Models and Solution Methods for the Pooling Problem

–Dissertation for the PhD-degree–

Mohammed Alfaki

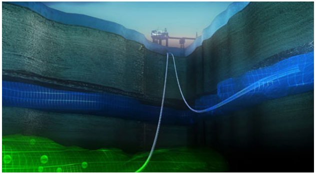
University of Bergen, Department of Informatics

June 18, 2012

Outline

Blending operations

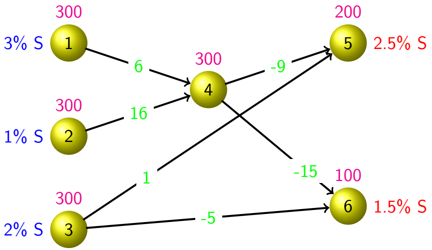
- The extracted crude petroleum contains impurities
  - Carbon-Dioxide: CO_2
  - Hydrogen-Sulfide: H_2S
  - Total Sulfur contents: S
  - Water vapor
  - ...



- Two major processes for purification:
  - Separation: Chemical and physical technologies
  - Blending: Mixture of different flow streams in pools (storage tanks)

The pooling problem – Haverly example

- Sulfur quality contents
- Sulfur quality bounds
- Arc unit cost
- Node capacity



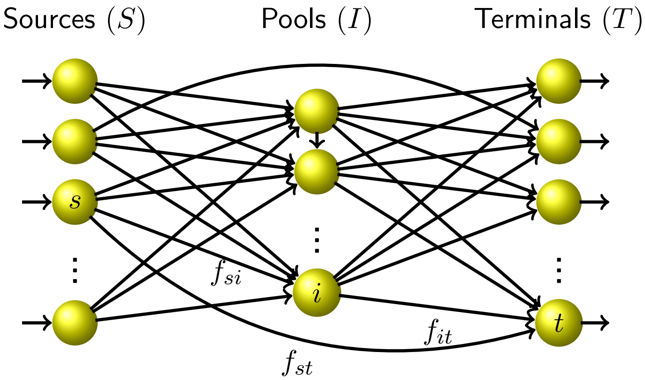
- The flow quality out of node 4 (the pool):
  - The volume-weighted average of the qualities of entering flow
- The goal is min the total cost, while satisfying the consumer
  1. Quality standard
  2. demand

The pooling problem in the industry

- Oil refining
  - Mixing crude oils
  - Blending gasoline components
- Pipeline transportation of natural gas
  - Flow from different sources share common pipelines
- Waste-water treatment planning

Network model

- A digraph $G = (N, A)$ with 3 sets of nodes

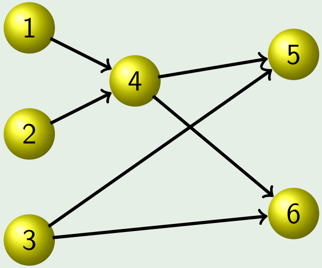


- Where f_{ij} is the flow along the arc $(i, j) \in A$

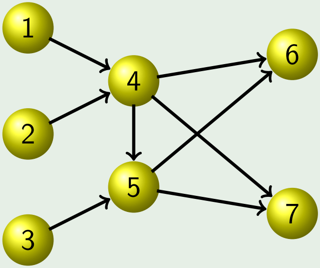
Pooling problems

- Standard pooling problem: $A \subseteq (S \times I) \cup (I \times T) \cup (S \times T)$
- Generalized pooling problem: General digraphs

Example (SPP)



Example (GPP)



Notations

Parameters

- Flow capacity: $b_i,\ i \in N$
- Unit cost: $c_{ij},\ (i,j) \in A$
- A set of quality attributes: K
- Quality parameter: $q_s^k,\ s \in S, k \in K$
- Quality bound: $q_t^k,\ t \in T, k \in K$

The flow constraints

$$\sum_{j:(s,j)\in A} f_{sj} \leq b_s,\ s \in S$$

$$\sum_{j:(j,i)\in A} f_{ji} \leq b_i,\ i \in N \setminus S$$

$$\sum_{j:(j,i)\in A} f_{ji} = \sum_{j:(i,j)\in A} f_{ij},\ i \in I$$

- Define

$$w_i^k = \begin{cases} q_i^k & \text{if } i \in S \\ \frac{\sum_{j:(j,i)\in A} w_j^k f_{ji}}{\sum_{j:(j,i)\in A} f_{ji}} & \text{if } i \in I \end{cases}$$

Mathematical model

P-formulation

- **Objective:** $\min \sum_{(i,j)\in A} c_{ij} f_{ij}$
- **The flow constraints**

$$\sum_{j:(j,i)\in A} w_j^k f_{ji} - \sum_{j:(i,j)\in A} w_i^k f_{ij} = 0, \quad i \in I,\ k \in K, \quad (1)$$

$$\sum_{j:(j,t)\in A} w_j^k f_{jt} - q_t^k \sum_{j:(j,t)\in A} f_{jt} \leq 0, \quad t \in T,\ k \in K. \quad (2)$$

- (1) and (2) $\Rightarrow$ nonconvex problem

Strong $\mathcal{NP}$-hardness

Maximum independent vertex set problem (MIVS)

find max subset of the vertices in a simple graph, s.t. all pairs of vertices in the subset are non-neighbors.

Proposition

The pooling problem is strongly $\mathcal{NP}$-hard.

- The proof is by a polynomial reduction from MIVS

Consequence

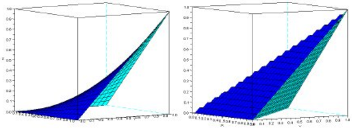
- The strong $\mathcal{NP}$-hardness of the pooling problem persists in two interesting special cases.
- Corollaries:
  1. The pooling problem for networks with only one pool
  2. The maximum flow pooling problem: replace min cost by max flow
- For the instance class where $|I| = 1$, $A = (S \times I) \cup (I \times T)$ and $|K| \leq \kappa$, then the pooling problem can be solved in polynomial time.

What we mean by strong formulation?

Linear relaxation of a bilinear problem

- McCormick's relaxation of wf on the rectangle $[\underline{w}, \overline{w}] \times [\underline{f}, \overline{f}]$, are

$$\text{Vex}(wf) = \max\left\{\underline{w}f + \underline{f}w - \underline{f}\underline{w}, \overline{w}f + \overline{f}w - \overline{f}\overline{w}\right\},$$
$$\text{Cav}(wf) = \min\left\{\underline{w}f + \overline{f}w - \overline{f}\underline{w}, \overline{w}f + \underline{f}w - \underline{f}\overline{w}\right\}.$$



- F_1 and F_2 are to formulations. F_1 is stronger than F_2 if $z_R^{F_1} \leq z_R^{F_2}$ and the inequality is strict for some instances

The standard pooling problem (SPP)

A model with source proportions

- The PQ-formulation (Tawarmalani and Sahinidis, 2002): By defining the source proportions[1]

$$y_i^s = f_{si} / \sum_{t \in T} f_{it}, \ (s,i) \in A,$$

then $w_i^k = \sum_{s \in S} q_s^k y_i^s$.

- PQ-formulation dominates the P-formulation

[1] Ben-Tal et. al (1994)

A model with terminal proportions

- TP-formulation: Analogous to the source proportions y_i^s, define the terminal proportions

$$y_i^t = f_{it} / \sum_{s \in S} f_{si}, \ \ (i,t) \in A$$

- By adding the cuts

$$\sum_{t \in T} f_{si} y_i^t - f_{si} = 0, \quad (s,i) \in A,$$
$$\sum_{s \in S} f_{si} y_i^t - b_i y_i^t \leq 0, \quad (i,t) \in A.$$

- Comparison to the PQ-formulation
  1. They do not have equal strength
  2. None of them dominates the other

A strong formulation for SPP

STP-formulation:

- Combine the source and terminal proportions
- The quantities $y_i^s f_{it}$ and $y_i^t f_{si}$ both can interpreted as the flow along the path (s, i, t) in G
- This model dominates both the PQ- and the TP-formulation

STP-relaxation:

- Given a rectangle for each bilinear term the linearization achieved by replacing $y_i^s f_{it}$ and $y_i^t f_{si}$ by
  1. Vex$(y_i^s f_{it})$, and Cav$(y_i^s f_{it})$
  2. Vex$(y_i^t f_{si})$, and Cav$(y_i^t f_{si})$

Branching strategy for the STP-formulation

- Choosing the branching pair in the STP-formulation
  1. Let $(\hat{y}, \hat{f})$ be the solution of the LP relaxation
  2. For each bilinear term define:

$$\mathsf{Infeas}(yf) = \max\left\{\hat{y}\hat{f} - \mathsf{Vex}(\hat{y}\hat{f}), \mathsf{Cav}(\hat{y}\hat{f}) - \hat{y}\hat{f}\right\}$$

  3. Find the max infeasibility pair in set $\{y_i^s f_{it}\}$
  4. Find the max infeasibility pair in set $\{y_i^t f_{si}\}$
  5. The branching pair: the min infeasibility pair from (3) and (4)
- We coded our own branch-and-bound with this branching strategy

Computational experiments (Paper A)

PQ vs. STP with BARON

- On a set of 20 large-scale instances
- STP has better lower bounds in 12 instances out of 20 (60%)
- Both they have identical lower bounds in the remaining instances

BARON vs. Our B&B

- Our B&B wins in 39% of instances
- BARON wins in 29% of instances

- The relative differences are however small

Consider the generalized pooling problem (GPP)

A hybrid formulation (HYB):

- The PQ-model is not applicable to the GPP (since pool to pool connections exist)
- Audet et. al (2004) suggested a model with both quality and proportion variables
- For pools that are 'close' to S, y_i^s are used. For other pools, w_i^k are used

A multi-commodity flow formulation

- Instead of w_i^k, associate a flow commodity with each $s \in S$
- f_{ij}: The total flow along (i, j)
- y_i^s: Proportion of commodity s of the flow leaving $i \in S \cup I$
- The quantity $y_i^s f_{ij}$ defines the flow of commodity s along arc (i, j)

The model for GPP

MCF-formulation

- **Objective: Minimize the total cost**
- **Flow capacity constraints**

$$\sum_{j:(j,i)\in A} y_j^s f_{ji} - \sum_{j:(i,j)\in A} y_i^s f_{ij} = 0, \quad s \in S,\ i \in I,$$

$$\sum_{j:(j,t)\in A} \sum_{s\in S} \left(q_s^k - q_t^k\right) y_j^s f_{jt} \leq 0, \quad t \in T,\ k \in K,$$

$$\sum_{s\in S} y_i^s = 1, \quad i \in I,$$

- **Adding RLT cuts**

Strength of MCF-formulation

- MCF is equivalent to PQ for the SPP
- MCF-formulation dominates bot of P- and HYB-formulations

MCF vs. P and HYB (results of Paper B)

- On a set of 40 instances (extended+randomly generated)
- We have used BARON as global solver
- MCF has solved 95% of the instances to optimality with 1 CPU-hour time limit
- Both P and HYB missed the global optimum in 45% of the instances

Linear Matrix Inequalities (LMI) Relaxation

- The polynomial optimization problem $f^* = \min_{x \in \mathbb{R}^n} \{f(x) : x \in \Omega\}$
- Is equivalent to

$$\mu^* = \min_{\mu \in \mathcal{B}(\Omega)} \int f(x) d\mu, \tag{3}$$

- A truncation of (3) is the LMI relaxation of order i:

$$\mathbb{Q}_i = \begin{cases} \min_y & \sum_\alpha (g_0)_\alpha y_\alpha, \\ \text{s.t.} & M_i(y) \succeq 0, \\ & M_{i-1}(g_k y) \succeq 0, \quad k = 1, 2 \ldots, m, \end{cases}$$

- $M_i(y)$ is the moment matrix of order i

LMI relaxation properties

- $\inf \mathbb{Q}_i \leq \inf \mathbb{Q}_{i+1} \leq f^*, \ i = 1, 2, \ldots.$
- If Ω has nonempty interior
  - Then, in many cases we have $\inf \mathbb{Q}_i = f^* \quad \forall \ i \geq i_0$

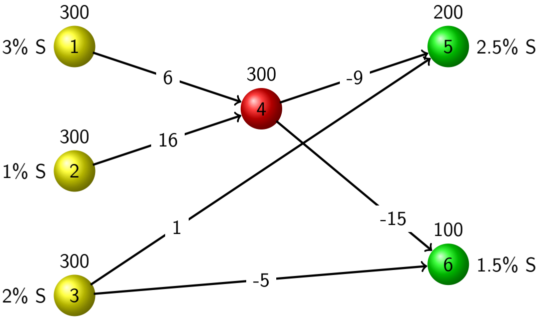
Results of Paper C

- The second-order relaxation is very strong for the single-quality pooling problem
- Max flow seems to be easier than min cost
- Many min cost instances can be solved with relaxation order $i \leq 3$
- Applicable only (for now) to small instances

Example – discretize the proportions

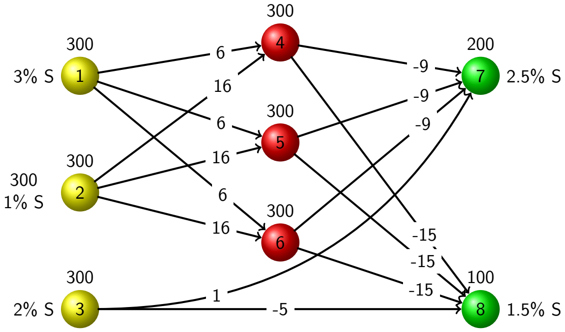
$y_4^1, y_4^2 \in [0,1] \leftarrow y_4^1, y_4^2 \in \left\{0, \frac{1}{2}, 1\right\}$. Since $y_4^1 + y_4^2 = 1$

$$\text{discretized proportions} = \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{pmatrix}$$



Example – extend the network

Then the discretized version of Haverly1 is shown below:



$$\text{discretized proportions} = \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{pmatrix}$$

A discretization method

- Extend the network to include the additional pools
- Each pool j corresponds to a unique combinations set of the discretized proportions
- Define:

$$p_j = \begin{cases} 1, & \text{if pool j is chosen,} \\ 0, & \text{otherwise,} \end{cases}$$

  and impose the constraint $\sum_j p_j = 1$
- Now the pooling problem is approximated by an MIP model with objective function value $z(n)$ that converges to the true global optimum as $n \to \infty$

n is $\#$ discretized points

Computational experiments (Paper D)

- The discrete model gives a good feasible solution even with coarse discretizations.
- The discretized model is superior to the continuous approach in most of the cases
- The discrete model inherits error in the original model

A construction heuristic

In iteration $p = 0$

- For each $\tau \in T$, define the subgraph G_τ that has only terminal τ
- $\left[P_\tau^0\right]$: The pooling problem define in G_τ with obj. value z_τ
- Order the terminal $T = \left\{\tau^1, \ldots, \tau^{|T|}\right\}$ such that $z_{\tau^1} \leq \cdots \leq z_{\tau^{|T|}}$

In iteration $p = 1, 2, \ldots, |T| - 1$

- F^{p-1} be the flow accumulation up to iteration $p - 1$ $(F^0 = 0)$
- $[P^p]$: find an optimal augmentation of F^{p-1} in the graph where $T^p = \left\{\tau^1, \ldots, \tau^p\right\}$

A construction heuristic (cont.)

Algorithm 1: The construction method.

Solve $\left[P_\tau^0\right]$ for all $\tau \in T$
Define $\left\{\tau^1, \ldots, \tau^{|T|}\right\}$ such that $z^0_{\tau^1} \leq \cdots \leq z^0_{\tau^{|T|}}$
$p \leftarrow 0$, $T^0 \leftarrow \emptyset$, $F^0 \leftarrow 0$
repeat
    $p \leftarrow p + 1$
    Solve $[P^p]$ to obtain the optimal solution (z^p, f^p, w^p)
    if $z^p < 0$ **then**
        $F^p \leftarrow F^{p-1} + f^p$
        $T^p \leftarrow T^{p-1} \cup \{\tau^p\}$
    end
until $p = |T|$

Computational experiments (Paper E)

- Experiments with 20 large-scale SPP
- Comparison with Multi-start heuristics (MINOS and BARON)
- We us BARON as global solver in the heuristic
- The construction method found the best solution in large instances

Concluding remarks

The main contributions of the thesis:

- A formal proof of the strong $\mathcal{NP}$-hardness
- Strong formulations:
  1. STP-formulation for SPP
  2. MCF-formulation for GPP
- New solution methods
  1. A method based on LMI relaxations
  2. A discretization method
  3. A greedy construction algorithm