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!
**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: enumeration 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

node (production facility)
node (production & delivery)
node (harbor)
node (delivery location)
arc (pipeline)
transhipment node
The Pipeline Operation

‘In-transit inventory’ on pipelines

In-transit Inventory:

- diesel
- gasoline

(always completely filled!)

Push & Delivery:

- push
- deliver

Flow Reversal:

- deliver
- push
Problem Formulation

Node: inbound and outbound paths
\[ \forall n \in N, c \in C, t \in T \]

\[ \gamma_{nc}(t-1) \]

\[ P_{net} \]

\[ D_{net} \]

\[ \sum \alpha^0_{pct} \]

\[ \sum \alpha^1_{pct} \]

Parameters:
- \( \gamma_{nc0} \): Node inventory
- \( P_{net} \): Production
- \( D_{net} \): Demand

Decision variables:
- \( \gamma_{nc0} \): Node inventory
- \( \gamma'_{nc0} \): Node inventory
Problem Formulation

Paths: sequence of among facilities and terminals

\[ \forall \ p \in P, c \in C, t \in T, j \in \{1...l_p - 1\} \]

Parameters:
- \( \beta^j_{pc(0)} \) in-transit inventory
- \( \alpha^0_{pc} \) receipt
- \( \alpha^j_{pc} \) withdrawal
- \( \alpha^j_{pc} \) transshipment

Decision variables:
- \( \beta^j_{pc(t)} \) in-transit inventory
The ‘arc inventory relaxation’:

∀ p ∈ P, c ∈ C, t ∈ T, j ∈ \{1...l_p - 1\}
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.
The ‘arc inventory relaxation’ revealed:

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

\[
\min \sum_{p \in P, \ c \in C, \ t \in T} \mathcal{E}_{opc}(\alpha_{pct}^0) + \sum_{p \in P, \ c \in C, \ t \in T, \ j \in [1 \ldots l_p - 1]} (\rho_{\alpha} \alpha_{pct}^j + \rho_{\beta} \beta_{pct}^j \cdot (iii))
\]
Problem Formulation

The ‘arc flow relaxation’:

\[ \forall a \in A, t \in T \]

\[ \alpha^0_{pct} \leq \beta^0_{pct}(t-1) \leq \beta^1_{pct}(t-1) \leq \beta^2_{pct}(t-1) \leq \alpha^1_{pct} \leq \alpha^2_{pct} \]

Each Arc

Total inbound amount = Total outbound amount

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

\[ \sum \alpha^j_{pct} \leq T_t \]
**Example**

**Arc & Inventory Relaxation Model:**

- **Refinery A**
  - 9.0 H
  - 9.0 L
  - 5.0 L

- **Dist. Center B**
  - 5.0 H
  - 5.0 L

arc ab

2.0 H: path ab

83% utilization

**Classic Network Flow Model:**

- **Refinery A**
  - 9.0 H
  - 9.0 L
  - 5.0 L

- **Dist. Center B**
  - 5.0 H
  - 5.0 L

arc ab

5.0 L

17% utilization

- **Refinery A**
  - 9.0 H
  - 4.0 L

- **Dist. Center B**
  - 7.0 H
  - 8.0 L

arc ab

2.0 H

5.0 L

2.0L: path ab

83% utilization

2.0H

3.0H
Experiments

Typical instance:

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

<table>
<thead>
<tr>
<th>Time Slices</th>
<th>Variables</th>
<th>Constraints</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>100,000</td>
<td>50,000</td>
<td>1 min</td>
</tr>
<tr>
<td>8</td>
<td>300,000</td>
<td>200,000</td>
<td>10 min</td>
</tr>
</tbody>
</table>
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!