			
	$H_D/H_I = 10$	$H_D/H_I = 20$	$H_D/H_I = 30$	$H_D/H_I = 40$
1	3.427	5.853	8.280	10.706
2	4.234	6.060	7.886	9.711
3	5.686	6.974	8.262	9.550
4	8.055	8.414	8.772	9.131

- DP moves to the right (i.e. toward the retail outlets) as H_D/H_I increases

Petri nets for Biological Networks

Overview

- Summary paper to raise awareness
- "Petri Net Basics" section summarizes class
- Table 1, available software tools
- 50+ references to specific BioPN papers
- Outlines a few ways PN structures can describe biological processes

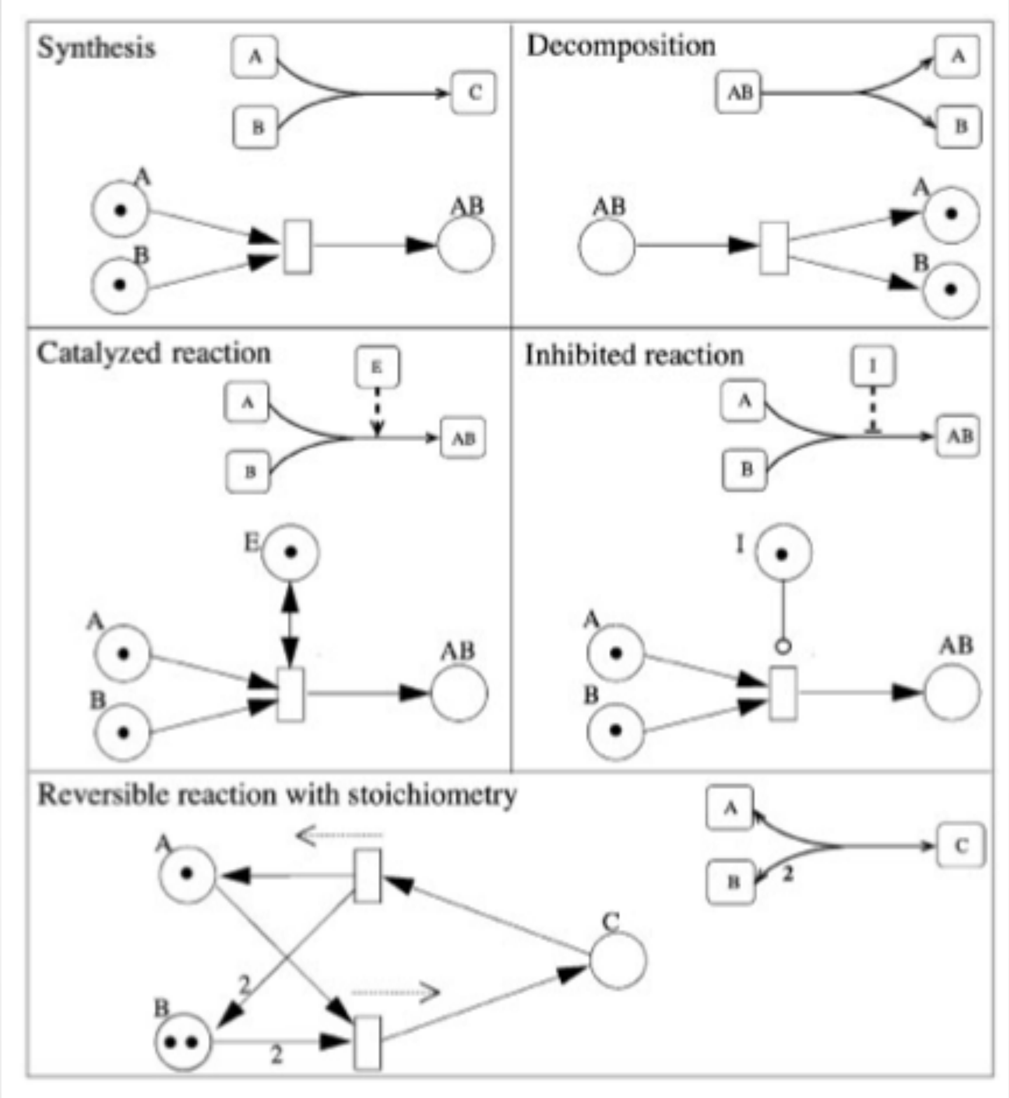


Figure 2, PN structures for biological reactions
Note: complex formation, binding process, transcriptional regulation, etc.

Applicability

- PNs a significant tool for practical biological abstraction
- Wide selection in software analysis tools
- PNs well suited to the qualitative types of data available in biological sciences
  - Relation, not concentration
- Boundedness: in biology, means no product can accumulate
- T-invariants: could represent a cyclical metabolic process

New PN types

- Stochastic Petri Nets (SPNs)
  - Random molecular interaction rates
- Hybrid Petri Nets (HPNs)
  - Allowing continuous and discrete time
- Functional Petri Nets (FPNs)
  - Including state-dependent control logic
- Hybrid Functional Petri Nets (HFPNs)
  - Introduced specifically for biological modeling
- Colored Petri Nets (CPNs)
  - Multiple types of tokens

Most biological models can be unified into some type of Petri Net

Sample applications

- Genetic networks
  - Standard PNs possible, but CPNs enable a compact model and automated analysis
  - HPNs and HFPNs also considered, but data to create kinetic models is lacking
- Signaling networks
  - HFPNs, standard PNs, timed PNs, timed CPNs all considered to model certain aspects
- Many viable options exist to model given processes

Petri Net Synthesis/Scheduling

Intro

- Paper focuses on the scheduling problem
  - Only necessary for concurrent use of shared, limited resources
- Three steps to scheduling/synthesis:
  - Static, with truly fixed dependencies
  - Quasi-static, with periodic dependencies
  - Dynamic, with dependencies decided at runtime
- If the dynamic tasks are schedulable, the algorithm decomposes it into many quasi-static states

PN review section

- Yet another summary of Petri Nets
- Much more technical overview
- Paper focus is on Free Choice Petri Nets
  - Each transition has only one predecessor

Schedulability?

- Only schedulable if rules aren't broken
  - Every transition is fired
  - No unbounded accumulation of tokens
  - Playing against adversary: anti-game theory
- Finding a schedule
  - Decompose into Conflict Free components (s. 77)
    - T-allocations such that only one transition is fired
    - T-reductions eliminate inactive net portion
  - Check if every component is statically schedulable
    - When looping back to a given system state/marking, ensure no deadlock
  - Schedule a loop that returns to the initial marking

Application

- Generate C-code from schedule
  - Insert code for blocks: ABBCC, ABCBC, etc.
  - Lines of C scales linearly with PN size
    - Generating schedule is exponential/polynomial
  - if-else: traverses both possibilities
  - multirate nets: counter determines how to proceed
- Over partitioning tasks by hand:
  - 23% savings in lines of code
  - 21% savings in clock cycles
  - 1000% savings in designer time

Conclusions

- Items (1) through (4) in last paragraph
  - Maximizes work done at compile time
  - Free-Choice PNs are decidable
  - Schedulability is verified, not assumed
  - Functions with same executions rates are combined

Thank you!