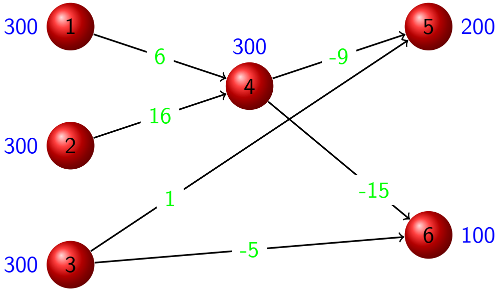
Comparison of discrete and continuous models for the pooling problem

Mohammed Alfaki

University of Bergen, Department of Informatics
Joint work with D. Haugland

ATMOS 2011 – Saarbrücken, Germany
September 8, 2011

Outline



Parameters

Given a network with

- Arc costs
- Node capacities

Min-cost flow problem

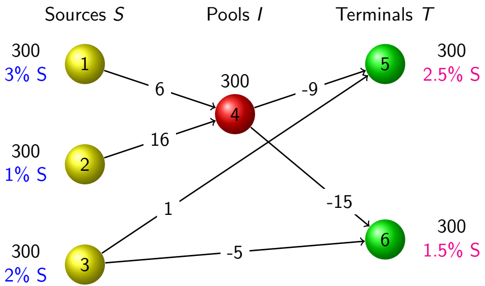
Assign flow to each arc such that the total cost is minimized

Extension of the Min-cost flow problem

Additional parameters

Relative Sulfur

- quality contents
- quality bounds



Haverly1 (Haverly, 1978)

The flow quality leaving any pool is defined by:

The weighted average of the quality of entering flow, where the flow values constitute the weights (linear blending)

Min-cost flow problem vs. pooling problem

Homogeneous flow (e.g. clean water)

- One flow type in the network
- Easy to solve (an LP model)

Inhomogeneous flow (e.g. Natural gas)

- More than one flow type
- Different flow qualities
  1. Qualities are given at each source
  2. Quality balances around each pool
  3. Quality restrictions at each terminal
- An NP-hard problem (Alfaki-Haugland, 2010)

Multi-commodity flow formulation

- Instead of quality variables, associate a flow commodity with each $s \in S$
- S_i: the set of all $s \in S$ where there is a path between s and $i \in S \cup I$
- f_{ij}: the total flow along (i, j)
- y_i^s: proportion of commodity s of the flow leaving $i \in S \cup I$ ($y_i^s = 0$ if $s \notin S_i$ and $y_s^s = 1$)
- The quantity $y_i^s f_{ij}$ defines the flow of commodity s along arc (i, j)

Multi-commodity flow formulation (cont.)

- **Objective**: minimize the total cost
- In addition to the flow capacity constraints we have

$$\sum_{j \in N_i^-} y_j^s f_{ji} - \sum_{j \in N_i^+} y_i^s f_{ij} = 0, \qquad s \in S_i,\ i \in I,$$

$$\sum_{j \in N_t^-} \left(\sum_{s \in S_j} q_s^k y_j^s - q_t^k \right) f_{jt} \leq 0, \qquad t \in T,\ k \in K,$$

$$\sum_{s \in S_i} y_i^s = 1, \qquad i \in I.$$

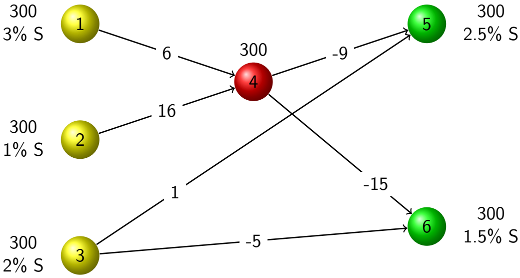
- The multi-commodity flow formulation for the pooling problem is stronger than the formulation that based on explicit quality variables (Alfaki-Haugland, 2010)

Motivations

- Bilinear constraints seem inescapable in continuous models
- Traditional solution methods (e.g. Branch-and-bound) have slow convergence
- Approximate the NLP model using other well-studied models, e.g. MIP
- Strategy to find a good feasible solution in reasonable time

Example

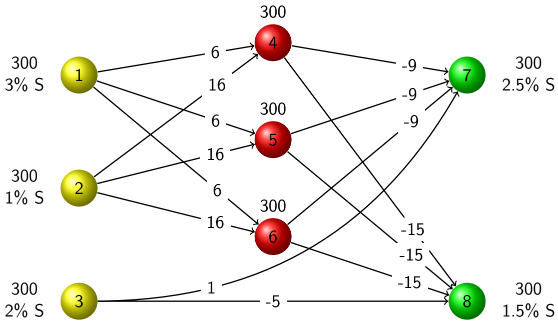
$y_4^1, y_4^2 \in [0, 1] \leftarrow y_4^1, y_4^2 \in \left\{0, \frac{1}{2}, 1\right\}$. Since $y_4^1 + y_4^2 = 1$



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

Example

Then the discretized version of Haverly1 is shown below:



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

The discrete model

- Extend the network to include the additional pools
- Each pool j corresponds to a unique combinations set 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

Instance characteristics

- 35 large-scale instances
- BARON to solve the continuous model
- CPLEX to solve the discrete model
- With one hour of CPU time

Group	#instances	Size of node and quality sets				#arcs range
		$\lvert S \rvert$	$\lvert I \rvert$	$\lvert T \rvert$	$\lvert K \rvert$	
arbC	5	8	6	6	4	57 – 82
arbD	5	12	10	8	5	114 – 166
arbE	5	10	10	15	12	181 – 248
stdA	10	20	10	15	24	171 – 407
stdB	6	35	17	21	34	384 – 1044
stdC	4	60	15	50	40	811 – 1451

Computational results

Instance	Continuous model		Discrete model		
	LB	UB	$z(n=2)$	$z(n=3)$	$z(n=5)$
arbC0	-1352.72	**-1352.72**	-1262.38	-1348.83	-1350.30
arbC1	-673.86	**-673.86**	-508.00	-615.50	-655.62
arbC2	-1716.62	**-1716.62**	-1688.69	-1705.81	(-1710.76)
arbC3	-1512.10	**-1512.10**	-1489.70	-1505.43	(-1508.92)
arbD0	-1994.00	-1571.11	-1833.33	**-1911.35**	—
arbD2	-2071.00	-2065.85	-2069.06	**-2070.16**	—
stdA0	-37402.74	-5383.70	-31990.52	-34175.71	(**-34853.43**)
stdA1	-30362.74	-29276.56	-24590.16	-25179.84	(**-28389.31**)
stdA2	-23044.16	**-23044.16**	-19846.94	-20666.60	(-21795.71)
stdA3	-41113.10	-31258.05	-36233.75	-37116.64	(**-38624.98**)
stdA4	-42999.89	-8770.94	-38126.91	(**-39331.58**)	(-39345.90)
stdA5	-28257.75	-6369.59	-26447.07	(**-27008.30**)	(-26729.51)
stdA6	-42463.05	-9555.82	-41777.00	(**-42022.93**)	(-41829.91)
stdA7	-44682.25	-5762.08	-42582.29	(**-43309.48**)	(-42227.89)
stdA8	-30666.87	-6576.76	-30341.61	(**-30435.00**)	(-30265.99)
stdA9	-21933.99	-14059.98	-21887.77	(**-21891.96**)	(-21527.08)
stdB0	-45441.79	-9075.24	-40171.43	(**-41036.54**)	(-40600.32)
stdB1	-65468.81	-34069.43	-60720.54	(**-62445.97**)	(-61858.06)
stdC0	-98792.76	-2400.00	(-77517.74)	(**-79384.25**)	—
stdC1	-119006.17	-12114.75	(**-97290.27**)	(-91215.32)	—
stdC2	-135916.19	-6342.08	(**-117024.36**)	(-115594.77)	—

Summary

1. Discretized the pooling problem by MIP model
2. The discrete model gives a good feasible solution even with coarse discretizations.
3. Computational experiments indicate that the discretized model is superior to the continuous approach in most of the cases
4. The discrete model inherits error in the original model

Thanks for your attention!