On Modeling the Tactical Planning of Oil Pipeline Networks

Daniel Felix Ferber

PETROBRAS

ICAPS 2012

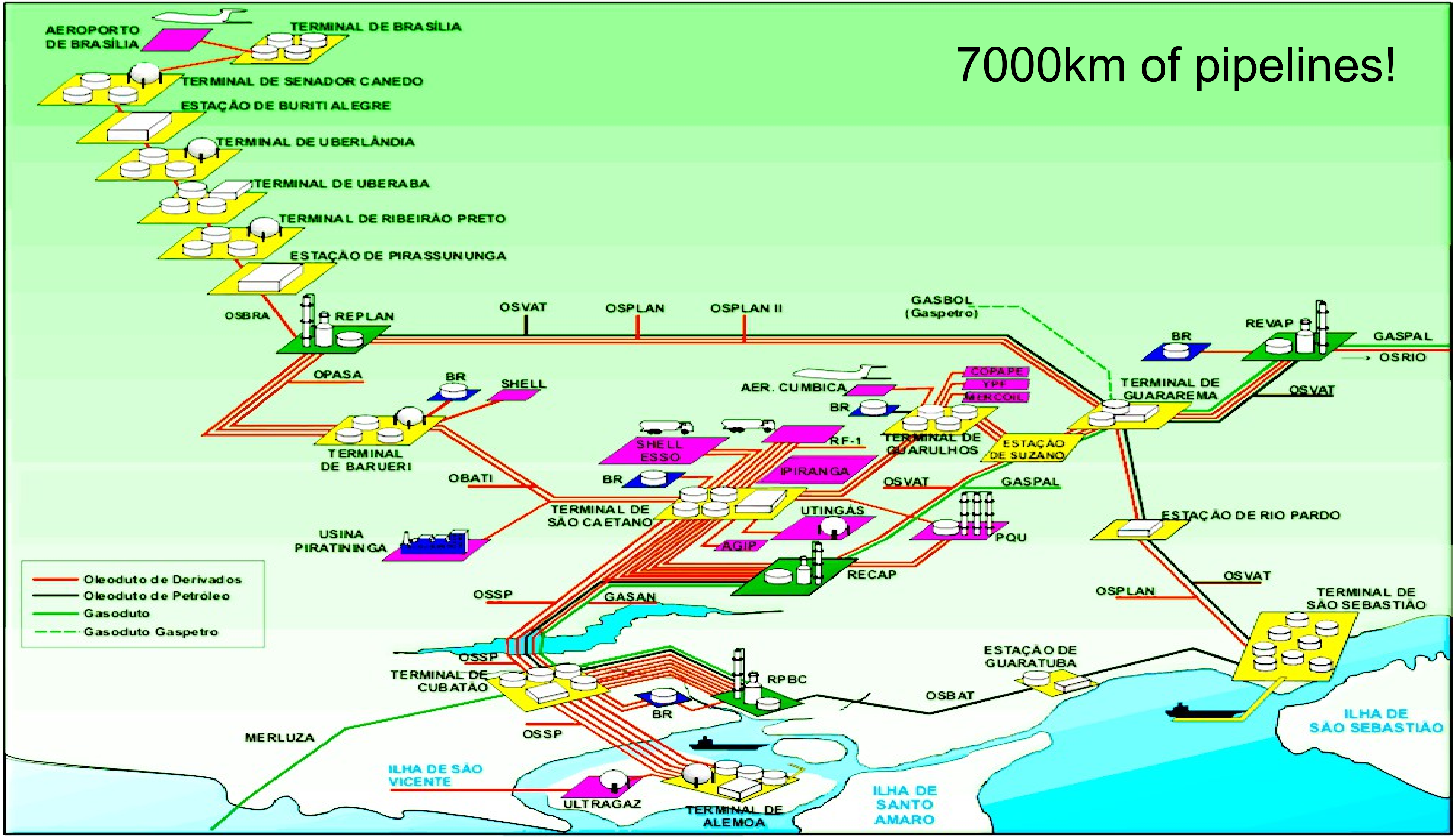
Introduction

The **supply chain** at Petrobras:

- Pipeline Networks
- Oil refined commodities

- Multi-commodity
- Multi-period

Motivation


7000km of pipelines!
AEROPORTO DE BRASÍLIA
TERMINAL DE BRASÍLIA
TERMINAL DE SENADOR CANEDO
ESTAÇÃO DE BURITI ALEGRE
TERMINAL DE UBERLÂNDIA
TERMINAL DE UBERABA
TERMINAL DE RIBEIRÃO PRETO
ESTAÇÃO DE PIRASSUNUNGA
OSBRA
REPLAN
OSVAT
OSPLAN
OSPLAN II
GASBOL (Gaspetro)
REVAP
GASPAL
OSRIO
TERMINAL DE GUARAREMA
OPASA
SHELL
AER. CUMBICA
COPAPE
YPF
MERCOIL
TERMINAL DE BARUERI
SHELL ESSO
RF-1
TERMINAL DE GUARULHOS
ESTAÇÃO DE SUZANO
IPIRANGA
OBATI
TERMINAL DE SÃO CAETANO
UTINGÁS
PQU
ESTAÇÃO DE RIO PARDO
USINA PIRATININGA
AGIP
RECAP
Oleoduto de Derivados
Oleoduto de Petróleo
Gasoduto
Gasoduto Gaspetro
OSSP
GASAN
TERMINAL DE SÃO SEBASTIÃO
ESTAÇÃO DE GUARATUBA
TERMINAL DE CUBATÃO
RPBC
OSBAT
MERLUZA
ILHA DE SÃO SEBASTIÃO
ILHA DE SÃO VICENTE
ULTRAGAZ
TERMINAL DE ALEMOA
ILHA DE SANTO AMARO

Motivation

Our main goal:

- Assure minimal inventory levels at consumer facilities.

Decisions:

- Amount
- Timeframe
- Path
- Flow rate

The pipeline network plan:

- A description of flow among nodes.
- Ignores operational details: not yet a schedule.

Motivation

Current solution:

- Classic network flow model.
- Solution requires many "fixes":

  Inventory on pipelines, average flow capacity, etc.

Not a realistic flow description!

Motivation

Some desired **aspects**:

- Inventory of pipelines (*in-transit inventory*)
- Transit time
- Flow capacity
- Flow reversal

Incorporate scheduling aspects into the plan!

Motivation

A **linear programming** approach:

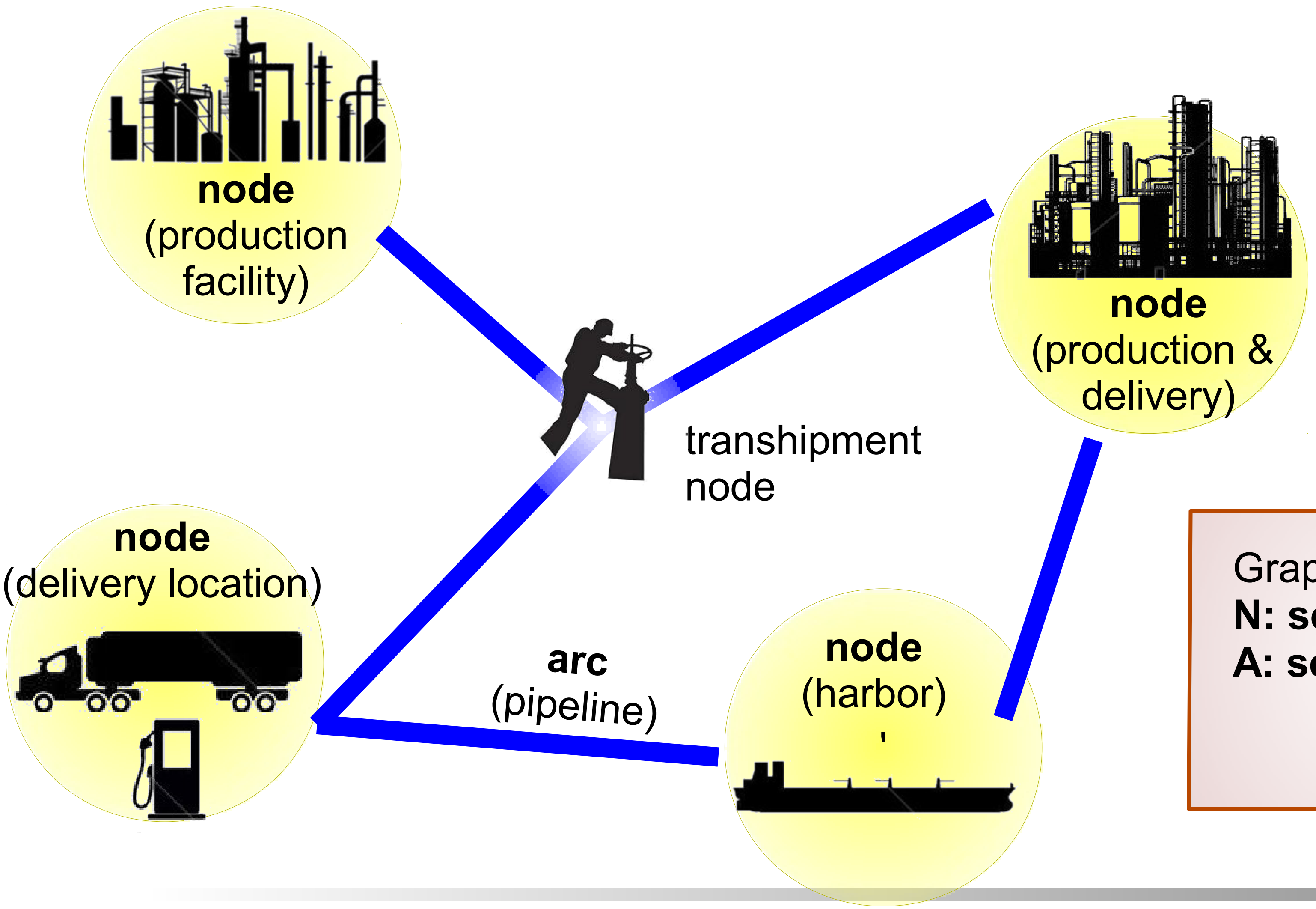
Well-known and proven solution

- **Challenge: NO integer variables!**
  - Fast execution
  - Large topologies

Suited for tactical planning.

The Pipeline Operation

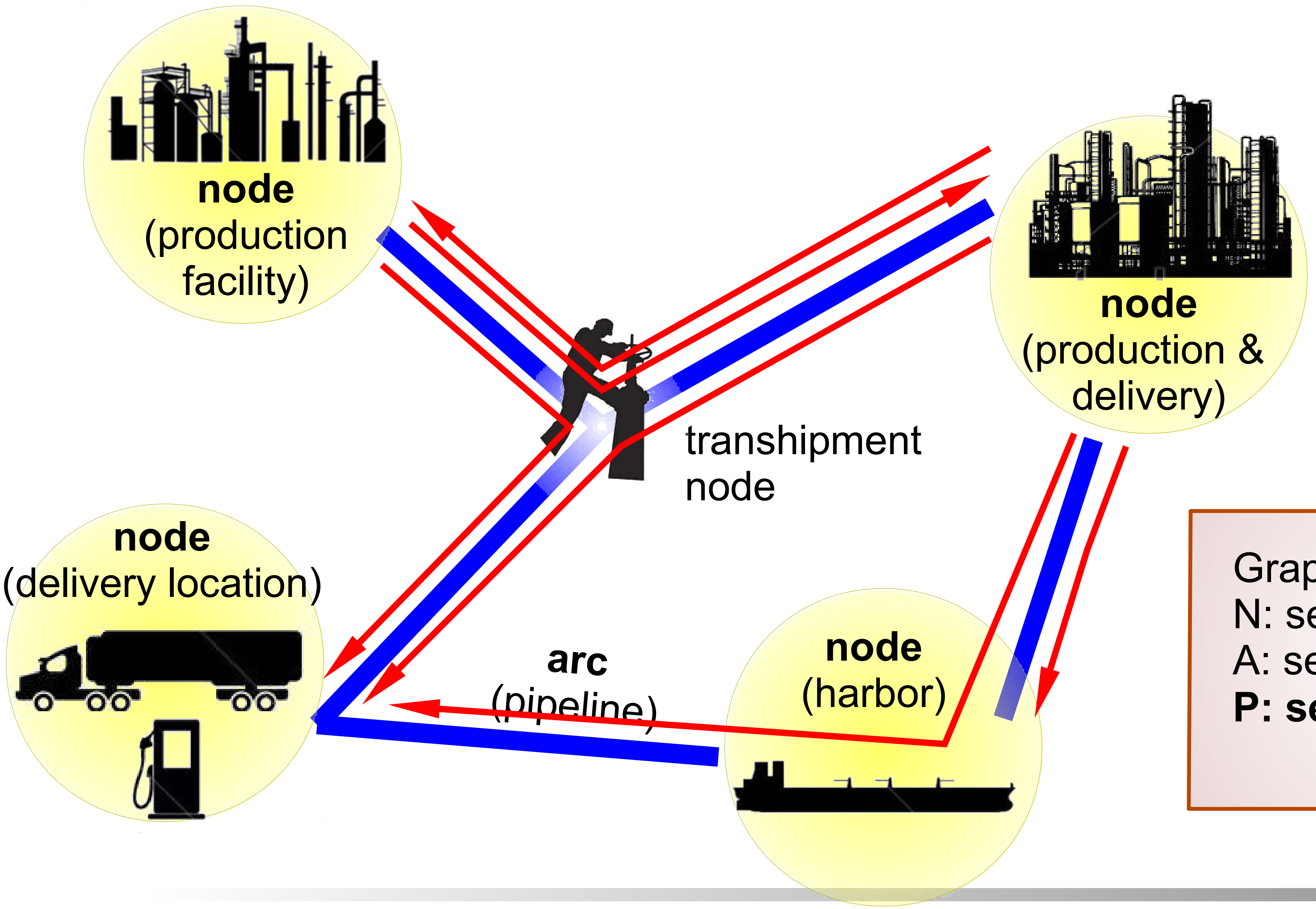
Pipeline network: a graph of ‘***arcs***’ and ‘***nodes***’



Graph G(N, A)
N: set of nodes
A: set of arcs

The Pipeline Operation

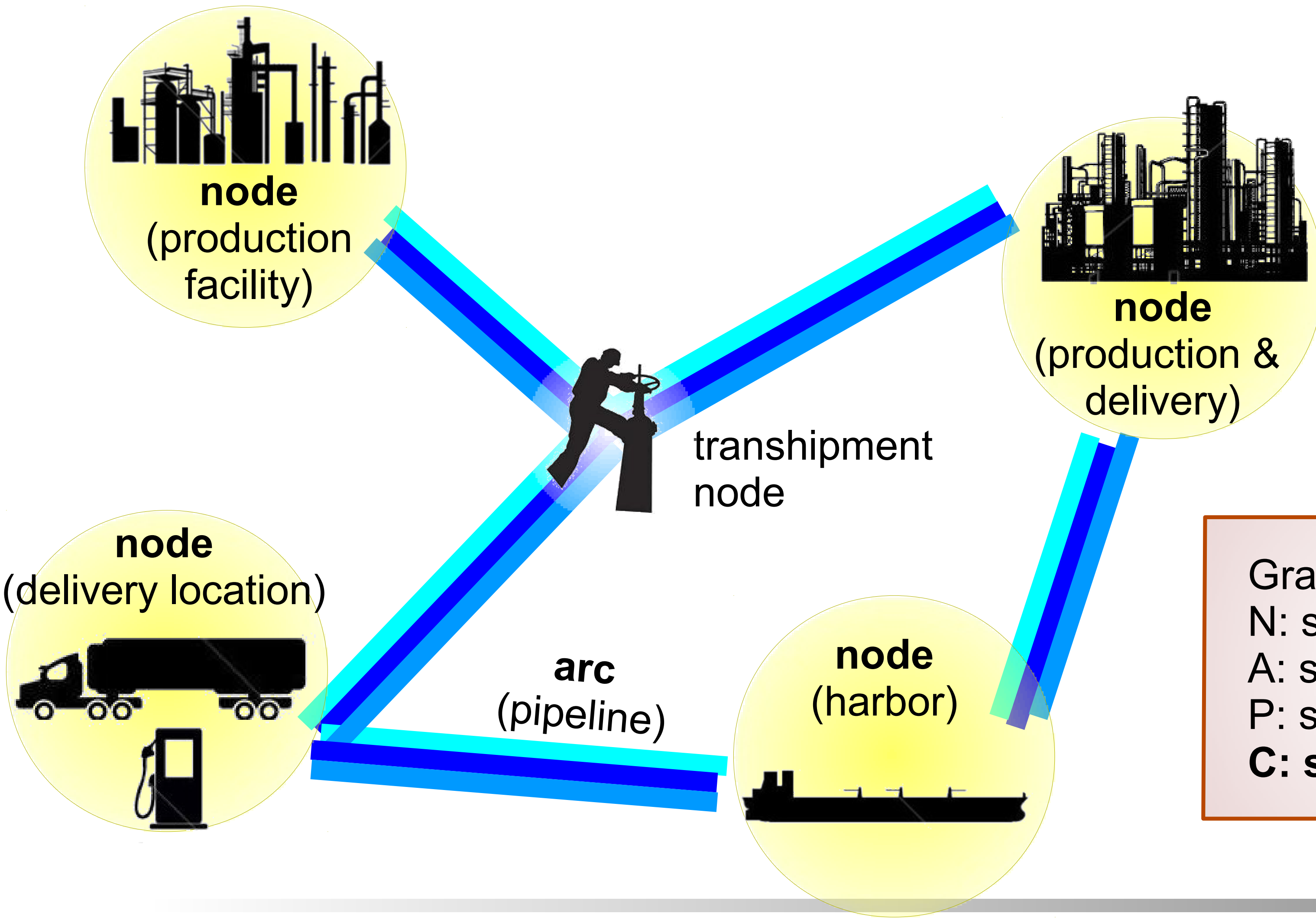
Flow constraints: enummeration of '*paths*'



Graph G(N, A)
N: set of nodes
A: set of arcs
P: set of paths

The Pipeline Operation

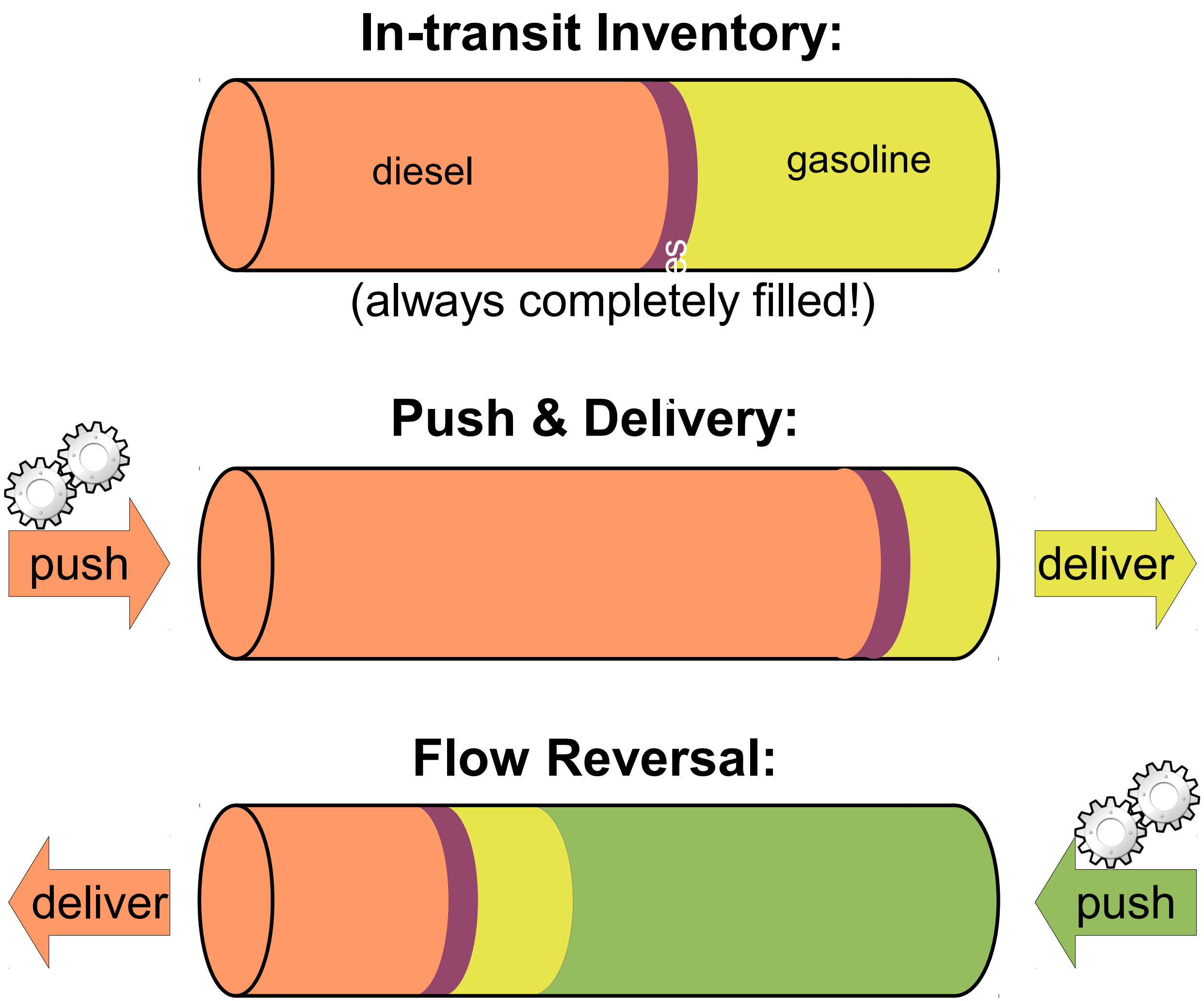
Layers of '*commodities*':



Graph G(N, A)
N: set of nodes
A: set of arcs
P: set of paths
C: set of commodities

The Pipeline Operation

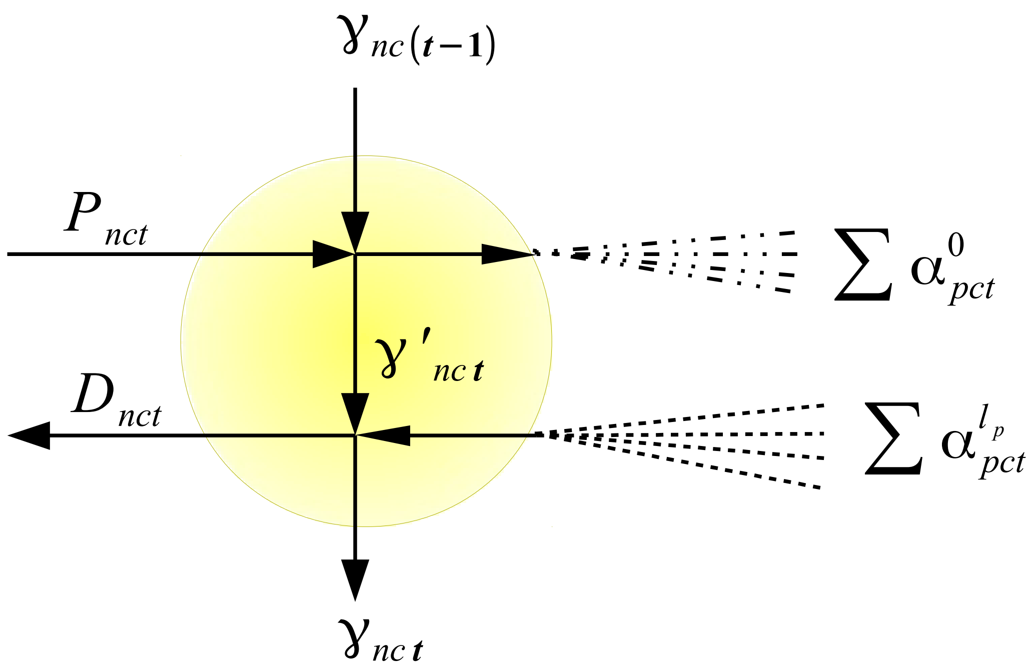
‘*In-transit inventory*’ on pipelines



Problem Formulation

Node: inbound and outbound paths

$\forall\, n \in N, c \in C, t \in T$



Parameters:

γ_{nc0} Node inventory
P_{nct} Production
D_{nct} Demand

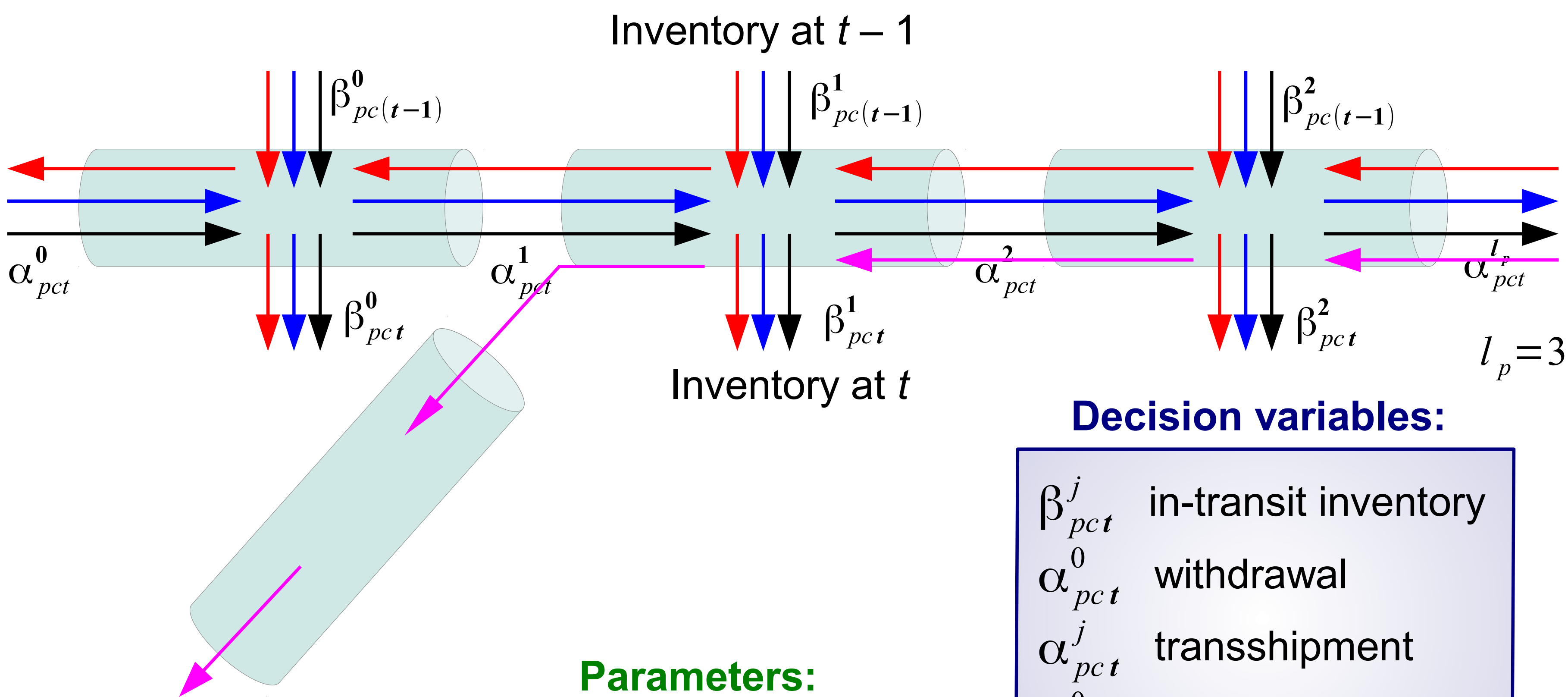
Decision variables:

γ_{nct} Node inventory
γ'_{nct}

Problem Formulation

<u>Paths</u>: sequence of among facilities and terminals

$\forall\, p \in P, c \in C, t \in T, j \in \{1 \ldots l_p - 1\}$



Decision variables:

β^j_{pct}	in-transit inventory
α^0_{pct}	withdrawal
α^j_{pct}	transshipment
α^0_{pct}	receipt

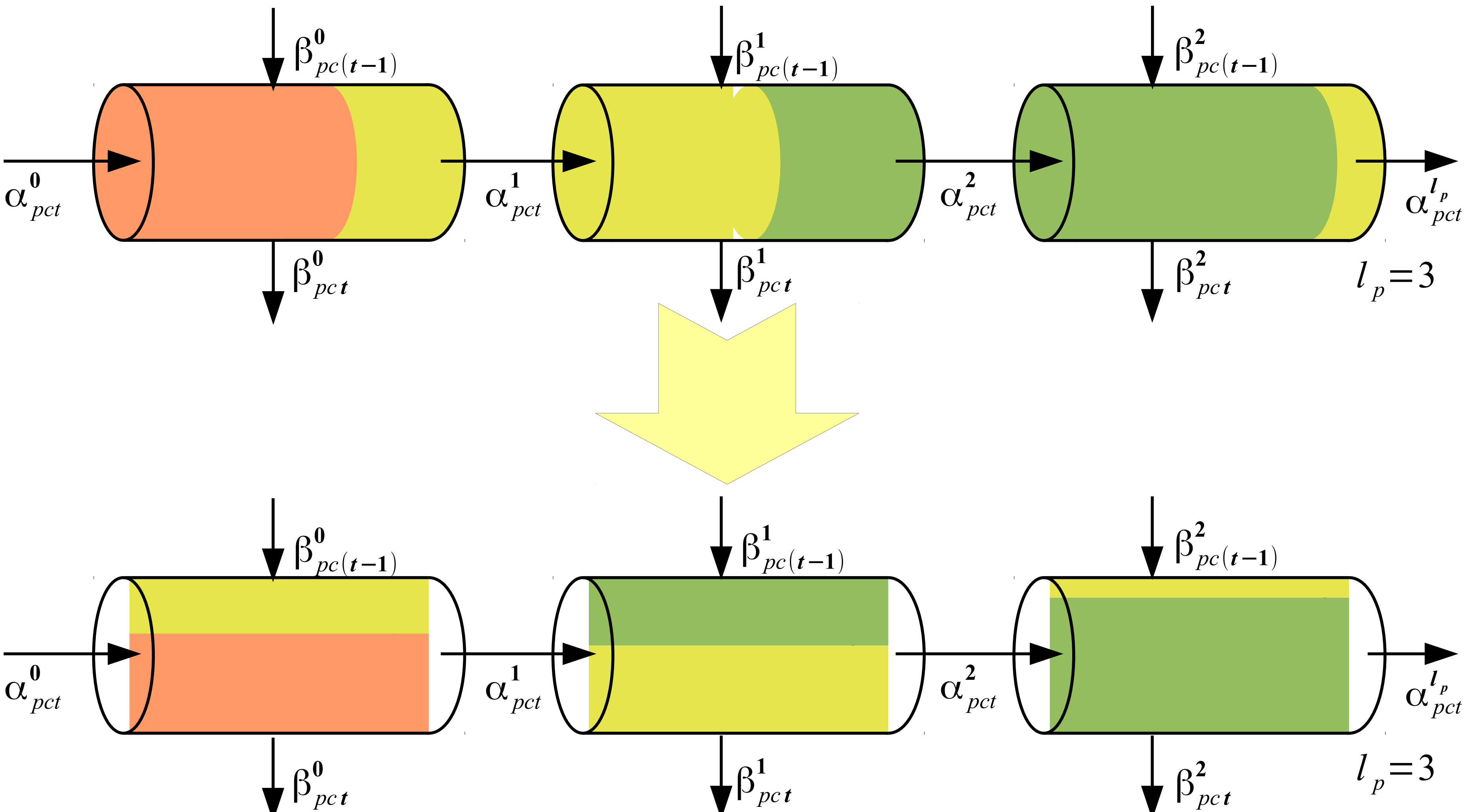
Parameters:

β^j_{pc0}	in-transit inventory

Problem Formulation

The '*arc inventory relaxation*':

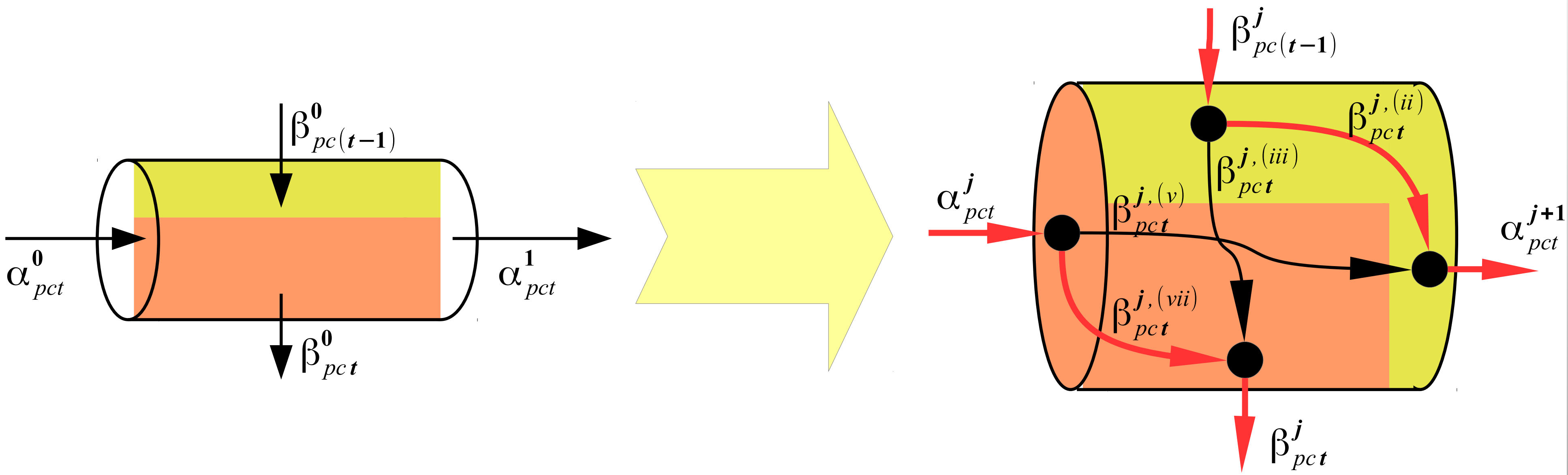
$\forall\, p \in P, c \in C, t \in T, j \in \{1 \ldots l_p - 1\}$



Problem Formulation

The '***arc inventory relaxation***' revealed:

$\forall\, p \in P, c \in C, t \in T, j \in \{1 \ldots l_p - 1\}$

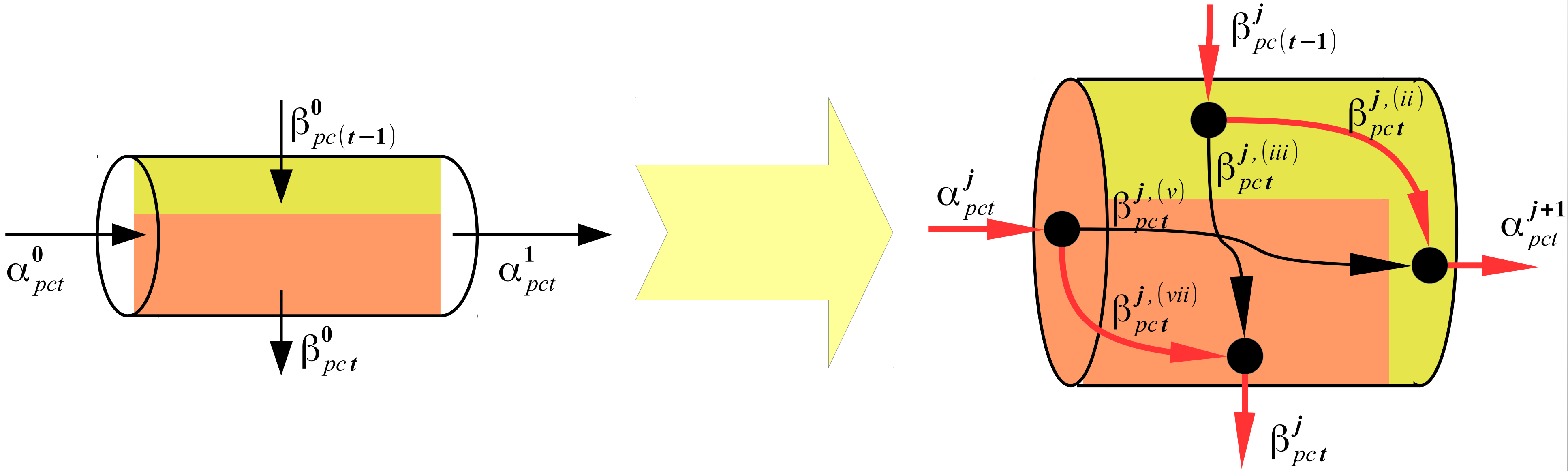


- First deliver current inventory.
- Only then transport the entering commodity.
- Keep part of the entering commodity as next inventory.

Problem Formulation

The ‘***arc inventory relaxation***’ revealed:

$\forall\ p \in P, c \in C, t \in T, j \in \{1 \ldots l_p - 1\}$

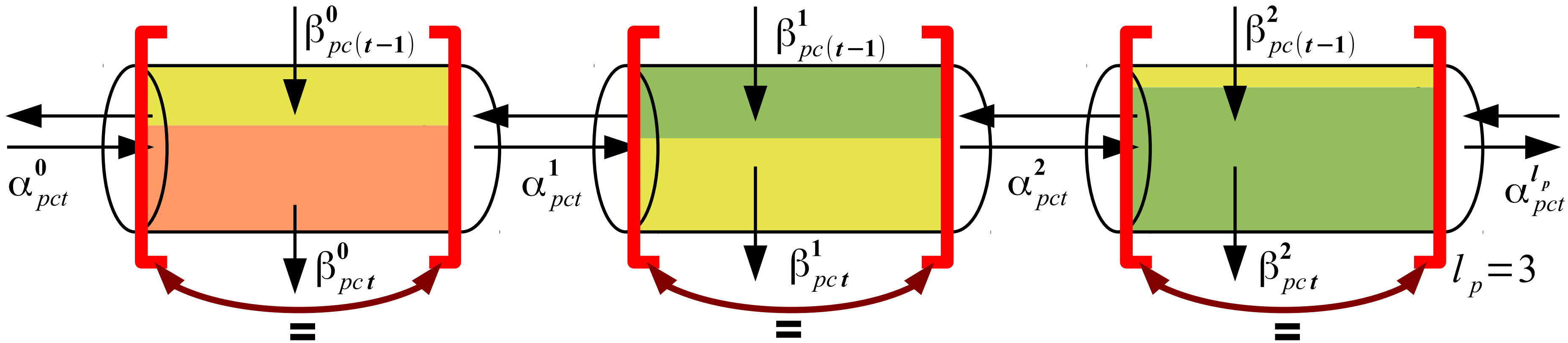


$$\min \sum_{p \in P,\, c \in C,\, t \in T} \mathcal{E}_{o_p c}(\alpha^{0}_{pct}) + \sum_{\substack{p \in P,\, c \in C,\, t \in T \\ j \in [1..l_p - 1]}} (\rho_\alpha \alpha^{j}_{pct} + \rho_\beta \beta^{j,(iii)}_{pct})$$

Problem Formulation

The '***arc flow relaxation***':

$\forall\, a \in A, t \in T$



Each Arc

Total inbound amount

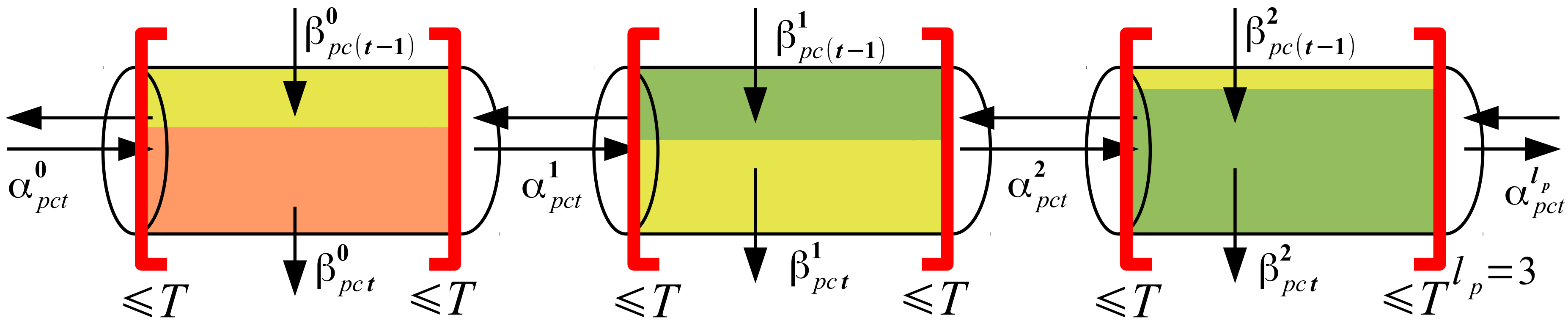
=

Total outbound amount

Problem Formulation

The ‘***arc flow relaxation***’ in action!

$\forall a \in A, t \in T$



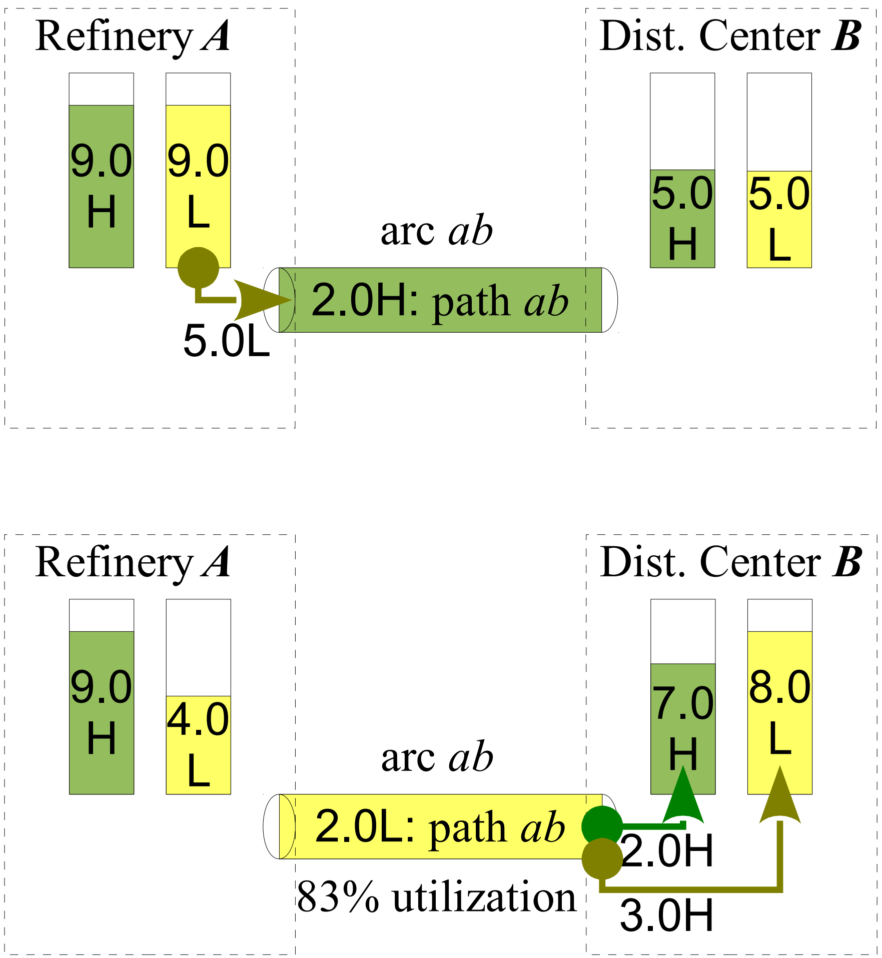
Each Arc End

Total time fits into the time slot

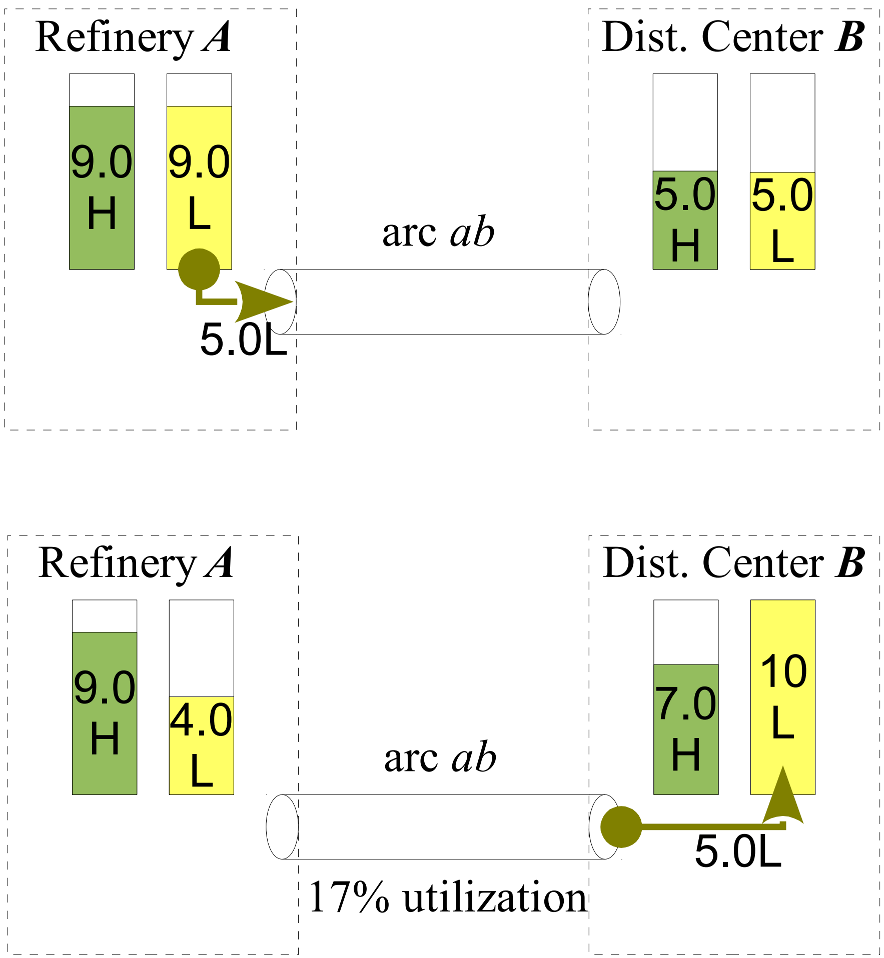
$$\sum \alpha^j_{pct} \leqslant T_t$$

Example

Arc & Inventory Relaxation Model:



Classic Network Flow Model:



Experiments

Typical instance:

- 75 classes of commodities,
- 25 nodes,
- 45 arcs
- 2 months planning horizon

Time Slices	Variables	Constraints	Execution Time
2	100,000	50,000	1 min
8	300,000	200,000	10 min

Conclusion

Network Flow Linear Programming:

- In-transit inventory
- Transit time
- Arc flow capacity
- Arc flow reversal

Benefits:

- More accurate flow and utilization rates
- Closer approximation to reality.

Challenge achieved:

No integer variables for a better pipeline network model!