A Path Formulation for the Generalized Pooling Problem

Mohammed Alfaki

3rd Nordic Optimization Symposium, Stockholm, KTH

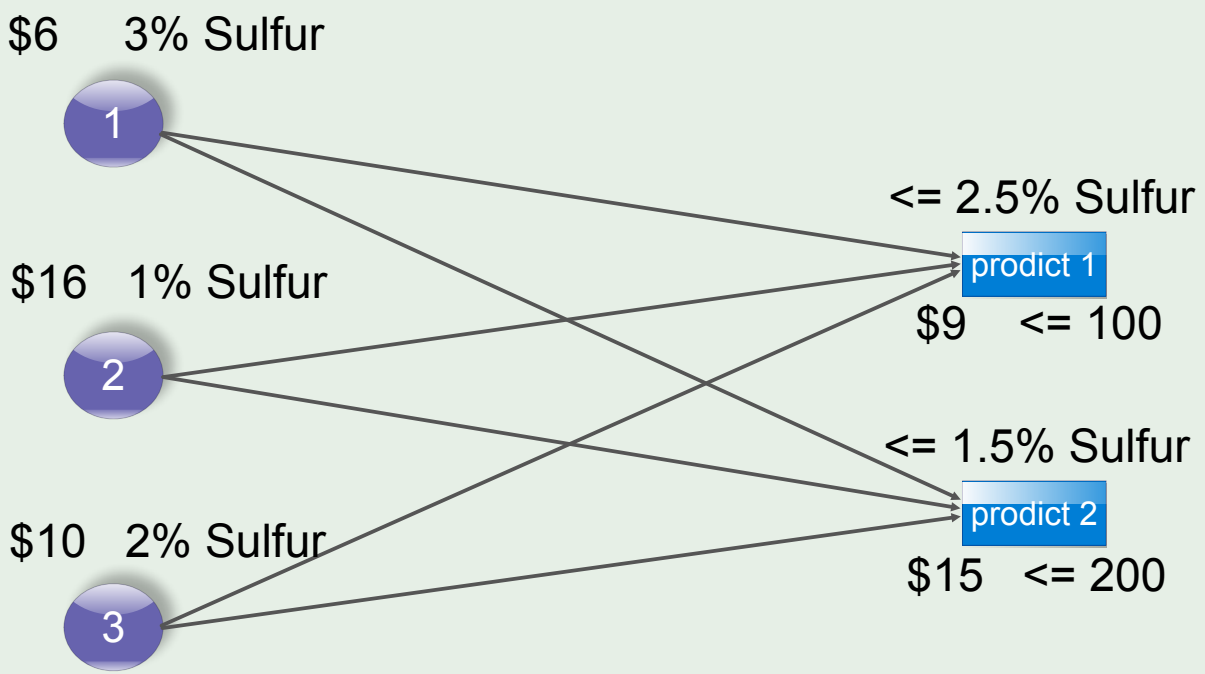
March 13, 2009

Outline

What is the pooling problem?

Example

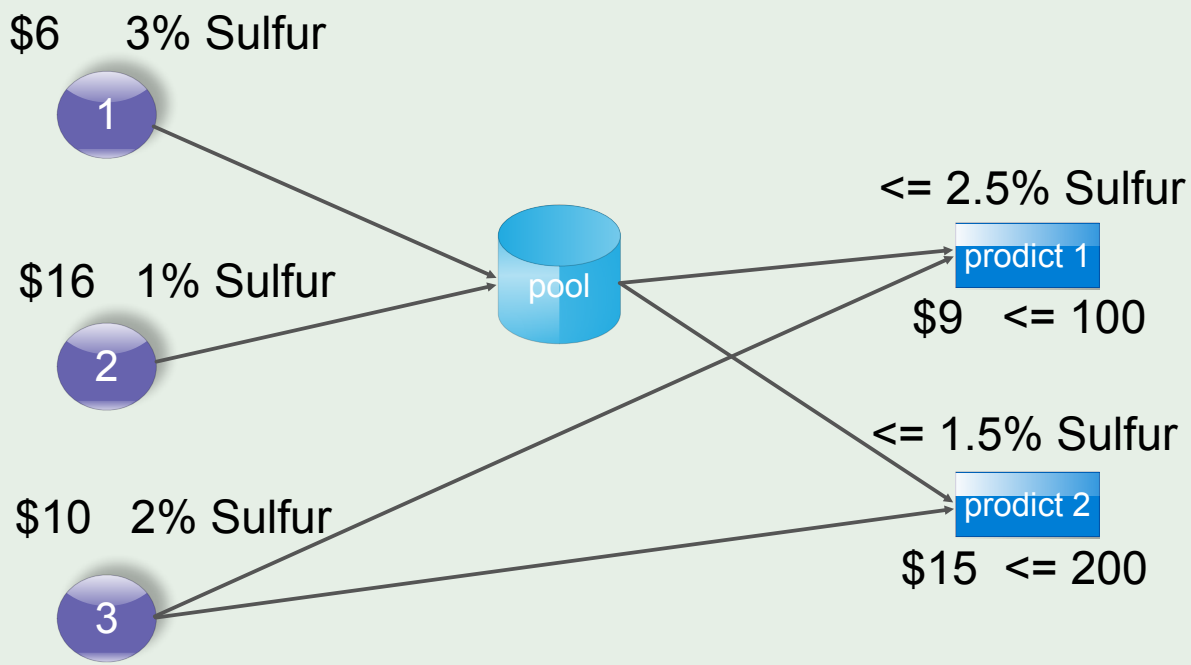
Blending problem:



What is the pooling problem?

Example

Pooling problem:



Haverly problem -1978

NP–hardness

Definitions

- Mixing raw materials in pools
- Two stage blending problem
- The model is non–linear
- Many local optima
- Hard to find global optimum

Theorem

The pooling problem is NP-hard even in the case of single-layer of pools.

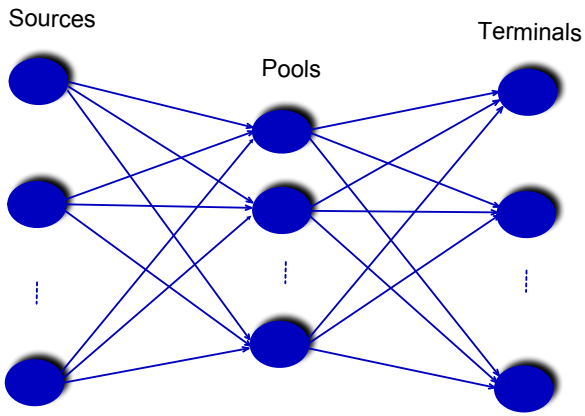
Proof: *A poly reduction from the 3-dimensinal matching problem to the single-layered pooling problem.*

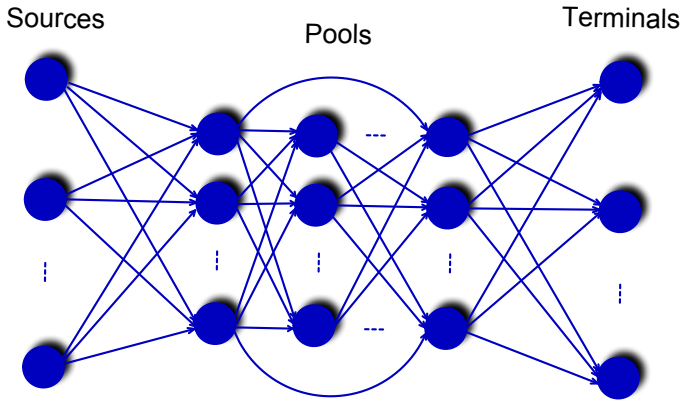
State of the art in global methods

- The mathematical models for the pooling problem based on non–linear formulation
- Nonlinearity appears in two type of constraints
  1. Quality balance at pools
  2. Quality bound at the terminals (products)
- Two formulations exist:
  1. p–formulation (Haverly 1978): consists of flow and quality variables
  2. q–formulation (Ben–Tal et al. 1994): replacing the quality variables with variable representing flow proportions.
- q–formulation is tighter than p–formulation
- Tawarmalani & Sahinidis 2005 gave the pq–formulation, which is even tighter

Generalization and important applications

- The pq–formulation is not applicable when we have multi–layered of pools
- Important application do exist fo the multi–layered pooling problem:
  1. Pipeline transportation of dry gas
  2. Multi–period inventory model

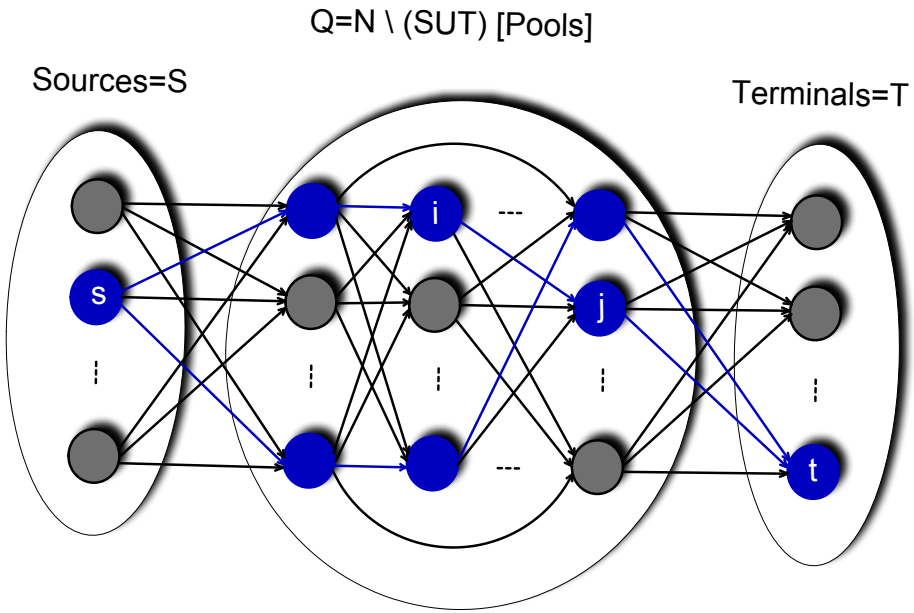




Arc–oriented formulation

- $G = (N, A)$ is DAG
- Parameters:
  1. q_i, $i \in S \cup T$, $k \in K$
  2. u_i node capacity of $i \in N$
  3. c_{ij} unit cost of $(i, j) \in A$
- Variables:
  1. w_i S quality leaving $i \in Q$
  2. v_{ij} the flow along $(i, j) \in A$
- Minimize the cost
- Sulfur Quality balance

$$w_i = \frac{\sum_{j \in N_i^-} w_j v_{ji}}{\sum_{j \in N_i^-} v_{ji}}, i \in Q$$



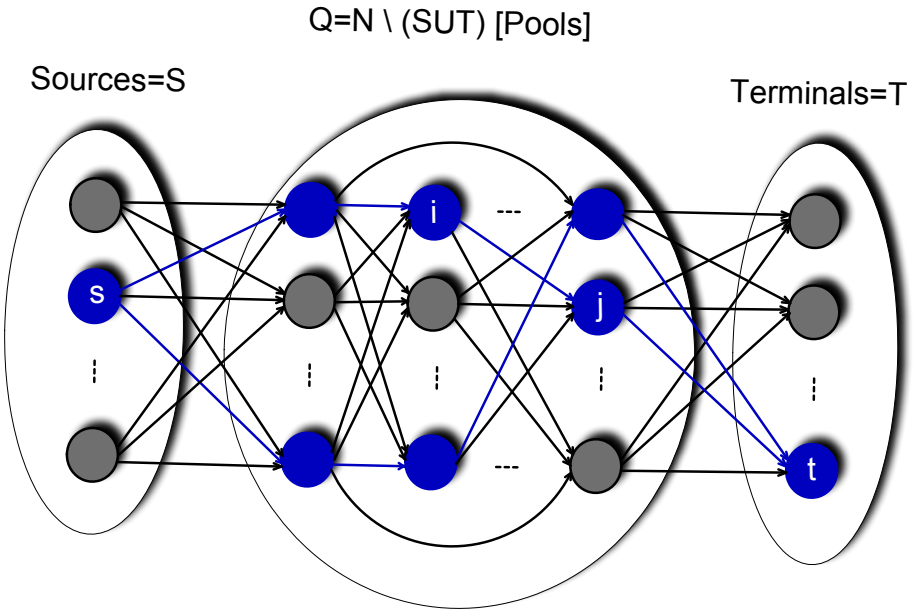
Path–oriented formulation

- Variables:
  1. v_{ij} the flow along $(i,j) \in A$
  2. v_{ij}^s flow originating from $s \in S$
  3. x_p flow along path $p \in P$
  4. y_i^s proportion of flow at pool i coming from $s \in S$
- Minimize the cost
- Propostion variables:

$$y_i^s = \frac{v_{ij}^s}{v_{ij}}, s \in S, i \in Q, (i,j) \in A$$

- legal flow:

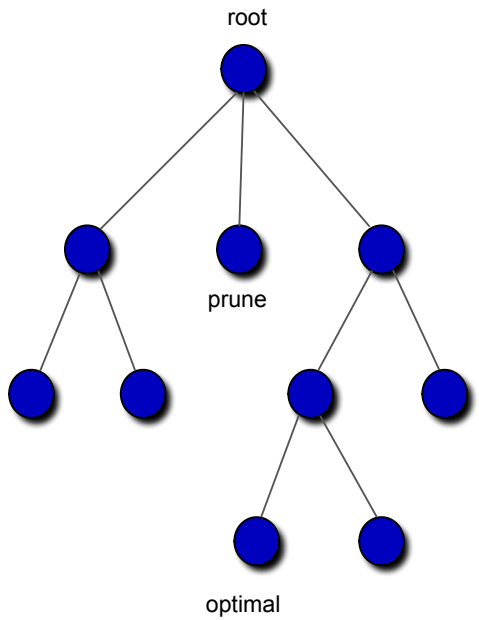
$$v_{ij}^s - \sum_{p \in P_{ij}^s} x_p = 0, s \in S, (i,j) \in A$$



- P: set of directed paths from some source to some sink
- $P^s \subseteq P$: set of paths start with source $s \in S$
- $P_{ij} \subseteq P$: set of paths intersecting $(i,j) \in A$
- $P_{ij}^s = P^s \cap P_{ij}$

Linear relaxation

- The idea based on B&B algorithms for integer programming
- Subproblem: linear relaxation
- Solve the relaxed problem at each node to get upper or lower bounds
- Prune: by bound, infeasibility, optimality
- Stop when the tree is empty



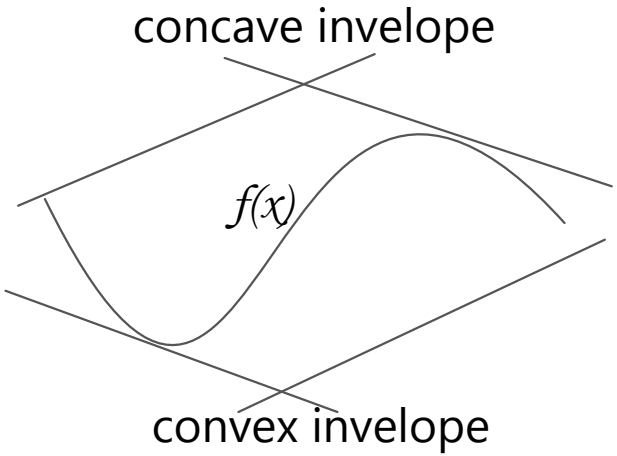
Convex and concave envelopes

- Sutting $\zeta = yv$ the convex and concave envelopes on rectangle $\mathcal{C} = [\underline{y}, \overline{y}] \times [\underline{v}, \overline{v}]$ are:

  $max\{\underline{y}v + y\underline{v} - \underline{y}\,\underline{v}, \overline{y}v + y\overline{v} - \overline{y}\,\overline{v}\}$

  $min\{\underline{y}v + y\overline{v} - \overline{y}\,\underline{v}, \overline{y}v + y\underline{v} - \underline{y}\,\overline{v}\}$

- Replacing the bilinear constraints with $\zeta^s_{ij} - v^s_{ij} = 0$
- Adding the above bounds to give linear relaxation problem R



Column Generation–General idea

- Column generation helpful in problems with many variables/columns but relatively few constraints
- Exclude some of the variables
- The master problem: original problem with only a subset of variables
- The subproblem: to identify whether there is excluded column that reduces the objective function value

How it works here?

- The master problem R' will be the problem R with only $P' \subseteq P$
- The idea is to solve R', and whenever we detect that the cost can be reduced by sending flow along any path $p \in P \setminus P'$, we augment P' by p
- We are looking for a path that reduces the cost
- The subproblem will be the Shortest–Path problem, which is solvable in polynomial time, since G is DAG

Algorithm for solving R

At each node in the B&B tree we will solve the following:

repeat
  solve the master problem R'
  for $(\forall\, s \in S)$ **do**
    define arc lengths
    $p \leftarrow$ shortest path from s to some $t \in T$ in G
    if $(length(p) < 0)$ **then**
      $P' \leftarrow P' \cup \{p\}$
    end
  end
until $(no\ path\ of\ length < 0\ \ found)$

Conclusion

- This is work in progress.
- The pooling problem is NP–hard
- The path–oriented formulation is equivalent to the pq–formulation, when we have one layer of pools
- Generalization of the strongest known formulation for one–layer instances

- Outlook
  - Experimental evaluation